



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA

# La Lógica Fuzzy y su Aplicación en la Limitación de Recursos

---

Máster en Edificación, Especialidad en Gestión

**Alumna:** María Josefa González Redondo

**Tutor:** José Luis Ponz Tienda

Curso Académico 2012- 2013



ESCUELA TÉCNICA SUPERIOR  
INGENIERÍA DE  
EDIFICACIÓN





Los grandes conocimientos engendran las grandes dudas.

*Aristóteles (384 AC-322 AC) Filósofo griego*

## 0. Índice de contenidos

<b>1. Introducción y Objetivos .....</b>	<b>5</b>
<b>2. Antecedentes Históricos de la Gestión y Limitación de Recursos .....</b>	<b>11</b>
<b>3. El Problema de la Programación de Proyectos con Recursos Limitados .....</b>	<b>17</b>
3.1. El Proyecto y su Gestión.....	17
3.2. Concepto y Tipos de Recursos .....	20
3.3. La Gestión de Recursos .....	23
3.4. Introducción a la Programación de Proyectos con Recursos Limitados .....	26
3.5. El Problema de la Programación o Secuenciación de Actividades.....	27
3.6. El Problema de la Limitación en el Consumo de los Recursos; el RCPSP estándar ....	34
<b>4. Clasificación y resolución de Problemas de Programación de Proyectos con Recursos Limitados .....</b>	<b>39</b>
4.1. Clasificación de Problemas de Programación de Proyectos con Recursos Limitados	39
4.2. Resolución de Problemas de Programación de Proyectos con Recursos Limitados...	42
4.3. Algoritmos Exactos.....	44
4.4. Algoritmos Heurísticos .....	44
4.5. Algoritmos Metaheurísticos.....	48
4.6. Algoritmos Hiperheurísticos.....	49
<b>5. El Problema de la Imprecisión en la Estimación de los Valores .....</b>	<b>51</b>
<b>6. La Limitación de Recursos con Valores Imprecisos .....</b>	<b>53</b>
6.1. Imprecisión vs Incertidumbre .....	53
6.2. La lógica Fuzzy.....	55
<b>6.2.1. Introducción a la Lógica Fuzzy. ....</b>	<b>55</b>
<b>6.2.2. Aritmética y ordenación de los números difusos.....</b>	<b>60</b>
6.3. El Estado del Arte de la Limitación de Recursos con Valores Imprecisos .....	62
6.4. Los valores difusos para el RCPSP .....	63
<b>7. Aplicación de la Lógica Fuzzy al RCPSP, un Caso de Estudio .....</b>	<b>65</b>
7.1. La PSPLIB – “Project Scheduling Library”-Biblioteca de Problemas Programación ....	65
7.2. Diseño y Estrategia de la Experimentación.....	67
7.3. Discusión de Resultados.....	80
<b>8. Conclusiones y Futuras Líneas de investigación.....</b>	<b>83</b>



---

<b>9. Índice de Figuras</b> .....	85
<b>10. Bibliografía</b> .....	87



## 1. Introducción y Objetivos

Desde la antigüedad (Grecia, Roma, Egipto...), el hombre ha organizado y programado sus proyectos, dejando una buena muestra de ello en las importantes construcciones históricas que han llegado hasta nuestros días en mejor o peor estado de conservación. Las obras faraónicas, los hermosos templos griegos, las grandes obras de infraestructuras romanas, eran estudiadas adecuadamente antes de su inicio partiendo de un proyecto realizado por expertos ingenieros. En el caso del Imperio Romano estos ingenieros incluso se desplazaban con las legiones que conquistaban nuevas tierras para Roma, de esta forma, ante cualquier obstáculo natural (ríos, montañas, zonas pantanosas) desarrollaban rápidamente sus proyectos, permitiendo en poco tiempo el avance de los implacables soldados.

La palabra “proyecto” que proviene del latín *proiectus* (diseño o pensamiento de ejecutar algo) aparece tratada en la literatura en múltiples ocasiones y es aplicable a diferentes entornos. Según indica el Project Management Institute (PMI) (PMI, 2012), en su “Guía del PMBOK” (“Project Management Body of Knowledge Guide”) (PMI, 2004), *un proyecto es un esfuerzo temporal que se lleva a cabo para crear un producto, servicio o resultado único*, es decir, no es repetitivo, como podría ocurrir por ejemplo en la línea de una cadena de fabricación. Aunque dos proyectos compartan el mismo diseño, los productos finalmente conseguidos serán siempre diferentes y singulares, derivándose de ello las dificultades que entraña todo el procedimiento. En un sentido amplio se podría decir que un **proyecto** es llevado a cabo por una o varias personas físicas o jurídicas, para desarrollar un conjunto de actividades ordenadas y relacionadas entre sí, con el objeto de alcanzar un fin determinado, producto o sistema.

Para (Demeulemeester & Herroelen, 2002) todos los proyectos tienen en común las siguientes características:

- *Una meta u objetivo.* El producto final debe realizarse de acuerdo a un coste, una calidad y un ritmo previstos con anterioridad al inicio de su construcción o fabricación. Si se consigue el proyecto habrá sido un logro.
- *Son únicos*, como ya se ha explicado anteriormente.
- *Son complejos.* Las relaciones entre las tareas a realizar pueden ser complejas y se requiere de la pericia de expertos para solucionar los conflictos que surjan. Todo ha de funcionar de manera fluida y debe encajar como en un puzle.
- *Son temporales.* Los proyectos tienen un inicio y un final definidos; esto implica que hay que dedicar el número de recursos necesarios para llevarlos a cabo. Será una derrota el no poder cumplir con los plazos establecidos.
- *Existe la incertidumbre.* Los proyectos necesitarán de una planificación, gestión y control para que estén expuestos a los mínimos riesgos posibles. Pero a pesar de estos esfuerzos, no se puede tener la certeza absoluta de que todo va a funcionar según lo previsto, es más, eso será imposible, siempre habrá algún elemento que no se haya



definido perfectamente o que se cambie, puede que se tenga algún problema con el suministro de recursos, que se convoque una huelga de trabajadores o que las inclemencias meteorológicas se pongan en nuestra contra..., sea como sea, el gestor del proyecto siempre encontrará impedimentos para la realización exacta del proyecto según se había previsto inicialmente. El riesgo es inevitable, es inherente a todo proyecto aunque se lleve un control riguroso y absoluto sobre el mismo.

En consecuencia, las actividades que forman el proyecto deben ser gestionadas y controladas adecuadamente para minimizar los riesgos que pueden surgir en los cuatro pilares fundamentales sobre los que se asienta todo proyecto: la calidad, los recursos, el plazo y la rentabilidad. Por tanto, para poder concluir con éxito las distintas tareas a realizar, es necesario hacer una buena **GESTIÓN** de todo el proceso y dentro de esa gestión, un área muy importante a tratar es la dedicada a los **RECURSOS**<sup>1</sup>.

Para gestionar los recursos se utilizan métodos basados en el raciocinio, en la **lógica**, la cual según la Real Academia Española, en adelante RAE, es la “*ciencia que expone las leyes, modos y formas del conocimiento científico*”, es decir, es la ciencia del pensamiento y la razón, dicho muy a grosso modo.

Puesto que un rasgo distintivo del ser humano es su capacidad para pensar, desde la antigüedad éste ha intentado desvelar los misterios del pensamiento y cuáles son los mecanismos utilizados por la mente para su desarrollo y transmisión, llegando en los últimos tiempos a emularlo mediante la inteligencia artificial o computacional, y encontrando en el cerebro humano, una máquina perfecta de la que en estos momentos apenas se conoce la punta del iceberg, tanto en lo referente a sus funciones fisiológicas como cognitivas. De esta manera las experiencias científicas emanan del análisis, la síntesis, la comparación, la generalización y/o la abstracción de las operaciones racionales, ya sean reales o imaginarias. Resumiendo, somos seres lógicos y racionales.

Del pensamiento formal surgió la **lógica clásica**, cuyo padre y fundador fue Aristóteles. Esta mantiene que dado un conjunto de elementos, se puede decir que una unidad pertenece o no a él, si posee o no la condición o condiciones que caracterizan el dominio de dicho conjunto. Efectivamente, si por ejemplo para pertenecer al conjunto de los seres vivos hay que tener capacidad para desempeñar las funciones básicas de nutrición, relación y reproducción, es evidente que un mosquito se puede considerar como un ser vivo puesto que cumple las condiciones necesarias y sin embargo una ventana no lo sería.

Pero a veces no resulta tan fácil determinar si un elemento pertenece o no al 100% a un conjunto, ya que puede hacerlo sólo en un determinado grado. En ocasiones las cosas no son blancas o negras, ni tan siquiera grises; baste recordar como muestra de esto que los esquimales debido a su experiencia y su instinto de supervivencia, son capaces de distinguir hasta 30 tipos de blancos diferentes, o que usan 40 sustantivos diferentes para darle nombre a la nieve. En consecuencia, se hace necesaria una lógica más flexible, llamada **lógica difusa**, **borrosa** o **fuzzy**. La lógica borrosa o difusa, es la definida por la RAE como “*la que admite una cierta incertidumbre entre la verdad o falsedad de sus proposiciones, a semejanza del*

---

<sup>1</sup> Ver la definición de RECURSO en el apartado 3.2.

*raciocinio humano*” y a la hora de tomar decisiones, permite abordar los problemas en función de parámetros cualitativos.

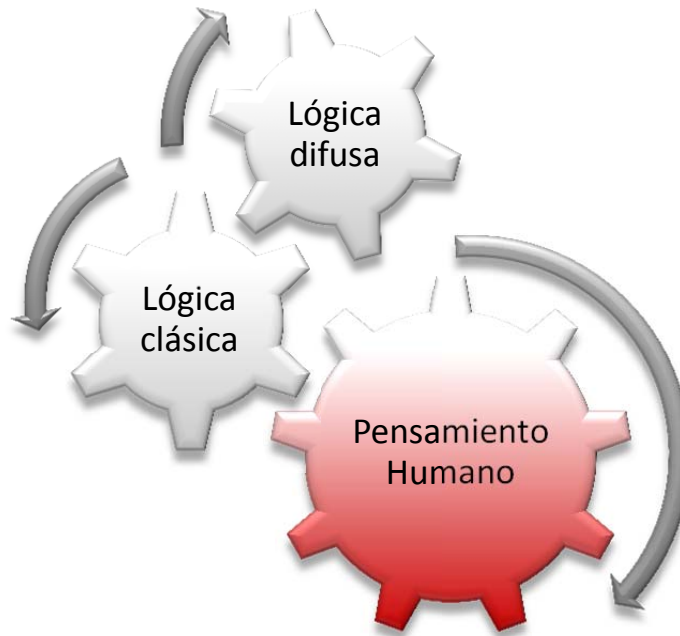


Figura 1: Funcionamiento del razonamiento humano

### Objetivos y contenido

El propósito de este proyecto es profundizar en los problemas que generan la asignación y la limitación de los recursos en la programación de proyectos, analizando y aplicando los conceptos de la lógica fuzzy para encontrar soluciones.

En el Capítulo 2 se examinan cuáles son los antecedentes históricos de la gestión y limitación de recursos, cuyo origen se remonta a la civilización egipcia y la organización del trabajo que desarrolló esta civilización para poder realizar sus grandes obras arquitectónicas, siendo a partir del siglo XIX cuando se establece el nacimiento de la organización del trabajo y de los recursos como ciencia. Su rápido avance va de la mano de otras ciencias como la Investigación Operativa o la Informática.

A continuación el Capítulo 3 define en qué consiste cada una de las fases del ciclo de vida del proyecto y la importancia que tienen las etapas de la planificación y la programación para garantizar el éxito del producto final que se obtendrá al terminar un proyecto. Se analiza el concepto de recurso y se clasifican los recursos en distintas tipologías tras repasar la literatura de los artículos técnicos existentes sobre la asignación y la limitación de recursos, e indicando ejemplos de las distintas modalidades aplicadas al sector de la edificación para una mejor comprensión. Se explica cómo se gestiona el consumo de recursos y cuáles son los problemas que aparecen en los proyectos debido a las restricciones; estas restricciones están



relacionadas con dos ideas, por un lado las restricciones en los tiempos de finalización de las actividades y por otro, las correspondientes a la cantidad y la disponibilidad de los recursos para realizar dichas actividades, siendo esta segunda en la que más se insiste en este trabajo.

También se sientan las bases para programar un proyecto cuando nos encontramos ante los problemas que surgen con la limitación de recursos, pues en un mundo real lo normal es que la cantidad de recursos a utilizar esté restringida. Se establecen dos tipos generales o estándares de problemas derivados del consumo de recursos, los de Nivelación de Recursos y los de Programación de Proyectos con Recursos Limitados, o Resource Constrained Project Scheduling Problems, a partir de ahora RCPSP. Se estudia cómo se pueden reconocer las diferencias entre ellos y se profundiza en los de tipo RCPSP, tratando asuntos como la diferencia entre programación y secuenciación de actividades, los modelos matemáticos o la complejidad matemática.

Ya en el capítulo 4 se clasifican los problemas RCPSP de acuerdo a un criterio establecido por (Demeulemeester & Herroelen, 2002), basado en otros semejantes que (Graham, et al., 1979), (Blazewicz, et al., 1983), (Lawler, et al., 1989) y (Brucker, 1995) utilizaron para la programación de máquinas y comparándola con otras notaciones diferentes que se encuentran en la literatura. Este sistema sirve para unificar criterios entre los investigadores a la hora de tratar las numerosísimas variedades y ramificaciones que surgen a partir del problema estándar RCPSP. Además se estudian los métodos exactos (deterministas) y los aproximados (algoritmos heurísticos, metaheurísticos e hiperheurísticos) que se utilizan en la resolución del RCPSP. Los métodos deterministas son menos adecuados porque normalmente no son capaces de encontrar soluciones a los problemas complejos en un tiempo adecuado. Se desarrollan y se analizan algunos tipos de algoritmos.

En los capítulos 5 y 6 se trata la Teoría de la decisión, su relación con la imprecisión en la estimación de valores, cuál es la diferencia entre imprecisión e incertidumbre y como se pueden adaptar los valores imprecisos a la limitación de recursos, ya que en la planificación de proyectos nunca se puede tener la certeza de que está se va a cumplir al 100%, por consiguiente siempre existirán elementos ajenos o no al proyecto que ocasionarán cambios en ella. A partir de esto, se realiza una introducción a la lógica Fuzzy en dónde se explica su filosofía, los conceptos generales que la sustentan y cómo se opera con la aritmética difusa, haciéndose un repaso sobre el estado del arte en la limitación de recursos con valores imprecisos.

Se ha comprobado que hasta la fecha en los problemas RCPSP se ha utilizado la lógica fuzzy aplicando la imprecisión a las duraciones de las tareas sin estudiar que ocurre en el caso de datos difusos en las cargas y/o disponibilidades de los recursos. En el capítulo 7 se aplican los conocimientos tratados en el trabajo exponiendo el estudio de una instancia sacada de la PSPLIB (ver apartado 7.1 La PSPLIB – “Project Scheduling Library”-Biblioteca de Problemas Programación), utilizándola para 4 supuestos y discutiendo los resultados obtenidos:

1. Solución al problema de forma clásica, suponiendo solamente las relaciones de precedencia entre las distintas actividades.
2. Solución nuevamente con datos nítidos contemplando además el cumplimiento de las capacidades máximas de los recursos.





3. Solución teniendo en cuenta las restricciones de las actividades y los recursos, igual que en supuesto 2º, pero considerando como difusos los datos del consumo de recursos de cada una de las actividades por periodo de tiempo.
4. Solución con los mismos condicionantes que el caso 3º, valorando también que la disponibilidad máxima por cada tipo de recurso y periodo de tiempo es difusa.

Finalmente en el capítulo 8 se enumeran las distintas conclusiones a las que lleva el desarrollo de este proyecto, señalando como se puede mejorar y optimizar la programación y la eficiencia de los recursos que son necesarios para realizar las distintas tareas de un proyecto de edificación. De igual forma, se enumeran una serie de posibles investigaciones a las que el trabajo da pie y que se pueden utilizar para el desarrollo posterior de una tesis doctoral.





## 2. Antecedentes Históricos de la Gestión y Limitación de Recursos

“Las soluciones a los problemas de la producción son esencialmente de orden técnico: los regímenes políticos pueden cambiar, las necesidades permanecen”.

*Jean Chevalier (1906-1993) Historiador francés*

Estamos inmersos en un mundo en donde las necesidades son cada vez mayores. El estado del bienestar exige soluciones prácticas fundamentadas en la teoría y en el estudio, pero simplemente ya no es suficiente con fabricar o construir bien, se exige optimizar la eficiencia de la producción. La revolución técnico-científica junto con el desarrollo informático ha supuesto un cambio en las prioridades de los sectores económicos más importantes que mantienen los pilares de la sociedad, centrándose actualmente en la gestión de recursos para solucionar los problemas de producción.

En un principio, los recursos no suponían en la mayoría de las ocasiones una dificultad que tuviese mucha repercusión durante la ejecución de un proyecto. Los **recursos materiales** no se agotaban; su cantidad no era un obstáculo importante, más bien las dificultades aparecían a la hora de manipularlos, transportarlos o colocarlos en su lugar definitivo debido a las rudimentarias técnicas utilizadas. Algo parecido ocurría con los **recursos humanos**; en una economía basada en el trabajo artesanal, la mano de obra se podía sustituir con relativa facilidad, era barata y los tiempos de ejecución de los trabajos no apremiaban.

Pero con el transcurso del tiempo la situación fue cambiando, fueron apareciendo los criterios de control y vigilancia sobre los productos y los trabajadores, comenzando a aplicarse la selección y adiestramiento de los trabajadores y estableciéndose las jornadas laborales.

Ya Leonardo da Vinci en el **siglo XVI**, deja el primer testimonio escrito sobre la medición del trabajo e inicia el camino para el análisis y estudio de la división del trabajo. En 1801 Matthew R. Boulton, describe el concepto de puesto de trabajo, provocando que posteriormente estudiosos como Babbage, Taylor, Fayol, Gantt, Bunker, etc., abran toda una ciencia alrededor del trabajo tratando asuntos como la especialización, la incentivación, la psicología del trabajo, la secuenciación de tiempos, etc. (Carrascosa Martínez, et al., 2007)

Se pasa de una economía basada en el trabajo artesanal y manual, a una economía de manufacturación. Aparecen la mecanización y las cadenas de producción. La mano de obra va adquiriendo cada vez mayor relevancia, temas como el rendimiento, los tiempos de producción y los costes se convierten en importantísimos para la fabricación, en resumen, se trata de ser muy eficientes produciendo más que la competencia con los mínimos recursos posibles.

Desde que en **1895** aparecen los estudios de FREDERICK WINSLOW TAYLOR (1856-1915), considerado "el padre" de la ORGANIZACIÓN entendida como ciencia, incluso las guerras van contribuyendo al aumento del potencial de la producción, a la aceleración del progreso y los cambios al tener que agudizar el ingenio en la búsqueda de estrategias y en la investigación de operaciones tácticas y de nuevas armas. Por ejemplo, el origen del **método de la simulación**



**de Monte Carlo** se encuentra en las investigaciones llevadas a cabo por Stanislaw Marcin Ulam (1909-1984) y Jhon Von Neuman (1903-1957) en el Proyecto Manhattan cuyo objetivo fue el desarrollo de la bomba atómica durante la Segunda Guerra Mundial. La simulación de Montecarlo es una técnica cuantitativa que mediante la estadística e imitando el comportamiento aleatorio de sistemas reales no dinámicos, permite obtener los resultados de las decisiones que se toman y analizar el riesgo que conllevan las mismas.

En **1956** Morgan Walker y James E. Kelley presentan la teoría de las técnicas de redes y establecen el Método del Camino Crítico o “Critical Path Method” (en adelante CPM). La industria ha alcanzado un gran desarrollo.

En el diagrama de una red CPM, que se conoce como Red de Actividades en Flecha, las prelacones<sup>2</sup> de las actividades se presentan en los arcos dirigidos entre los sucesos o nodos sobre los que acometen o de los que salen las flechas. En los nodos se reflejan los tiempos más pronto y más tarde permitidos tras la ejecución de las actividades. Con estos tiempos y mediante algoritmos de cálculo se obtiene la duración del proyecto.

Simultáneamente al método determinista<sup>3</sup> del CPM, van surgiendo otras herramientas relacionadas con la organización, programación y control de proyectos. Una muestra de estos métodos es la siguiente:

- La técnica de **Evaluación y Revisión de Programa** o “Program Evaluation and Review Technique (**PERT**) ”: planteada en **1957** por La Booz, Allen & Hamilton y Charles E.Clark para el proyecto “Polaris” (proyecto de submarinos atómicos de la Marina de Estados Unidos) en el cual se debía gestionar a más de 9.000 empresas (en contratación directa o subcontratadas). Gracias a esta técnica probabilística, la duración del proyecto que inicialmente estaba prevista en 5 años quedó sustancialmente reducida a 2 años de trabajo.  
PERT permite estimar la varianza de la duración de un proyecto y calcular la probabilidad de terminar el proyecto entre unas fechas objetivo. La duración de las tareas con incertidumbre, se modela a través de variables aleatorias con funciones de distribución tipo beta que además serán normales<sup>4</sup>. La representación gráfica del proyecto, al igual que en CPM, se hace mediante Redes de Actividades en Flecha.
- La Estructura Desagregada de Tareas (EDT) o Work Breakdown Structure (WBS) también fue creada y presentada como parte del proyecto “Polaris” en **1962**. En la EDT se organiza el proyecto desglosándolo en tantos niveles jerárquicos como sean necesarios hasta llegar al nivel de máximo detalle representado por las tareas o

<sup>2</sup> Las prelacones o relaciones de precedencia de las actividades determinan que actividades deben terminarse para poder iniciar otras, es decir cuáles son las tareas antecesoras y las sucesoras.

<sup>3</sup> Mediante el CPM se hace una estimación determinista de las duraciones de las tareas basada en la experiencia de expertos o en bases de datos surgidas a partir de proyectos similares. Con el método PERT, en contraposición, cuando exista una incertidumbre en las duraciones de las actividades por carecer de datos o del asesoramiento de expertos se procederá a realizar una estimación probabilística.

<sup>4</sup> Distribución normal: La distribución de probabilidad continua de una variable aleatoria es una función que asigna a cada suceso definido sobre dicha variable la probabilidad de que este suceso ocurra. Además se dirá que es normal si la gráfica es simétrica respecto de un parámetro estadístico, o lo que es lo mismo, si tiene forma de campana de Gauss.



actividades del proyecto. La EDT permite explicar el proyecto de forma que se evite la superposición de objetivos y de esfuerzos.

- La técnica de los **potenciales** o de **ROY**: creada y desarrollada por B. Roy entre **1958 y 1960**, paralelamente al método del camino crítico. Hoy en día está prácticamente en desuso. Este método se caracteriza por el desarrollo de un diagrama en donde las actividades se muestran en nodos o sucesos unidos por arcos dirigidos que representan las prelación de estas, es decir, el proyecto se representa mediante Redes de Actividades en Nodo. Con Roy las relaciones de precedencia pueden ser positivas o negativas; una actividad puede empezar antes de que termine otra, por tanto se permite el solapamiento de actividades o puede iniciarse después de cierto tiempo de finalizar su antecesora. En consecuencia, al poder representar con más facilidad las distintas situaciones que pueden darse entre las tareas, es un método que permite reflejar mejor la realidad del proyecto.
- El Diagrama de Precedencias o “Precedence Diagramming Method (**PDM**): creado por Jhon W. Fondal en **1961** y desarrollado después por IBM, es sin duda el método más utilizado. El PDM matemáticamente es un grafo acíclico y sin bucles. En esta técnica las actividades se representan en los nodos o sucesos al igual que en la técnica de potenciales, añadiendo a las relaciones entre tareas de tipo final-inicio<sup>5</sup>, las de inicio-inicio, inicio-final y final-final. Al igual que en el método de Roy, las prelación entre tareas pueden ser negativas o positivas.

A partir del año **1962** se abre una nueva etapa para la resolución de problemas de programación de proyectos, además de las precedencias entre las actividades se introducen las **limitaciones de recursos**. El objetivo es minimizar el *makespan*<sup>6</sup> del proyecto suponiendo que inicialmente se dispone de los recursos suficientes para desarrollar las actividades. Como se verá más adelante, la cantidad de recursos podrá o no estar limitada en función del tipo de recurso que se utilice.

Así mismo, en la década de los **años 60**, desde el punto de vista de la Investigación Operativa y las Matemáticas Aplicadas, empiezan a desarrollarse los **Algoritmos heurísticos** de nivelación y asignación de recursos que estudian los problemas que aparecen cuando los recursos del proyecto están limitados.

En **1971**, el Teorema de Cook supuso un avance en la Teoría de la Complejidad (la Complejidad se trata con más detalle en el [apartado 3.5](#)) al probar que existen Problemas NP-completos<sup>7</sup>. Desde entonces se ha comprobado que miles de problemas son de este tipo. Stephen Cook demostró que si tenemos un problema NP, entonces existe y conocemos una Máquina de Turing no Determinista que lo resuelve en un tiempo polinomial conocido.

<sup>5</sup> Si una tarea  $j$  tiene una relación de final-inicio con una actividad antecesora  $i$ , la actividad  $j$  no podrá iniciarse antes de que haya finalizado  $i$ . Por el mismo motivo en una relación inicio-inicio el comienzo de  $j$  dependerá del inicio de  $i$ ; en una de tipo inicio-final, el final de  $j$  dependerá del inicio de  $i$ ; y en el caso de final-final, la terminación de  $j$  dependerá de acabar  $i$ .

<sup>6</sup> Makespan: Tiempo total en que todas las actividades de un proyecto completan su ejecución.

<sup>7</sup> NP: Nondeterministic Polynomial Time o Tiempo Polinomial No Determinístico.



Dentro de la gestión de proyectos, todo el rápido desarrollo que se aprecia en los **últimos 30 años** no es fruto de la casualidad o del azar, si no que coincide con los avances técnicos en informática y la gran popularidad que estos han alcanzado. Las mejoras en el software y hardware hacen que hoy en día se puedan resolver problemas de gestión en segundos cuando hasta hace poco hubiesen tardado años en resolverse.

Según el Centro de Investigación Operativa de la Universidad Miguel Hernández ([Instituto Universitario Investigación, 2012](#)) con la llegada de los PC<sup>8</sup>, el primero fue el Apple II en **1977**, se crea una interacción entre persona-computador que en términos generales se ve favorecida por el incremento de la potencia de las máquinas y el desarrollo del software, permitiendo que para el gestor del proyecto sea cada vez más fácil la utilización de técnicas que ayuden en la optimización y la gestión del proyecto.

En **1980** se realiza la primera certificación en gestión de proyectos profesional (Project Management Professional) ofrecida por el Project Management Institute (PMI).

El Programa **Harvard Project Manager** en **1983** ([Medina Ramón, 2008](#)), es el primer software para la gestión de proyectos. Sienta precedente para todos los que surgen más tarde.

A partir de esta década de los **años 80**, expertos investigadores como Pristker, R.H. Möhring, Demeulemester, Brucker, Álvarez-Valdés, etc. mantienen abierta una dinámica de propuestas a los problemas cuyo objetivo fundamental es la optimización de los recursos del proyecto, haciendo uso de nuevas herramientas como son los algoritmos metaheurísticos e hiperheurísticos o la lógica borrosa.

Recogiendo de manera resumida los antecedentes históricos, en el siguiente cuadro se hace una cronología escueta con las fechas, las personalidades o los hitos importantes que han marcado la gestión de proyectos y la limitación de recursos:

---

<sup>8</sup> PC: Personal Computers u Ordenadores Personales.



FECHA, PERSONALIDAD		APORTACIÓN
4000 a.C		En Egipto la organización del trabajo está presente en las construcciones que han llegado hasta nuestros días. Aparecen los primeros ingenieros de la Organización.
5º V a.C		En la Antigua Grecia el aumento poblacional llevó a la agricultura intensiva y su organización. En la ciudad los artesanos formaban equipos y talleres, el trabajo era controlado por los escribas que apuntaban todo lo referente a sus actividades.
5º I		Durante el Imperio Romano el trabajo estaba organizado en gremios. Ingenieros y arquitectos viajaban junto con las legiones romanas realizando grandes obras públicas y de ingeniería; su planificación permitía que las construcciones fuesen realizadas rápidamente y con calidad.
1240	WALTER OF HENLEY	Establece criterios de control para la vigilancia de tierras, ganado y trabajadores; selección y adiestramiento de trabajadores, establecimiento del trabajo mínimo por jornada, y utilización de recursos para arar la tierra.
1452-1519	LEONARDO DA VINCI	Primer testimonio escrito sobre la medición del trabajo previamente descompuesto en partes. Primer paso en la división del trabajo para su análisis y estudio.
1760	JEAN PERRONET	Descripción completa del ciclo de producción de una fábrica de alfileres, quedando reflejado el cálculo de los tiempos de las distintas tareas y partes del proceso de producción.
1801	JAMES WATT y MATTHEW R. BOULTON	Métodos de trabajo y tiempos, introduciendo el concepto del puesto de trabajo.
5º XIX	CHARLES BABBAGE (1832)	La división y especialización del trabajo según criterios en cuanto a la forma de realizar el trabajo o a los requerimientos profesionales del trabajador para su realización.
		Criterios de incentivación y bonificación al trabajador en función de la producción y calidad. Se cronometra el tiempo para establecer un control de rendimientos.
1856-1915	FREDERICK W. TAYLOR	Inicia el estudio del trabajo como Ciencia. Nueva visión de los problemas, el trabajador no siempre es la causa (errores empresa, desconocimiento tiempos, métodos de trabajo, exigencia rendimientos, maquinaria obsoleta ...). Estudió rendimientos y tiempos.
1891	FREDERICK A. HALSEY	Diseña plan para aumentar la productividad a partir de los costes de la mano de obra.
1861-1919	HENRY L. GANTT	Trabajó con Taylor. Creador diagrama Gantt. Introduce la psicología en las relaciones con el trabajador.
1868-1924	FRANK y LILLIAN GILBRETH	Convinan conocimientos psicológicos y de ingeniería. Estudian los movimientos, ayudándose de filmaciones y cronómetros de manera que podían analizar la secuencia de movimientos de tiempos. Establecen normas de Ergonomía.
1945		Se construye la primera computadora digital electrónica en Universidad de Pennsylvania.
1948-1975	EIJI TOYODA y TAICHI OHNO	Sistema Toyota. Lean production.
1956	Morgan Walker y James E. Kelley	Teoría de las técnicas de redes, estableciendo las bases para el método de la ruta crítica.
1957	Booz Allen & Hamilton y Charles E. Clark	Crean la Técnica de Evaluación y Revisión de Programa (Program Evaluation and Review Technique o PERT).
1961	JOHN W. FONDAHL	Diagrama de Precedencias.
1971	STEPHEN COOK	Teoría de la Complejidad.
1977	STEVE WOZNIAC	El Apple II fue el primer microcomputador producido a gran escala.
1980	PROJECT MANAGEMENT INSTITUTE	Primera certificación "Project Management Professional"
1984	ELIYAHU M. GOLDRATT	Publica "La Meta". Teoría de las limitaciones o restricciones. Tecnología de Producción Optimizada.
1989	MARK L. SPEARMAN	Sistema Kanban. Optimización de las cadenas de suministro.
1990		Lenguajes de modelamiento. Computación de gran escala. Desarrollo de herramientas de optimización y simulación.
2000...		Desarrollo software y hardware. Resolución problemas complejos. Reingeniería de procesos. Sistemas de decisiones automáticas. Necesidad de coordinar el uso eficiente de recursos en cadenas de suministros.

Figura 2: Cronología de Gestión y Limitación de Recursos.

Fuente: elaboración propia y (González Rodríguez, s.f.)







### 3. El Problema de la Programación de Proyectos con Recursos Limitados

#### 3.1. El Proyecto y su Gestión.

En el [apartado 1](#) se explicó de forma muy general el concepto de proyecto, seguidamente se profundiza en las etapas por las que este deber ir pasando y como dirigirlas. En primer lugar debemos definir el concepto de “**ciclo de vida**” de un proyecto. Se denomina ciclo de vida del proyecto al periodo transcurrido desde el inicio de un proyecto hasta su finalización. Todo proyecto empieza a partir de una idea que se desarrolla para conseguir unos objetivos; lo habitual es que predominen los intereses del promotor de dicho proyecto; estos intereses no necesariamente tienen que ser siempre o exclusivamente de carácter económico, también pueden ser sociales, técnicos, políticos, etc. El ciclo de vida del proyecto de forma esquemática consta de las siguientes fases (más adelante se explica cada una de ellas):

- Fase de concepto.
- Fase de definición.
- Fase de planificación.
- Fase de ejecución.
- Fase de terminación y cierre.

Los nombres de estas etapas pueden diferir en la literatura, aunque en esencia son similares, dependen del tipo de producto que se obtenga, del tipo de empresa o del sector a quién va dirigido. Por ejemplo en el sector académico y de investigación ([Klein, 2000](#)) las nombra como se ha indicado en el párrafo anterior; en ([Demeulemeester & Herroelen, 2002](#)) aparecen las fases de concepción, definición, planificación, programación, control y terminación; de acuerdo a ([Kerzner, 2009](#)) las fases serían las de concepción, planificación, pruebas, implementación y cierre. Para proyectos informáticos ([Cervantes Posada, 2010](#)), considera cuatro etapas, planificación, programación, control y terminación; según ([Montoya Molina, s.f.](#)) serían concepción, planificación, implementación y finalización; de acuerdo a ([Navascués Fernández, s.f.](#)) el ciclo de vida total puede descomponerse en fase previa o de programación, fase ejecución y fase posterior o de terminación. En proyectos de construcción, ([Uher, 2003](#)) considera las fases de concepción, diseño, construcción y puesta en marcha; para ([Pellicer Armiñana & Yepes Piqueras, 2005](#)) el esquema básico del proceso de un proyecto de construcción consta del establecimiento del objeto (la declaración de la detección de un problema, necesidad u oportunidad), al que seguiría una decisión de resolverlo, la planificación inicial, la fase de diseño, la fase de construcción y la fase de uso y explotación.

Hay que destacar como significativo que en el último ejemplo indicado, para el profesor doctor Eugenio Pellicer de la Universidad Politécnica de Valencia ([Pellicer Armiñana & Yepes Piqueras, 2005](#)), el proyecto continúa más allá de la terminación del producto, ya que también considera como parte del proceso del proyecto su uso y explotación, extendiéndose por tanto a lo que sería el ciclo de vida de la construcción o edificación. Este planteamiento está mezclando el concepto de ciclo de vida del proyecto con el de ciclo de vida del producto, puesto que el proyecto no se daría por terminado hasta que se hubiese alcanzado la etapa de obsolescencia, retirada, falta de utilidad o de uso del producto. En consecuencia, se podría



decir que durante el ciclo de vida del producto podrían intervenir uno o más proyectos, los cuales tendrían como objetivo distintas facetas del mismo.

Para la realización de un proyecto por ejemplo de edificación, lo primero a fijar sería el objetivo y el objeto (concepto y definición) del proyecto que se pretende alcanzar. En el momento de la planificación inicial se establecerán las tareas necesarias para la ejecución del edificio, así como la cantidad de mano de obra, de materiales y los equipos necesarios, es decir, se asignarán los **recursos** precisos para la ejecución de los trabajos, tras lo cual se habrá definido qué se ha de hacer y cómo.

Dentro de la planificación, el paso siguiente será la estimación de la duración de los trabajos basada en los datos y entradas anteriores. Determinar las duraciones de las actividades no es una labor fácil, pues requiere de la propia experiencia de la persona que la realiza, o bien de disponer de una base de datos desarrollada por la organización a partir de los conocimientos prácticos obtenidos tras la actuación en otros proyectos similares. Según (Klein, 2000) la consideración de la duración del proyecto también puede llevarse a cabo mediante la aplicación de técnicas forecasting<sup>9</sup>; a través de métodos de pronósticos cualitativos y cuantitativos, y analizando las previsiones de la demanda del producto se conseguirá una planificación más eficiente y exacta del proceso.

Seguidamente se programará el proyecto. La programación es la acción de ordenar en el tiempo las acciones necesarias para realizar un proyecto, marcando las prioridades y relaciones entre ellas, creando un calendario. Requiere detallar las fechas de inicio y terminación, los hitos a alcanzar y la probabilidad de cumplimiento de los plazos programados. Con toda seguridad, al desarrollar la programación y considerar su optimización, se tendrá que volver atrás para retocar la planificación, reasignándose nuevamente los recursos y haciéndose los ajustes sucesivos hasta que ambas encajen adecuadamente en los objetivos a conseguir.

Más tarde, el proyecto se ejecutará de acuerdo con la planificación y programación previstas. La ejecución se controlará para poder aplicar medidas correctoras en el caso de que se detecten desviaciones, garantizando de esta manera que el proyecto se desarrolle correctamente. En un proyecto no se debe permitir que se violen los plazos o presupuestos, ni se debe permitir poner en peligro su contenido; para (Demeulemeester & Herroelen, 2002) el beneficio final de la gestión de proyectos en general, y de la programación de proyectos, en particular, es tener un cliente satisfecho.

La terminación y cierre, será la última etapa del proyecto y tan importante como el inicio del mismo. Un buen cierre del proyecto servirá de gran ayuda al permitir una retroalimentación que se podrá aplicar en futuros proyectos semejantes al finalizado, puesto que se estudiarán las diferencias entre lo planificado y programado, y el resultado final conseguido. Así mismo, se analizará si se han conseguido o no los objetivos, en qué se ha fallado (excesos en el presupuesto, especificaciones del proyecto complicadas o hitos no alcanzados) y cuáles han sido las causas de estos errores.

Además de que se lleve a término cada una de las fases del ciclo de vida del proyecto, para conseguir que sea todo un éxito se deberá hacer una correcta gestión del mismo. Básicamente

---

<sup>9</sup> Forecasting o pronóstico, es la estimación y el análisis de la demanda futura para un producto o servicio, utilizando entradas de datos históricos o estimaciones de marketing. Asume que el futuro puede ser identificado y reconocido.



se puede afirmar, como comenta (Hekimoglu, 2007), que la **gestión** de un **proyecto** se fundamenta en el desarrollo de tres aspectos, la **planificación**, la **programación** y el **control**. Para (Cervantes Posada, 2010) la gestión de proyectos es la disciplina de organizar y administrar recursos de manera tal que se pueda culminar todo el trabajo requerido en el proyecto dentro del alcance, el tiempo, y los costes definidos.

Antes de iniciar la ejecución del proyecto ha de prepararse y se ha de comprender su contenido, pero no será suficiente con esto, sino que también durante la fase de realización hay que permanecer alerta para detectar cualquier incidente que pueda dar al traste con las pretensiones del proyecto, cuanto mayor sea el control, más rápida será la capacidad de reacción ante un problema, más pronto se solucionará y menos consecuencias tendrá en el producto final.

La gestión requiere pues de la toma de decisiones que son importantes para la empresa promotora del proyecto, además estas no han de ser genéricas ya que deben responder a las exigencias particulares de cada caso. Y ¿de qué herramientas se valen las organizaciones para conseguir que las medidas o decisiones sean las adecuadas?, pues además de la propia experiencia, de las que ofrecen disciplinas como la Informática o las Matemáticas, en concreto la Investigación Operativa. Hay que remarcar que estas herramientas no se utilizan de manera aislada, sino que interactúan unas con otras, por esto los avances científicos y tecnológicos de estas ciencias han repercutido (y continúan haciéndolo) de manera relevante en el arte de la Organización y Gestión de proyectos.

El hecho de que la **gestión** de actividades ocupe un papel **fundamental** dentro de las compañías, se debe a la relación que tiene con el ideal básico de todo proyecto, el obtener un producto dentro del **coste**, **tiempo** y **calidad** exigidos. Para alcanzar estos tres objetivos se han de asignar a las distintas tareas **la cantidad y el tipo de recursos necesarios**, estableciéndose unos intervalos o periodos para que las actividades puedan ser ejecutadas dentro de los plazos o hitos a los que la organización se ha comprometido con el cliente.

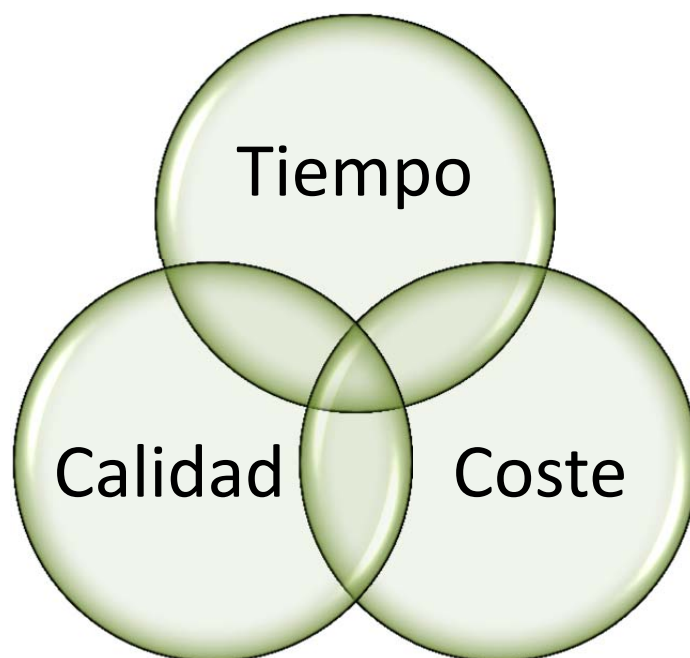


Figura 3: Objetivos básicos en la gestión de proyectos.



Durante la gestión de un proyecto nos encontraremos con dos tipos de restricciones a considerar y el no tenerlas en cuenta puede hacer que el proyecto se retrase o que incremente su coste final. Por un lado tendremos las restricciones en relación al tiempo, es decir, la relaciones de precedencia entre las actividades obligan a que éstas terminen o empiecen en unos instantes determinados de manera que se respeten las prioridades entre ellas. Y por otra parte, la existencia de las restricciones de recursos en relación a su tipología o cantidad deja claro que habrá una limitación que también impactará en el proyecto en función de la escasez o abundancia que se tenga de recursos. Como dice (Brucker, 1995) para resolver un problema de programación hay que satisfacer las restricciones.

Los proyectos consumen una cantidad de recursos diaria para poder realizar las tareas de acuerdo a la planificación (Ponz Tienda, 2011), lo habitual es que esa cantidad de recursos no sea inagotable, es decir, los **recursos** están **limitados** y tienen una disponibilidad máxima en cada periodo; si esto no se considera así se generarán conflictos durante la ejecución del proyecto, por tanto desde el momento de su planificación y programación hay que estudiar y solucionar los problemas que surgen en los **proyectos con recursos limitados**.

Por supuesto la asignación de recursos y sus restricciones, generarán problemas que seguramente no tendrán una respuesta clara e inmediata, ni serán sencillos ni rápidos a la hora de resolverse. Más adelante, en el punto 3.5, se explica en que consiste la complejidad de un problema hablando en términos matemáticos.

### 3.2. Concepto y Tipos de Recursos

Por definición, se dice que los **recursos** son el conjunto de elementos (de cualquier tipo) disponibles para resolver una necesidad o llevar a cabo una empresa, es decir, son el medio o instrumento que permite mediante su uso conseguir un producto deseado. Por ejemplo, en el caso de un proyecto de edificación, los recursos serían la mano de obra u operarios de los distintos oficios que intervienen en la ejecución del edificio; todos aquellos materiales o equipos que forman las distintas unidades de obra (hormigón, ladrillos, revestimientos, calderas, etc.); las herramientas, la maquinaria, y los medios auxiliares utilizados por los operarios para la colocación o la puesta en servicio de esos materiales y equipos.

Como se ha indicado en el apartado 3.1, es durante la fase de planificación cuando debe decidirse que clases de recursos se van a utilizar. En función de ello, se establecerá cuál es su coeficiente de participación<sup>10</sup> y cuántos serán necesarios para realizar las distintas actividades del proyecto.

Al buscar en la literatura científica, se encuentran diferentes categorías y tipos de recursos según su naturaleza y el campo en el que se desarrolle el proyecto (finanzas, construcción, sostenibilidad, programación, etc.), el origen de los recursos (humanos o materiales), la posibilidad o no de reutilización que tengan, la compatibilidad entre ellos, la divisibilidad, etc.

<sup>10</sup> Coeficiente de participación: la cantidad de recursos de un tipo  $r$ , que se necesita para realizar una determinada unidad producción. Por ejemplo, si se precisa la cantidad de 0.50 horas de oficial para realizar 1 m<sup>2</sup> de tabicón de ladrillo hueco de 7 cm. de espesor, es necesario un coeficiente de participación de oficial de 0.50 h/m<sup>2</sup>.



Según las fuentes consultadas [Para estudiar en profundidad los tipos y la enumeración de recursos que se hace en la literatura se aconseja examinar (Ponz Tienda, 2010), (Demeulemeester & Herroelen, 2002), (Klein, 2000), (Ballestín González, 2002), (Alcaraz Soria & Maroto Álvarez, 2009), (Agirre Pérez, 2007), (Brucker & Knust, 2012), (Klastorin, 2010), (Areces, et al., 2011), (Hartmann & Briskorn, 2010), (Kolisch & Sprecher, 1996)], los recursos se pueden clasificar como sigue:

- **Categorías básicas de Recursos \_**

- **Recursos Renovables:** son los utilizados para realizar más de un proyecto, o en más de una ocasión dentro del mismo proyecto, no se consumen con el uso. Una muestra de este tipo de recursos pueden ser la mano de obra, la maquinaria, los equipos, etc.

Se dice que el uso de los recursos totales renovables está disponible en cantidades limitadas y constantes en cada periodo de la planificación total, o lo que es lo mismo, cada recurso renovable está disponible en una cierta cantidad constante por cada unidad de tiempo y su utilización no debe exceder esa cantidad en ninguna de esas unidades; la cantidad es renovable de un periodo a otro quedando restringido el número de recursos que se pueden utilizar por período; tienen una tasa de utilización por cada unidad de tiempo.

Esa tasa puede ser constante o variable para cada unidad de tiempo, por ejemplo, se puede decir que se dispone de un número constante de 4 oficiales por día, para realizar durante 3 días la tabiquería de una vivienda; o bien que se dispone de un número variable de oficiales en función del día, 4 oficiales para el primer día, 5 oficiales el segundo día y 2 oficiales el último día.

- **Recursos No Renovables:** son empleados en una única ocasión dentro de las tareas del proyecto, formando parte de los elementos que componen el producto final, es decir, se consumen durante la ejecución a medida que se van utilizando. Una muestra clara de este tipo de recursos son materiales como el acero, el hormigón, etc.

Se consideran que no son agotables, pero pueden estar restringidos a una determinada cantidad total para la completa realización del proyecto.

Por ejemplo, la cantidad de ladrillos para hacer una tabiquería es inagotable, se pueden pedir a uno o a varios fabricantes todas las unidades de ladrillos que se quieran, pero en el presupuesto total del proyecto se ha previsto que sólo dispone de 5.000 € para la adquisición de los ladrillos, si cada ladrillo tiene un precio de 0.10 €, se dispondrá de una cantidad total 50.000 ladrillos que se irán consumiendo a medida que se van usando en cada unidad de tiempo.

- **Recursos Doblemente Restringidos:** se ven limitados en cada periodo de tiempo y para todo el proyecto, por tanto son fungibles y además tienen un consumo máximo permitido por periodo.

Son una combinación de recursos renovables y no renovables, es decir, se restringen tanto en su disponibilidad por tiempo como los renovables y en su total acumulado para la totalidad del proyecto como los no renovables.



Un ejemplo de este tipo de recurso sería tener un presupuesto total de 5.000 € para comprar ladrillos, pero además que sólo se pueda gastar un máximo de 150 € en ladrillos por día, en este caso los ladrillos estarían doblemente restringidos.

### ● Otras categorías de Recursos \_

- **Recursos Parcialmente Renovables o Parcialmente No Renovables:** son una generalización de los recursos renovables y no renovables, es decir, se dispone de un recurso parcialmente renovable o parcialmente no renovable en función de si cada uno de ellos puede ser procesado dentro de un subconjunto de periodos establecidos en el horizonte de la planificación y las actividades sólo lo consumen si son utilizados en esos periodos. Este tipo de recursos son muy útiles para resolver problemas de secuenciación de tareas.

Como ejemplo de recurso parcialmente renovable, tendríamos un oficial de albañilería que trabajase de lunes a viernes y luego trabajase el sábado o el domingo, pero nunca los dos días; el sábado y el domingo formarían un único periodo de tiempo con disponibilidad de 1 oficial.

- **Recursos continuamente o discretamente divisibles:** desde el punto de vista de la divisibilidad, existen dos tipos de recursos, los recursos que pueden ser continuamente divisibles (la energía eléctrica, el consumo de agua, etc), o los discretamente divisibles, es decir, que se pueden asignar a las tareas en cantidades discretas a partir de un conjunto finito de asignaciones dado, como por ejemplo la mano de obra o la maquinaria.
- **Recursos interrumpibles o no-interrumpibles (preemptible, non-preemptible):** Un recurso es interrumpible (preemptible) cuando puede ser detenido para que un recurso de mayor prioridad pueda ser usado. En consecuencia, puede ser retirado de la tarea que se está procesando y puede ser asignado a otra y luego puede volver a la tarea anterior. Si no se puede paralizar será no interrumpible (non-preemptible).
- **Recursos Dedicados:** los que solamente pueden ser asignados a una única actividad al mismo tiempo, serán recursos renovables con disponibilidad por periodo. Como ejemplo se puede pensar en una pantalladora o máquina que hace los muros pantalla para la cimentación especial de un edificio, sólo podrá estar trabajando en un tramo de muro pantalla y no podrá pasar a otro hasta que no termine el primero.
- **Recursos Acumulativos:** se pueden almacenar, se pueden agotar y se pueden reponer con el tiempo. Un ejemplo sería que durante la ejecución de la tabiquería de un edificio se agotasen los sacos de cemento y se retirasen nuevos sacos de la zona de almacén de la obra; estos sacos serían recursos acumulativos.
- **Recursos Espaciales:** no son requeridos por una sola actividad, sino por un grupo de actividades. Un recurso espacial es ocupado desde que es usado



por primera vez por una actividad del grupo y hasta que la última actividad del grupo ha terminado.

- **Recursos con disponibilidad que varía con el tiempo:** la disponibilidad de los recursos no tiene porqué ser constante, puede aumentar o disminuir. Como ejemplo, dada una obra se incrementará la disponibilidad de recursos si se paraliza una segunda obra y se tienen que trasladar recursos de mano de obra de esta segunda a la primera, o se podrá reducir si llega un periodo de vacaciones o hay bajas laborales entre los operarios.

### 3.3. La Gestión de Recursos

Al gestionar los recursos se determinará el tipo, cuáles y de cuántos se dispone. Se asignarán a todas y cada una de las distintas actividades que forman el proyecto, distribuyéndose para que se obtenga el máximo provecho de ellos de manera que las asociaciones que se hagan entre ellos no interfieran en las prelaciónes entre las tareas.

Tras realizar la planificación del proyecto, teniendo en cuenta las mediciones por unidad de cada una de las actividades y la asignación de los recursos que son necesarios para realizar las distintas actividades, lo mejor será disponer de los recursos de un modo constante, sin grandes altibajos, puesto que:

1. Si se trata de recursos no renovables, es más conveniente recibirlos poco a poco durante la ejecución del proyecto, de acuerdo a las necesidades de los trabajos en que intervienen, para evitar problemas de:

- Suministro.
- Logística y distribución.
- Almacenaje.

2. En el caso de recursos renovables, también es aconsejable que se repartan uniformemente, ya que no es fácil disponer de recursos unos días sí y otros no, por qué para ello, deberían coordinarse entre sí los distintos proyectos a los que estuviesen asignados, lo cual sería extremadamente complicado de gestionar. Además el traslado de recursos de un proyecto a otro, supone siempre unos gastos económicos e incrementos de tiempo adicionales que por supuesto hay que tratar de minimizar.

Mediante la gestión de los recursos, lo que tradicionalmente se ha pretendido es la nivelación de estos y reducir el *makespan* a través de la optimización de los procesos y el uso, restringido o no, de determinados recursos, (Ponz Tienda, 2011) y (Vitoriano Villanueva, 2009). Además, la correcta gestión de los recursos conllevará una anticipación a los problemas; de esta forma estos problemas se verán reducidos tanto en cantidad como en dificultad, consiguiendo así una mejora continua del proyecto.



El gestionar los recursos implicará una revisión de la eficiencia de estos. La **eficiencia** se puede definir en términos de producción como la relación entre los resultados obtenidos (ganancias, objetivos cumplidos, productos, etc.) y los recursos empleados:

$$EFICIENCIA = \frac{PRODUCCIÓN ÚTIL}{PRODUCCIÓN PLANIFICADA}$$

Por ejemplo, si en una obra se ha planificado que un equipo de albañiles puede hacer 100 m<sup>2</sup> de tabiquería al día y tras el control diario se observa que sólo se han ejecutado 70 m<sup>2</sup>, la eficiencia de la producción realizada en ese periodo para este recurso será del 70%.

$$EFICIENCIA = \frac{70}{100} = 0,70 \longrightarrow 70\%$$

El recurso está un 30% por debajo de su capacidad de trabajo. Así pues, el recurso aumentará su eficiencia cuanto más se acerque al 100% y será tanto más eficiente, cuanto más por encima de ese porcentaje se encuentre.

Se deberá cuestionar y estudiar qué ha ocurrido en el ejemplo anterior. Se está hablando de mano de obra, de recursos humanos, entre otras causas podemos encontrar que la cantidad de oficiales y peones que forman el equipo sea insuficiente y se necesita aumentar su número; o bien no están cualificados para el trabajo a desarrollar; tal vez los operarios desconocen la obra y conseguirán alcanzar el objetivo según vayan mejorando su curva de aprendizaje<sup>11</sup> (learning curve); es posible que alguno de ellos no haya ido a trabajar o haya sido asignado a otra tarea; puede haber algún fallo en el suministro de los materiales de la actividad; los medios auxiliares de los que se disponía no eran los adecuados; se han dado inclemencias meteorológicas; etc. Se podrían seguir enumerando múltiples causas posibles, pero lo que está claro es que el fallo en la eficiencia de un recurso no dependerá exclusivamente de si se han relajado más o menos durante la jornada laboral, las causas pueden tener otros orígenes distintos, y efectivamente será necesario establecer los errores y examinar cual ha sido la asignación de recursos con la intención de ir reajustando para mejorarla y optimizarla.

En general tras el análisis de la eficiencia en la asignación de recursos nos encontraremos con tres supuestos posibles; primero, que la asignación realizada sea la correcta; segundo, que se hayan sobre-asignado o lo que es lo mismo, que la carga de trabajo sea excesiva para el periodo de tiempo del que disponen los recursos; tercero, que la asignación de recursos esté por debajo de la capacidad de trabajo que tienen y en consecuencia se están desaprovechando, es decir, que se encuentren infra-asignados. Se actuará en consecuencia en función del caso que se dé. Pero...

- **¿Cómo se puede calcular y representar la carga de recursos asignada a las tareas?**

<sup>11</sup> Curva o línea de aprendizaje: diagrama que refleja la mejora en el grado de éxito obtenido gracias al aprendizaje conseguido con el paso del tiempo al ir trabajando sobre una actividad nueva.





Para asignar los recursos se empezará por estimar el coeficiente de intervención del recurso, es decir, las unidades de recursos necesarias para poder ejecutar una unidad de producción y también cuántas de estas unidades se elaborarán por periodo de tiempo, o lo que es igual, se programarán las unidades de producción que se harán por periodo de manera que el proyecto sea factible, estableciendo la cantidad de recursos por actividad. Se le llamará unidad de producción al máximo desglose de las tareas o actividades que formarán el producto final; por ejemplo serán unidades de producción de un proyecto de construcción de un inmueble 1 m<sup>2</sup> de tabiquería, 1 m<sup>3</sup> de losa de cimentación, 1 m<sup>2</sup> de impermeabilización de cubierta... En el caso de proyectos de construcción, el periodo mínimo de tiempo que debe considerarse es el día; en otros sectores como la industria metalúrgica, se puede llegar a hablar de tiempos en horas, minutos o incluso segundos.

Mediante la planificación de las distintas tareas y el calendario del proyecto, se puede establecer la carga de los recursos para cada periodo. Por consiguiente, la carga de recursos no es más que la cantidad de recursos que intervienen o se consumen cada día. La carga de recursos se puede representar en un gráfico, llamado histograma de cargas. En uno de los ejes (eje X) se representan las unidades de tiempo del proyecto y en el otro (eje Y) las frecuencias o cantidades de los distintos tipos de recursos que intervienen por periodo de tiempo. Gracias a este esquema visual se observará con un simple golpe de vista el perfil de la carga y por tanto, si la suma de recursos por día es más o menos uniforme o constante a lo largo del desarrollo del proyecto y cuando está prevista su finalización. Si para cualquier tipo de recurso, la cantidad de recursos requeridos supera el límite admisible en uno o más periodos, se produce un conflicto de recursos y la programación que había sido determinada no se puede realizar. De acuerdo a [\(Klein, 2000\)](#) en este caso, se puede aplicar una de las siguientes técnicas:

- Se puede adelantar el inicio de aquellas tareas que tengan un tiempo de espera, permitiendo una mayor holgura a las que requieran mayor número de recursos, con lo cual al disponer de más tiempo se necesitarán menos recursos. Se deberá analizar como este aumento de tiempo afecta al presupuesto y si el mismo es admisible.
- Prolongar si es posible la duración de las tareas para suavizar la asignación de recursos y una menor utilización de los que se encuentren sobre-cargados. Esta solución conllevará un aumento en la fecha de terminación del proyecto.
- Subdividir los trabajos en sub-tareas sin alterar significativamente las relaciones entre las tareas.
- Incrementar los recursos en aquellos periodos que lo necesiten y en consecuencia se producirá un aumento de los costes.

Cada una de estas técnicas nos enfrentará a un tipo de problema de programación de proyectos a solventar, como puede ser la limitación de tiempo, la limitación de recursos y la nivelación de recursos. El solucionar estos conflictos de manera que se optimice la relación entre tiempo de ejecución y recursos del proyecto, no es una labor fácil y requerirá del uso de paquetes de software para hallar la mejor solución posible o la cuasi-óptima.



### 3.4. Introducción a la Programación de Proyectos con Recursos Limitados

Desde que surgen los primeros estudios sobre **Programación y Control de Proyectos con Recursos Limitados**, estos van adquiriendo cada vez mayor relevancia, existiendo una extensa literatura que recoge los numerosos tipos de problemas aparecidos a partir de la programación de recursos, en un contexto muy dinámico y de continuo cambio, encontrando a día de hoy cuestiones que continúan abiertas y a la espera de ser solucionadas por la comunidad científica.

Al hablar de Programación en este documento, no nos referimos exclusivamente a la Programación Informática, nos referimos también a la Programación Matemática. Hay que indicar que aunque acaban necesitando la una de la otra y llegan a entremezclarse o confundirse produciéndose una asociación entre ambas, la disciplina de la Programación Matemática es anterior a la Ingeniería Informática.

El problema de la resolución de un sistema lineal se remonta al año 1.826 con Joseph Fourier y Carl Friedrich Gauss. (Martín Martín, 2010) señala que en 1.939 el matemático Leonid Vitalevich Kantorovitch publica “Métodos matemáticos de organización y planificación de la producción”; en este libro se recogen por primera vez un grupo de problemas en los que interviene un gran número de variables y a los que se les aplica la programación lineal para solucionarlos. Cabe recordar que la primera computadora digital es de 1.945 y los ordenadores personales empezaron en los años 80; estas máquinas han acabado siendo tan necesarias que un mundo real sin ellas no sería posible en estos momentos.

Reorganizar fechas, actividades y recursos, tener en cuenta las restricciones y dependencias, reducir costes, la mejora continua, adelantar plazos, optimizar, gestionar, controlar, .... y además todo esto hacerlo a la vez, es un arduo trabajo. No es suficiente con la pericia de un grupo de expertos, es preciso el uso de las técnicas más perfeccionadas que les ayuden a la toma de decisiones, de lo último en hardware y software, de la Investigación Operativa y de las Matemáticas Aplicadas más avanzadas.

Los métodos utilizados para programar van desde los gráficos como pueden ser PERT y CPM, a métodos algebraicos, simplex, heurísticos, metaheurísticos, etc. Según se va avanzando en la potencia de los ordenadores van apareciendo nuevos sistemas que son capaces de solucionar problemas con una gran cantidad de cálculos y en pocos segundos, haciendo impensable que hoy en día estos se puedan resolver manualmente. En concreto, **el consumo de recursos da lugar a dos tipos generales de problemas:**

#### 1. Los Problemas de Nivelación o “Resource Leveling Project (RLP)”.

Con su solución se busca la optimización en la eficiencia del consumo de recursos. Cuando se acepta que en la totalidad del proyecto se ha considerado el **número suficiente de recursos**, pero se quiere suavizar la carga de trabajo compensando los periodos de mayor con los de menor consumo de recursos, o lo que es lo mismo, para **equilibrar la demanda** de recursos en cada momento **sin aumentar el makespan del proyecto**, aparece el Problema de Nivelación de Recursos, en adelante RLP.



El objetivo del RLP es conseguir que el histograma de cargas del proyecto o el reparto diario de la suma de los distintos tipos de recursos, sea lo más uniforme posible; dicho de una manera coloquial, debe presentar las menores crestas o valles posibles (Ponz Tienda, 2011) (Ponz Tienda, 2008) , es decir, las diferencias entre las oscilaciones de las cantidades de recursos en cada periodo de tiempo deben ser mínimas o no existir. Gracias a esta nivelación se consigue mejorar la eficiencia en el consumo de los recursos que intervienen en el proyecto.

Tras lo expuesto, se puede afirmar que obviamente, las **ventajas** de la nivelación de recursos al hacer que el flujo de trabajo sea lo más uniforme posible son:

- Se suaviza la carga de los recursos, favoreciendo que no se produzca un agotamiento de los mismos en un momento puntual.
- Desde el punto de vista financiero, los balances al final de cada periodo estarán más nivelados y por tanto, las previsiones de pagos serán más similares.

## 2. Los Problemas de Asignación de tareas con un uso limitado de recursos.

También llamados **Problemas de la Secuenciación de Proyectos con Recursos Limitados** o "**Resource Constrained Project Scheduling Problem (RCPS)**", en adelante RCPS, se analiza ampliamente en el [apartado 3.6](#) ya que es uno de los temas básicos a considerar en la planificación de proyectos.

### 3.5. El Problema de la Programación o Secuenciación de Actividades

"Si no puedes resolver un problema, entonces hay una manera más sencilla de resolverlo: encuéntrala".

**George Pólya (1887-1985)** Matemático húngaro

Los estudios sobre la producción y el trabajo, relacionados con la programación de proyectos y la puesta en práctica de estas experiencias, han dado lugar a que los especialistas se planteen los problemas reales que se generan y las formas o metodologías a utilizar para darles una solución, a ser posible la óptima, la mejor. Se crea cierta confusión con las diferentes denominaciones que reciben estos problemas en la literatura, por ello en el [apartado 4](#) se utilizará un sistema de clasificación basado en unos parámetros generales que ayudará a solucionar las dudas que al respecto le puedan surgir al lector del trabajo.

No solamente hay una gran variedad de problemas derivados del básico, sino que además el mismo problema básico recibe distintos nombres. Es conveniente indicar los diferentes modos de nombrar al problema que nos ocupa en estos momentos, pues se le conoce como **Problema de Planificación de Actividades**, **Problema de Programación de Actividades**, **Problema de Secuenciación de Actividades**, o en inglés, **Project Scheduling Problem (PSP)**.

Si bien se utilizan indistintamente en los escritos unos apelativos u otros, sabemos que planificación, programación y secuenciación no son términos equivalentes entre sí. Planificar



respondería a las preguntas de qué hacer, cómo hacerlo y con qué; programar sería indicar el instante en que deben iniciarse y terminarse las distintas actividades; y secuenciar correspondería a marcar el orden en que se deben ejecutar. A partir de este momento, en este trabajo con el fin de una menor complicación se seguirá el criterio extendido de no distinguir entre el significado semántico de estos términos.

Dentro del proceso de planificación y programación del proyecto, las metodologías tradicionalmente más usadas para determinar las tareas y secuenciarlas, son los grafos, mencionados anteriormente en el [apartado 2](#). Con ellos se dispone de unas herramientas de cálculo y de representación gráfica muy usados. Los grafos a su vez se subdividen en:

- Grafos de Actividades en Flecha:
  - Método del Camino Crítico.
  
- Grafos de Actividades en Nodo:
  - Método de los Potenciales o de ROY.
  - Red de Precedencias.

Con estos podemos realizar la planificación y programación de un proyecto, resultando muy prácticos cuando se trabaja con pocas actividades; pero a medida que el número de estas actividades va aumentando, la empresa se va haciendo cada vez más y más complicada ([Ponz Tienda, 2008](#)); algunos de estos problemas pueden ser muy difíciles, se llega a un punto en el que los grafos dejan de ser eficaces y hay que hacer uso de algoritmos y formulaciones matemáticas para resolverlos.

Para dar respuesta a estos problemas, aparece y se desarrolla una nueva rama de las Matemáticas Aplicadas conocida con el nombre de **Investigación Operativa**, basada en postulados como la Teoría de la Decisión, la Programación Matemática o Teoría de la Optimización, la Teoría de grafos, la Teoría de control,..., y cuyo objetivo básico es el servir de apoyo en la toma óptima de decisiones cuando se afrontan problemas de decisión complejos.

Así pues, la Investigación Operativa es la ciencia que estudia las situaciones complejas que se originan en la vida real, desarrollando técnicas para modelar y resolver esas situaciones, transmitiendo a la colectividad científica los logros obtenidos. Según esta ciencia, no solamente es necesaria la utilización de herramientas para el conocimiento, sino que es fundamental verificar la aplicabilidad de los modelos matemáticos y de los algoritmos<sup>12</sup> desarrollados, haciendo hincapié en el **aspecto práctico** de los mismos; en este pensamiento de ser útil reside la importancia de la Investigación Operativa.

Por tanto, la Investigación Operativa aporta las herramientas necesarias para el examen y resolución de problemas en múltiples contextos como la ingeniería, la física, la biología, la medicina..., y otras muchas ciencias básicas empleadas para solucionar la **incertidumbre** que generan y la **optimización** que requieren, el progreso y los continuos avances tecnológicos.

En un estado donde no hay incertidumbre, la optimización persigue la solución óptima (la mejor de las factibles), de acuerdo a la función o funciones objetivo que inicialmente se han marcado como meta. En presencia de incertidumbre se buscará además optimizar los

---

<sup>12</sup> Algoritmo: Conjunto finito y ordenado de órdenes cuya aplicación permite resolver un problema.



objetivos, pronosticando cada alternativa de decisión mediante probabilidades. Interesa saber si dado un conjunto de variables de entrada se van a dar siempre los mismos datos de salida o estos cambiarán en función del azar, se estaría hablando respectivamente de ambientes deterministas o de ambientes estocásticos.

Pero sigamos profundizando. Una de las ramas más importantes y más emprendedoras dentro de la Investigación Operativa es la **Programación Matemática o Teoría de la Optimización**. En sus orígenes fue una forma de resolver problemas de programación mediante métodos matemáticos; hoy fundamentalmente es utilizada en las empresas para la toma de decisiones y está basada en la optimización de la llamada función objetivo, la cual está supeditada a ciertas condiciones o restricciones sobre unas variables.

Dependiendo de la forma de esas funciones, las variables y las restricciones, surgen diferentes tipos de **Problemas de Programación Matemática**, como por ejemplo, el **Problema de Secuenciación de Actividades**, el Problema del Transporte, el Problema del Empaquetado, el Problema de la Asignación Óptima, el Problema de la Mochila, el Problema del Transbordo, el Problema de la Dieta, el Problema de Recubrimiento,..., y otros muchos que pueden aparecer como extensiones de estos, despertando todos ellos desde hace décadas el interés de los investigadores.

A su vez, del **Problema de Programación o Secuenciación de Actividades**, o "**Project Scheduling Problem (PSP)**", al que se refiere este apartado, recoge una extensa clase de problemas de Optimización Combinatoria<sup>13</sup>, de los que se encuentran variantes o extensiones tales como el Problema estándar RCPS (Resource Constrained Project Scheduling Problem), el problema RCPS con diferentes modos de ejecución o MRCPS, el problema RCPS con ventanas de tiempo, el problema RCPS multiproyecto o RCMPSP, el problema multiproyecto con múltiples modos de realización o MRCMPSP, el problema de ventanas de tiempo con múltiples modos de ejecución o MRCPS/max, el Problema RLP (Resource Leveling Problem), el Problema RLP/max, etc.

Las distintas extensiones del problema presentan entre ellas algunas diferencias, de manera que por ejemplo pueden tener más de una función objetivo, las actividades se pueden realizar de varios modos (con holguras o sin holguras), etc. En ocasiones estas diferencias aunque parezcan ser muy pequeñas, cambian sustancialmente la dificultad del problema y el método de resolución; así se distinguen unos de otros, pudiéndose clasificar a su vez en distintas familias en función del área a la que pertenezcan dentro del concepto general de la secuenciación de actividades.

Dada la dificultad que reúnen la mayoría de estos problemas, es necesaria la utilización de modelos para representarlos, entendiendo como modelo *la abstracción de una parte o segmento de la realidad*, simplificando dicha realidad, además, estos modelos *pueden ser de diferentes tipos (a escala, lógicos, matemáticos)* (Yepes Piqueras, 2012) y (Yepes Piqueras, 2002). Para (García Sabater & Maheut, 2011) son una representación externa y explícita de la realidad; sirven para entenderla, se pueden implementar en un ordenador para gestionar la información que genera una empresa y mediante su resolución ayudan en la toma de decisiones.

---

<sup>13</sup> Optimización Combinatoria: rama de la Optimización que busca el óptimo entre conjuntos discretos de soluciones, es decir, finitos o infinitos numerables).



En concreto el modelo matemático, para mostrar las relaciones entre los distintos elementos que lo forman, utiliza un lenguaje simbólico matemático como lenguaje formal de expresión, el cual permite reflejar ideas complejas. Los modelos matemáticos se pueden clasificar según distintos criterios, en función de sus cualidades distintivas en empíricos o basados en principios físicos, estocásticos o deterministas, agregados o distribuidos, con o sin restricciones, lineales o no lineales, estáticos o dinámicos, ..., y así podríamos continuar puesto que como se puede vislumbrar hay múltiples formas de clasificarlos, tantas como características propias presentan.

Se parte de la base de que los modelos matemáticos son resolubles, pero esto no siempre es así; para ello se tiene que definir bien el problema, deben estar bien planteados, generados y validados, y además se han de poder utilizar en la realidad ([García Sabater & Maheut, 2011](#)); recordemos que el interés básico de la Investigación Operativa es su aplicación a casos prácticos.

A modo de resumen de lo comentado en los últimos párrafos, a continuación se ilustra de manera esquemática mediante la [Figura 4](#) como para resolver un problema, partiendo de un núcleo formado por los algoritmos como procedimientos para la resolución de los modelos matemáticos y estableciendo una jerarquía entre distintos subconjuntos, se alcanza la fusión de estos con el conjunto de conocimientos representado por la Ciencia.

Sin entrar en la enumeración exhaustiva de todas las bifurcaciones que encontraríamos dentro de cada uno de los niveles reflejados, el gráfico pretende aclarar la interdependencia entre las distintas ramas científicas; a su vez estas se van derivando unas a partir de otras hasta llegar a la resolución de problemas cada vez más complicados.

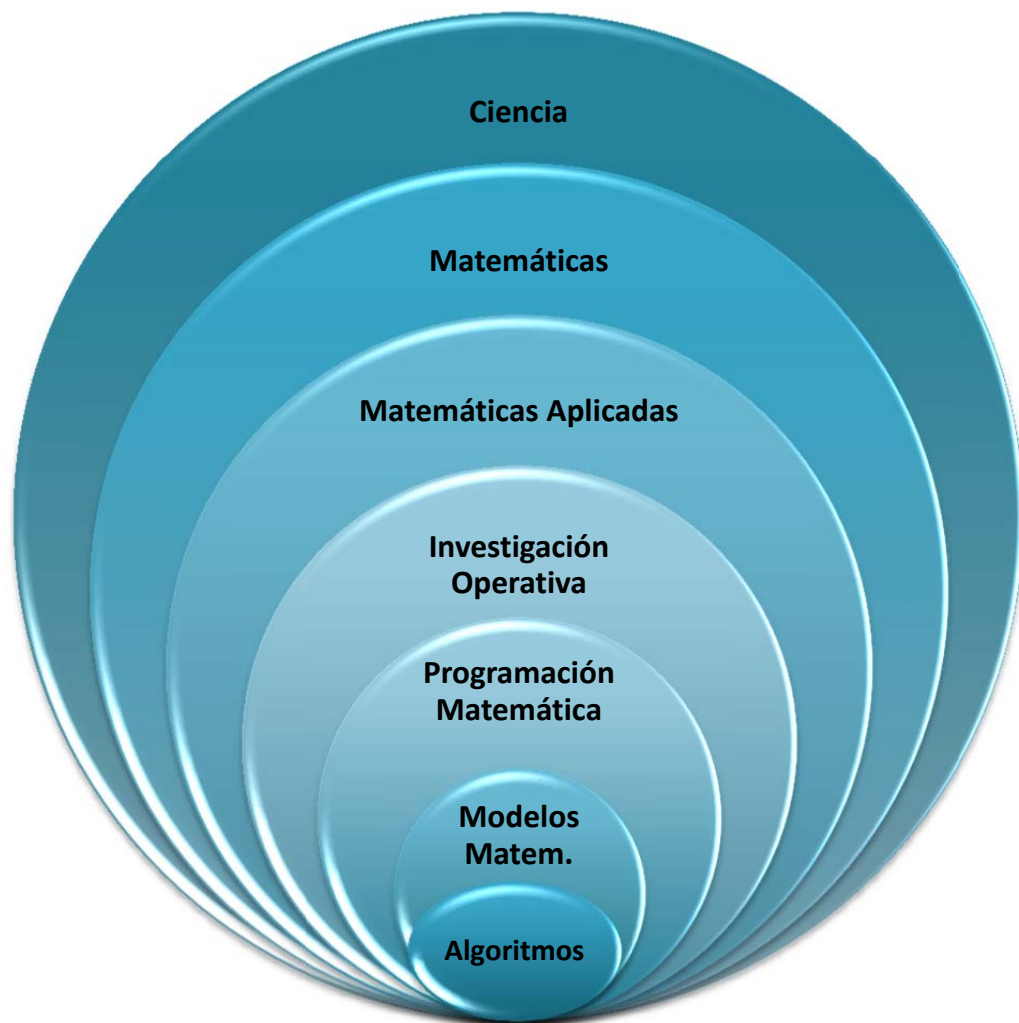


Figura 4: Jerarquía científico-técnica para resolución de problemas.

Ante problemas de secuenciación (o problemas de optimización combinatoria cuando las soluciones factibles son discretas), para poder desarrollar un algoritmo eficiente que los resuelva, es importante la cuestión de la complejidad de los problemas (Brucker & Knust, 2012). A poco que un modelo sea algo complejo, para resolver un problema va a necesitar que se realicen muchas operaciones matemáticas, siendo necesario el uso de máquinas para solucionar metódicamente el algoritmo.

La eficiencia o rendimiento de un algoritmo<sup>14</sup> para resolver un problema, se puede medir fundamentalmente por:

- Su tiempo de ejecución. El cual dependerá del número y tipo de pasos que necesite para procesar un conjunto de datos.
- La cantidad de memoria que necesite para ser ejecutado por un ordenador.

En función de lo anterior, los problemas según su complejidad se dividen en:

<sup>14</sup> Cuantos menos recursos (tiempo, espacio de memoria y procesadores) consume un algoritmo, más eficiente es. De estos recursos el más importante a minimizar es el tiempo.



- **Problemas P:** Pueden ser resueltos por un algoritmo en tiempo polinómico determinista.
- **Problemas NP:** El término NP proviene de tiempo polinómico no determinista y se deriva del uso de polinomios de tiempo en máquinas de Turing no deterministas (Sipser, 2006), o lo que es igual, un problema-NP no puede ser resuelto por un algoritmo en tiempo polinomial determinista si no por un algoritmo en tiempo polinomial no determinista.

Las definiciones anteriores ciertamente parecen un trabalenguas, intentemos explicarlo de forma más clara. Si en un algoritmo determinista sólo hay una única manera de seguir avanzando hacia adelante para resolver el problema, en uno no determinista habría distintas formas. Según (Sipser, 2006), en un algoritmo no determinista, entre todas esas formas se iría escogiendo aleatoriamente alguna de ellas para ir continuando.

En un algoritmo determinista el tiempo se mide desde que empieza la primera orden hasta que se termina el proceso. En uno no determinista se van escogiendo caminos que pueden ser los más largos y esta serie de malas decisiones nos llevarán a un mayor tiempo de ejecución, esto es precisamente lo que se entiende como tiempo no determinista.

Continuando con el tema de la complejidad, como indica José A. Mañas, profesor del Departamento de Ingeniería de Sistemas Telemáticos de la Universidad Politécnica de Madrid, en la práctica, para resolver un problema se necesita un algoritmo o método de resolución y que este se pueda ejecutar en un ordenador real mediante un programa o codificación (Mañas, 1997). Pero esto no siempre es posible.

La complejidad de un algoritmo es un conjunto de funciones cuya notación se presenta como  $O(g(n))$ . Así mismo, se dice que  $O(g(n))$  define una **orden de complejidad**, y  $g(n)$  es la función que determina el uso de una familia de recursos de la máquina (tiempo, espacio de memoria o procesadores).

Las órdenes de complejidad de un algoritmo presentan una jerarquía, de manera que cuando una es superior en complejidad, contiene como subconjuntos a todas las de rango inferior. Esa jerarquía es la siguiente (Mañas, 1997):

$O(1)$	Orden constante
$O(\log n)$	Orden logarítmico
$O(n)$	Orden lineal
$O(n \log n)$	Orden cuasi lineal
$O(n^2)$	Orden cuadrático
$O(n^a)$	Orden polinomial ( $a > 2$ )
$O(a^n)$	Orden exponencial ( $a > 2$ )
$O(n!)$	Orden factorial





Pues bien, los problemas **NP-difíciles** o **NP-hard** (como es el caso del RCPSP y del RLP), son aquellos para los que no se conoce (y se cree que no se conocerá nunca), un algoritmo que en un tiempo polinómico (en función de los datos), encuentre la solución óptima, es decir, no se pueden resolver en un tiempo polinomial determinado. El rendimiento medio o tiempo que necesita el algoritmo para procesar las variables de entrada en base a la función o funciones objetivo y ofrecer los datos de salida, es exageradamente enorme, no es aceptable; por tanto, el algoritmo no es eficiente, es intratable.

Estos problemas presentan pues, una **complejidad en tiempo exponencial** o **de orden exponencial**, lo que significa que el tiempo que utilizan para resolver un problema concreto crece según una función exponencial de acuerdo a su tamaño. También se puede decir que el universo de soluciones o de combinaciones posibles entre las que se encuentra la óptima, crece exponencialmente produciéndose una explosión combinatoria y, aunque la cantidad de soluciones sea un número finito, esta cantidad es tan enorme, tan abrumadora, que ni tan siquiera con un ordenador muy potente es viable encontrar el óptimo en un tiempo admisible.

Por ejemplo, suponiendo que se tuviese un proyecto formado por 30 actividades con 2 posiciones de holgura para cada una de ellas, el universo de soluciones para procesar el problema serían:

$$O(2^{30}) = 1.073.741.824 \text{ soluciones o } 1.073 \text{ millones de soluciones}$$

Si se dispusiese de una máquina capaz de procesar 1 millón de soluciones al problema por segundo, ésta tardaría 1.073'74 segundos o 17'90 minutos en realizar el proceso anterior completo. Pero, y si fuese aumentando el número de posiciones de holgura:

$O(3^{30}) = 205.891 \times 10^9$ soluciones	$\Rightarrow$	6,53 años
$O(4^{30}) = 1.152 \times 10^{15}$ soluciones	$\Rightarrow$	36.529 años
$O(5^{30}) = 931.322 \times 10^{15}$ soluciones	$\Rightarrow$	29.532. 026 años

Como se aprecia, los costes computacionales<sup>15</sup> o tiempos obtenidos son inviables y sin embargo, un proyecto de 30 actividades con 3, 4 o 5 posiciones de holgura es muy usual, lo raro es que se disponga de un proyecto con actividades sin holguras<sup>16</sup> y por tanto con un único modo de ejecución. Por esto, para resolver los problemas NP-hard son imprescindibles las heurísticas (de las cuales se hablará en el [apartado 4.4](#)), pues si bien no dan la solución óptima, si dan soluciones que por aproximarse mucho a la óptima pueden ser consideradas como válidas.

<sup>15</sup> Coste computacional: coste temporal de los algoritmos.

<sup>16</sup> La holgura de una actividad es el tiempo que podemos retrasar el comienzo de esta sin que afecte a la duración del proyecto.



## 3.6. El Problema de la Limitación en el Consumo de los Recursos; el RCPSP estándar

El **Problema de Programación o Secuenciación de Proyectos con Recursos Limitados** o “Resource Constrained Project Scheduling Problem (**RCPSP**)”, desde el punto de vista algorítmico es mucho más difícil de solucionar que el de la Nivelación de Recursos ([Ponz Tienda, 2011](#)). Esto se debe a que el universo de soluciones del RCPSP es regular y aunque este sea más grande que el universo del RLP, a medida que se va ramificando el problema y nos encontramos con un resultado A, mejor que otro B, se corta la rama de B ya que por ese camino no vamos a hallar una mejor solución que A, en consecuencia el problema se va acotando y reduciendo. Por el contrario al encontrarse las soluciones del RLP en un universo irregular, hemos de continuar por ambos caminos hasta el final, pues es posible encontrar más adelante otro resultado preferible, es decir, en este caso de irregularidad no podemos ir desechando ninguna rama, hemos de avanzar con todas hasta el final y el problema se complica mucho más.

El problema RCPSP consiste en secuenciar las actividades de forma tal que no se sobrepase *la disponibilidad de los recursos*<sup>17</sup> que es necesaria para ejecutar cada una de las actividades por requerimiento o demanda de éstas. Esto puede conducir a que aumente el *makespan* del proyecto si no se cuenta con el número suficiente de recursos. Los recursos suponen la mayor parte del coste del proyecto, por ello han de distribuirse de forma eficaz y no se deben incrementar a cambio de disminuir los beneficios previstos en el proyecto; para ([Blazewicz, et al., 1983](#)) cada trabajo se puede ejecutar como máximo por un equipo al mismo tiempo, pero como indican Demeulemeester y Herroelen ([Demeulemeester & Herroelen, 2002](#)), pueden ser compartidos por varias actividades y sólo el uso de los recursos totales a utilizar es limitado, es decir, las tareas se han de realizar sin exceder la cantidad máxima de recursos disponible para la totalidad del proyecto.

Por ejemplo, si tras la planificación de un proyecto, en el histograma de la carga de recursos obtenemos picos en determinados días de 12 operarios y sólo se puede disponer de un máximo de 8 por día, hemos sobrepasado las cantidades de recursos que se permiten en determinados periodos, es decir, nos encontramos ante un problema RCPSP. Hay que matizar que para la detección de posibles conflictos entre los recursos, en el problema **básico** RCPSP sólo se consideran los que son de **tipo renovable** y se encuentran disponibles en una cantidad determinada según se ha establecido en la planificación del proyecto ([Klein, 2000](#)). Realmente el contemplar solamente los de tipo renovable no se justifica en edificación, de hecho hay investigadores que trabajan también con los no renovables y con los parcialmente renovables, pero estos dan lugar a variaciones sobre el problema básico que no vamos a tratar en esta ocasión.

¿Cómo solucionar el problema? Para resolverlo de forma adecuada se deben combinar los recursos con la secuenciación de las distintas tareas, determinando los tiempos de ejecución, sin necesitar en ningún momento más recursos de los disponibles, de manera que el incremento en la duración final del proyecto sea mínimo.

---

<sup>17</sup> La disponibilidad de un recurso es la cantidad de recurso de la que se dispone en un instante o periodo de tiempo determinado.



Como **características** del **RCPS** estándar o **SMPSP**<sup>18</sup> del que se derivan otras extensiones o variedades del mismo, hay que enumerar:

- Es una **generalización** del Problema JOB-SHOP<sup>19</sup> y al igual que éste continúa estando abierto y es objeto de estudio por parte de los expertos.
- El **modelo matemático** que se genera es estático, se desarrolla en un ambiente determinista, por tanto las variables son conocidas, no son aleatorias.
- Se puede **resolver con algoritmos** exactos (Programación Lineal Entera), aunque en ocasiones el tiempo empleado para encontrar el óptimo puede ser muy elevado si se dispone de muchas variables, no resultando eficaz en la práctica, debiendo hacer uso de algoritmos aproximados (Programación con Heurísticas o Metaheurísticas), que si bien no dan el óptimo, si ofrecen soluciones cuasi-óptimas.
- En relación a las **actividades**, estas serán consideradas como continuas, no se podrá interrumpir su ejecución<sup>20</sup> para que se inicie otra que debe realizarse de manera más urgente, lo que en inglés se denomina “*no preemption*”. Tendrán una duración conocida y un consumo dado de recursos. Se ejecutarán de un único modo, de lo contrario podrían cambiar las necesidades en relación al tipo y cantidad de recursos que son necesarios. En las relaciones de precedencia solamente se darán las de tipo Final-Inicio
- En el problema sólo se tienen en cuenta los **recursos** que sean renovables<sup>21</sup> y estos están restringidos, de manera que la cantidad disponible se renueva periodo a periodo, estando limitada solamente la cantidad utilizada en cada instante y que queda definida en la planificación de las actividades. Obviamente, serán respetadas las limitaciones de los recursos. No hay un consenso entre los expertos sobre si sólo se deben considerar los recursos renovables humanos o de mano de obra, despreciando los otros tipos existentes, tal vez pensando en que la carga económica sobre el proyecto de este tipo de recursos es la más significativa, excluyendo por tanto a los recursos renovables materiales.
- El problema estándar **RCPS** o **SMPSP** es **mono-objetivo**, o lo que es lo mismo, dispone de una única función objetivo.  
 Muchos de los problemas reales, pero que ya no formarían parte del grupo de los SMPSP, requieren optimizar a la vez más de un objetivo, es decir, son problemas

<sup>18</sup> En ocasiones también se le denomina como “Single-Mode Project Scheduling Problem (**SMPSP**)”.

<sup>19</sup> Problema JOB-SHOP SCHEDULING: problema de optimización combinatoria NP-hard, consiste en encontrar una planificación factible que minimice el *makespan* del proyecto y en la cual cada trabajo tiene una ordenación determinada, cada trabajo tiene un orden propio con el cual visitar las máquinas; ninguna máquina puede realizar más de una operación y ninguna operación puede ser realizada al mismo tiempo en más de una máquina.

<sup>20</sup> La continuidad de las tareas, es una relajación del problema general para reducir el universo de soluciones factibles al problema.

<sup>21</sup> No se consideran los no renovables o los parcialmente renovables por tratarse del RCPS básico o estándar. En otras variedades del problema sí se tienen en cuenta.



multi-objetivos o MRCPSP<sup>22</sup>. Una muestra de estos últimos sería un proyecto en donde se tienen dos funciones objetivo, por un lado minimizar el coste y por otro minimizar el plazo de ejecución.

Como ya se ha comentado, estas características del problema estándar son muy rígidas y normalmente no reflejan la realidad. Si fijamos la atención simplemente en los recursos, como prueba de ello se observa que acaban agotándose con el tiempo, no duran eternamente; su disponibilidad puede ser predecible o no, a veces depende de circunstancias que no se pueden controlar; un recurso puede ser utilizado para varias tareas y una tarea puede necesitar varios recursos. El problema se extiende y se encuentra con un sinnúmero de posibilidades. Esta imposición de continuidad es una “relajación” o simplificación del problema general para reducir el universo de soluciones del mismo.

Este trabajo no tiene como finalidad entrar en el estudio pormenorizado de las distintas variantes y extensiones del problema estándar RCPSP, este análisis es tan extenso que podría formar parte de una investigación posterior. De las diferentes extensiones encontramos numerosas referencias en los trabajos de investigación como por ejemplo y sin tener la intención de hacer una lista completa: (Arecas, et al., 2011) presentan una clasificación de las distintas variantes; (Al-Anzi, et al., 2010) proponen un nuevo problema al que llaman Weighted Multi-Skill Resources Project Scheduling Problem (WMSPSP) en donde un recurso puede hacer más de una actividad y además se tiene en cuenta la habilidad del recurso para realizar el trabajo; (Hartmann & Briskorn, 2010) desarrollan varias extensiones del RCPSP como extensiones multi-modo u objetivos basados en el VAN<sup>23</sup> o Multi-modo-RCPSP; (Herroelen, et al., 1996) analizan problemas como el Preemptive-RCPSP, el Generalized-RCPSP, el RCPSP-GPR con relaciones de precedencia generalizadas que permite relaciones de tipo inicio-inicio, final-inicio, inicio-final y final-final con desfases máximos y mínimos, y el problema multi-modo RCPSP entre otros; (Lova, et al., 2006) también estudian el RCPSP multi-modo; (Álvarez-Valdés Olaguíbel, et al., 2009) analizan el RCPSP considerando los recursos parcialmente renovables por representar situaciones más próximas a la realidad; (Demeulemeester & Herroelen, 2002) y (Klein, 2000) igualmente trabajan con las distintas variantes del problema básico; (Ponz Tienda, 2011) establece un nuevo enfoque del Generalized Resource-Constrained Project Scheduling Problem bajo la perspectiva de la lógica borrosa; (Cervantes Posada, 2010) hace una nueva propuesta para el RCPSP multi-modo; (Navascués Fernández, s.f.) trata el RCPSP desde el punto de vista del multiproyecto.

Para (Hartmann & Briskorn, 2010) la gran cantidad de estudios existentes muestra que el uso del RCPSP con todas sus extensiones no se limita a las aplicaciones en su campo original, la programación de proyectos. El RCPSP y sus generalizaciones también se perciben como una colección de potentes herramientas que permiten describir un sinnúmero de problemas de optimización altamente complejos de diversos sectores o campos. De ahí la importancia de estas investigaciones.

<sup>22</sup>MRCPSP: Multi-Mode Resource Constrained Project Scheduling Problem.

<sup>23</sup> VAN: Abreviatura del concepto que se conoce en economía como Valor Actual Neto.



**Formulación modelo matemático del RCPSP estándar (Álvarez-Valdés & Tamarit, 1989)**

Existen diversas formulaciones de este problema presentadas por Pritsker, Kaplan, Klein, Álvarez-Valdés, y Mingozzi. Las diferencias entre las formulaciones de Pritsker, Kaplan y Álvarez-Valdés, se pueden consultar en (Ponz Tienda, 2010).

El RCPSP se define como sigue; el proyecto viene dado por una red de actividades en nodo  $G = (J, A, d, u)$ , con un conjunto de nodos  $V = \{1, \dots, n\}$  que representan las actividades y un conjunto de arcos  $A = \{(i, j) | j \in V \text{ y } i \in P_j\}$  que simbolizan las relaciones de inicio-fin de las actividades, siendo  $i$  la antecesora de  $j$  ( $j = 1, \dots, n$ ),  $j$  la sucesora de  $i$ . El proyecto iniciará y finalizará con dos actividades ficticias ( $j = 1$  y  $j = J$ ) de duración cero.  $Pred_j$  es el conjunto de antecesoras de  $j = 2, \dots, J - 1$ ; donde cada antecesora de  $j$  debe terminar dentro del horizonte de tiempo  $T$ , antes de que  $j$  pueda empezar.

**Hay una restricción que obligatoriamente se ha de cumplir en cuanto a la disponibilidad de recursos por periodo de tiempo, de forma que se han de programar las tareas en un periodo de tiempo de manera que la suma de las cantidades de los recursos que necesitan las tareas nunca exceda la disponibilidad existente. Se busca que el makespan del proyecto se reduzca lo más posible teniendo en cuenta las precedencias de las actividades y las limitaciones de los recursos.**

No se permite la interrupción de actividades, o lo que es igual, si una actividad  $j$  se ha iniciado en el instante  $t$ , esta debe realizarse inmediatamente en los periodos  $t + 1, \dots, t + d_j$  siendo  $d_j$  ( $d_j \geq 0$ ) la duración total de la actividad  $j$  ( $j = 1, \dots, n$ ). En cada periodo de proceso de  $j$  se requerirán  $r_{j,k} \geq 0$  unidades del recurso  $k$ . Una vez iniciado el procesamiento de  $j$ , este no puede ser interrumpido. Por último hay  $k = 1, \dots, m$  recursos diferentes, cada uno con una capacidad finita de  $R_k \geq 0$  unidades por periodo  $t$ . Se define la variable  $f_j \geq 0$  como el tiempo de finalización del procesamiento de la actividad  $j$  y  $A(t)$  es el conjunto de actividades que se procesan en el periodo  $t$ . Se puede elaborar una programación, es decir, un vector  $f$  de tiempos de finalización que sea realizable teniendo en cuenta las limitaciones de las relaciones de precedencia y las restricciones de los recursos mediante el empleo de las siguientes fórmulas (Kolisch, 1999):

$$\text{Min } f_n \tag{3.1}$$

$$f_h \leq f_j - d_j (= s_j), \quad j = 2, \dots, J - 1 \quad h \in \text{Pred}_j \tag{3.2}$$

$$\sum_{j \in A(t)} r_{j,k} \leq R_k \quad k \in K; t \geq 0 \text{ y } t = 1, \dots, f_n \tag{3.3}$$

$$f_j \geq 0 \text{ y enteras}, \quad j = 2, \dots, J - 1 \tag{3.4}$$



La función objetivo [3.1] minimiza el makespan del proyecto, siendo  $f_n$  el tiempo de terminación de la actividad ficticia final. La ecuación [3.2] representa las restricciones de las relaciones de precedencia, mediante las que se retrasa el inicio de una actividad hasta que todas sus predecesoras están terminadas;  $s_j$  es el tiempo de inicio de la actividad  $j$  y  $s_j = f_j - d_j$ . Las restricciones [3.3] limitan la cantidad de cada recurso  $k$  en cada instante  $t$  a la disponibilidad existente  $R_k$  en dicho instante, es decir, en cada periodo  $t$  la capacidad disponible de recursos  $R_k$  debe estar asignada a las actividades que están en curso, sólo una actividad que está procesándose puede hacer uso de los recursos disponibles. Por último [3.4] obliga a que las variables de decisión sean enteras y positivas.



#### 4. Clasificación y resolución de Problemas de Programación de Proyectos con Recursos Limitados

##### 4.1. Clasificación de Problemas de Programación de Proyectos con Recursos Limitados

Ante la gran variedad de tipos de problemas de programación de proyectos, tras unos debates del “Workshop on Scheduling and Heuristic Search” (Taller sobre Planificación y Búsqueda Heurística) celebrados en la Universidad de California en 1997, se fue desarrollando el esquema de clasificación que encontramos en (Herroelen, et al., 1998) y (Demeulemeester & Herroelen, 2002) que se explicará seguidamente. Este sistema es menos confuso y vago que denominar a los problemas con una serie siglas encadenadas como por ejemplo RCPSP, RCPSP-GPR, MRCPS, etc., facilitando de esta forma la presentación y la discusión de los problemas de planificación.

Por otro lado, existe otra notación de (Brucker, et al., 1999) que aparece con más frecuencia en la literatura, muy extendida entre la comunidad científica y que es la más citada. (Herroelen, et al., 2001) la analizan encontrándole determinadas limitaciones para su utilización. (Brucker, et al., 1999) distinguen entre programación de proyectos, programación de proyectos multi-modo y programación de máquinas. El modelo de clasificación de (Demeulemeester & Herroelen, 2002) es similar al definido para la programación de máquinas explicado por (Graham, et al., 1979), (Blazewicz, et al., 1983), (Lawler, et al., 1989) y (Brucker, 1995), los cuales se basan en la teoría de la complejidad computacional<sup>24</sup>, haciendo una analogía entre la programación de proyectos y de máquinas, llegando a la conclusión de que se pueden tratar de la misma forma al considerar las máquinas como recursos renovables.

Según (Demeulemeester & Herroelen, 2002), el sistema tiene las siguientes ventajas:

- Es flexible y viable pues permite tratar los problemas deterministas y estocásticos que se dan en la programación de proyectos.
- Se evita el uso de largas concatenaciones de siglas y en consecuencia se consigue reconocer con más facilidad las características de los problemas.
- Se identifica con rapidez el área de investigación al que pertenece el problema.
- Se simplifica la evaluación de la complejidad del problema, tema muy importante para conseguir una buena solución.
- Facilita la coincidencia y el encuentro de los investigadores.

#### Descripción de los campos

El sistema consta de 3 campos  $\alpha$  |  $\beta$  |  $\gamma$ .

- **Campo  $\alpha$**

Especifica las características de los recursos. A su vez está formado por cuatro elementos,  $\alpha_1, \alpha_2, \alpha_3$  y  $\alpha_4$

<sup>24</sup> El ir añadiendo restricciones a los recursos afecta a la complejidad computacional.



- $\alpha_1$  Define los recursos estructurales (máquinas). Si por la estructura de los recursos las actividades se hacen en una fase única o en multi-fases (modo único o multi-modos).  $\alpha_1 \in \{\circ, P1, P, Pm, Q, Qm, R, Rm, O, Om, J, Jm\}$ .
- $\alpha_2$  Indica el número de recursos.  $\alpha_2 = \{\circ, 1, m\}$ .
- $\alpha_3$  Especifica los tipos de recursos utilizados.  $\alpha_3 = \{\circ, 1, T, 1T, v, \chi, \sigma\}$ .
- $\alpha_4$  Concreta la disponibilidad de los recursos.  $\alpha_4 \in \{\circ, \kappa, va, a, \tilde{a}, va, v\tilde{a}\}$ .
- **Campo  $\beta$**

Detalla las características de las actividades. Consta de nueve parámetros.

- $\beta_1$  Señala la posibilidad de interrumpir las actividades para continuarlas más tarde.  $\beta_1 = \{\circ, pmtn, pmtn - rep\}$
- $\beta_2$  Describe las restricciones por precedencias.  $\beta_2 \in \{\circ, cpm, min, gpr, prob, fuzzy\}$ .
- $\beta_3$  Detalla cómo son los tiempos de preparación (cero, determinísticos, estocásticos o borrosos).  $\beta_3 \in \{\circ, \rho_j, \rho_j, \tilde{\rho}_j\}$ .
- $\beta_4$  Especifica cómo es la duración de las actividades.  $\beta_4 \in \{\circ, cont, d_j = d, d_j, \tilde{d}_j\}$ .
- $\beta_5$  Describe cómo son los plazos, hitos, o fechas límite del proyecto.  $\beta_5 \in \{\circ, \delta, \delta_j, \delta_n\}$ .
- $\beta_6$  Concreta la naturaleza de los requerimientos de las actividades.  $\beta_6 \in \{\circ, k, vr, disc, cont, int\}$ .
- $\beta_7$  Define el tipo y número de modos de ejecución posibles.  $\beta_7 \in \{\circ, mu, id\}$ .
- $\beta_8$  Precisa los tipos de flujos de caja de las actividades del proyecto.  $\beta_8 \in \{\circ, c_j, c_j^+, c_j, c_j^+, \tilde{c}_j, per, per, p\tilde{e}r, sched, sched, sch\tilde{e}d\}$ .
- $\beta_9$  Indica los tiempos de cambio para preparar las máquinas, procesos, etc.  $\beta_9 \in \{\circ, s_{jk}, s_{jk}, \tilde{s}_{jk}\}$ .
- **Campo  $\gamma$**

Indica los criterios de optimización como la minimización del makespan, la minimización del coste del proyecto, la maximización del Valor Actual Neto, etc. Hay infinidad de funciones objetivo, por lo que no se puede hacer una enumeración detallada de los distintos parámetros que se podrían considerar.

Para una mayor información sobre los factores de clasificación se recomienda al lector la consulta de (Demeulemeester & Herroelen, 2002), dados los objetivos de este trabajo sería demasiado extenso realizar una explicación detallada de los mismos. Con esta notación sistemática se definen más de 70 parámetros diferentes que si bien sirven como criterio diferenciador entre los distintos tipos de problemas y los determinan con mayor exactitud, han de conocerse para hacer un buen uso de ellos; este es precisamente su punto débil y





posiblemente el motivo de que hasta la fecha no esté más extendido su empleo. Por otra parte ofrece dos grandes ventajas a añadir a las indicadas por (Demeulemeester & Herroelen, 2002), conocer solamente con la notación todas las características y datos de los problemas que se están estudiando y permitir expresarlos dentro de un mundo real. Utilizando el esquema de clasificación anterior, un problema RCPSP en donde se pretende como objetivo minimizar el makespan, se designaría como:

$$P1,2,1,5 | cpm,^\circ | C_{\max}$$

En donde  $P1$  hace referencia a que el proyecto es procesado por una sola máquina<sup>25</sup>; el 2 implica que se consideran 2 tipos de recursos diferentes, el 1 significa que estos recursos son renovables y el 5 que hay una disponibilidad constante por periodo de 5 recursos; con  $cpm$  se define el tipo de precedencias como estrictamente de final-inicio con un desfase de 0, sin holguras, con  $^\circ$  que las actividades tienen duraciones arbitrarias enteras;  $C_{\max}$  establece que la función objetivo es minimizar el makespan del proyecto. Sin embargo en el mismo libro de (Demeulemeester & Herroelen, 2002), se encuentran notaciones que no se corresponden exactamente con los criterios establecidos, pues su intención es ofrecer al usuario un cierto grado de flexibilidad; con ello no intentan dar cabida a cualquier problema de programación de proyectos si no a que se muestre información suficiente en la etiqueta sin mayores complicaciones, encontrándose que el RCPSP es clasificado como:

$$m,1 | cpm | C_{\max}$$

En este caso se está tratando la generalización del problema RCPSP;  $m,1$  hace referencia a que se dispone de  $m$  tipos de recursos renovables para los que se especifica la disponibilidad por periodo de tiempo unitario,  $cpm$  y  $C_{\max}$  representan lo mismo que se ha indicado en el párrafo anterior. El mismo problema se nombra en (Brucker, et al., 1999) según se indica a continuación, en donde  $PS$  hace referencia a "Project Scheduling",  $prec$  a "restricciones de precedencia entre las actividades", y  $C_{\max}$  al igual que en los ejemplos anteriores representa a la función objetivo de minimizar el makespan del proyecto:

$$PS | prec | C_{\max}$$

Como puede observarse, todavía no se dispone de un esquema de clasificación de los problemas de programación de proyectos que ponga de acuerdo a todos los investigadores. Esto origina confusión y falta de praxis, haciendo que de momento no se pueda utilizar una notación sin reservas aplicable a los proyectos de construcción. Profundizar y recoger las distintas clasificaciones encontradas en la literatura puede ser motivo de una investigación posterior.

Para aplicar correctamente el esquema de una cadena de fabricación al mundo de la construcción debe ser fácil trasladar el concepto de máquina a este campo. En una cadena el producto se obtiene tras el paso en serie o en paralelo de los elementos básicos que se van transformando gracias a las distintas máquinas, pero un edificio es un producto estático muy

<sup>25</sup> Una máquina es un procesador que realiza una o más de las tareas que constituyen el trabajo.



complejo que no pasa de máquina en máquina durante su ejecución. En (Liao, et al., 2011) al tratar la programación de proyectos de construcción e ingeniería, se crea indirectamente un paralelismo entre los conceptos de fase, modo o máquina con los de sitio, lugar, o nivel. En este tipo de proyectos son necesarias instalaciones o equipos para la ejecución de las tareas que se encuentran en los distintos niveles que forman las plantas de un edificio. Las instalaciones o equipos serían semejantes a las máquinas en un proceso industrial, de manera que un edificio se podría construir con equipos en paralelo (máquinas en paralelo) considerando que en múltiples plantas, sitios o lugares del edificio se estaría realizando a la vez el mismo proceso o actividad, por ejemplo la colocación de pavimentos; o en serie, realizando secuencialmente las actividades de forma uni-modal, como sería el caso de iniciar la cimentación de un edificio tras haber realizado los trabajos de excavación o no empezar las tabiquerías hasta que esté completamente terminada la estructura. Así es posible entender cómo hacer compatibles los sectores de la ingeniería y construcción con el de la manufacturación, no solamente en temas de clasificación sino también en la aplicación de los diferentes métodos y herramientas de programación conocidos.

#### 4.2. Resolución de Problemas de Programación de Proyectos con Recursos Limitados

De acuerdo a (Soler Fernández, 2009), hay muchos problemas de optimización para los que todavía no se han encontrado algoritmos polinomiales que los resuelvan. Las investigaciones al respecto se desarrollan partiendo de tres puntos de vista: el diseño de algoritmos que busquen la solución óptima permitiendo resolver instancias que simplifiquen el problema consiguiendo que tenga un tamaño adecuado y pueda ser resuelto en un tiempo exponencial razonable, el diseño de algoritmos heurísticos con los que se podrán conseguir soluciones admisibles próximas a la óptima y por último la comparación con problemas semejantes para los que ya se disponga de una solución factible.

Dentro de los problemas de optimización encontramos un número ilimitado de problemas de programación de proyectos. Estos pueden ser del tipo determinista o estocástico. Para hacer que los problemas se simplifiquen, la mayoría de las veces se centran sobre un contexto determinista porque se ignora la incertidumbre que surge sobre la duración de las tareas (Lawler, et al., 1989). Se supone que se conoce toda la información relativa al proyecto por anticipado y con certeza, por tanto esta seguridad garantiza que existirá también un ambiente estocástico, en donde evidentemente también intervendrá el azar.

Sabemos que nuestro problema de programación de proyectos con recursos limitados, surge tras la planificación de las actividades de un proyecto al disponer de una cantidad restringida de recursos para realizar las distintas tareas. Esta barrera impide conseguir realizar el proyecto en el mínimo tiempo posible que habíamos considerado inicialmente al planificar sin considerar las restricciones de recursos. En el apartado 3.6 ya se ha representado el problema mediante la formulación de un modelo matemático que nos sirve para entender mejor el problema. Ahora hay que pensar en métodos o herramientas que se puedan utilizar para obtener una solución adecuada. Estos procedimientos no son más que algoritmos. Entre los algoritmos existentes puede que se encuentren algunos útiles para solucionar el problema y por tanto se aprovechen o también es posible diseñar uno nuevo que se ajuste a nuestros objetivos y restricciones.



¿Qué se espera de un algoritmo?:

- Ha de ser finito, debe terminar después de un número finito de órdenes.
- Ha de ser preciso, no debe existir ambigüedad en las órdenes.
- Dados unos datos de entrada tras su ejecución se han de obtener unos de salida.
- Al implementarlo deberá ser eficaz, deberá encontrarse la solución en un tiempo razonable.

Sería una labor complicada y muy minuciosa hacer una clasificación íntegra de todos los algoritmos que se han utilizado en los últimos años para resolver los problemas RCPSP, pues los métodos manejados por los investigadores son innumerables y van desde los de nueva creación y los de mejora, a los híbridos procedentes de la combinación de otros ya existentes.

La [Figura 5](#) recoge un esquema con los algoritmos más divulgados en la literatura agrupándolos en las tipologías comunes que los unen, siendo muy útil para tener una idea general de todos ellos. La información está argumentada en ([Cervantes Posada, 2010](#)), ([Agirre Pérez, 2007](#)), ([Demeulemeester & Herroelen, 2002](#)), ([Klein, 2000](#)), ([Kolisch & Hartmann, 2006](#))y ([Kolisch & Padman, 2001](#)).

<b>Métodos para resolución de problemas RCPSP</b>	<b>EXACTOS</b>	<b>Branch and Bound</b>	Árbol de precedencias			
			Alternativas de retraso			
			Alternativas de extensión			
			Schedule Schemes			
			Conjuntos prohibidos mínimos			
		<b>Programación entera</b>				
	<b>Programación dinámica</b>					
	<b>APROXIMADOS</b>	<b>Heurísticos</b>	<u>Constructivos</u>	Esquemas de Secuenciación (basados en reglas prioridad)	<i>En Serie</i>	<i>Planificación hacia delante</i>
						<i>Planificación hacia atrás</i>
						<i>Planificación bidireccional</i>
					<i>En paralelo</i>	<i>Planificación hacia delante</i>
						<i>Planificación hacia atrás</i>
						<i>Planificación bidireccional</i>
			<b>Arcos disyuntivos</b>			
			<b>Branch and Bound truncado</b>			
			<u>De mejora</u>	<b>Methheurísticos</b>	<i>A. Genéticos</i>	
<i>Técnica de las hormigas</i>						
<i>Búsqueda tabú</i>						
<i>Enjambre de partículas</i>						
<i>Enfriamiento de materiales</i>						
Aproximación descendente	<i>Máxima pendiente</i>					
	<i>Descenso rápido</i>					
	<i>Descenso iterativo</i>					
<b>Hiperheurísticos</b>						

Figura 5: Tipos algoritmos para resolución problema RCPSP.



### 4.3. Algoritmos Exactos

Los algoritmos exactos siguen la estrategia de evaluar todos los caminos posibles hasta conseguir el óptimo, esto puede representar un alto coste computacional para los problemas RCPS, es decir, puede tardarse mucho tiempo hasta obtener la mejor solución ya que el tipo de optimización de estos problemas es NP-hard (ver apartado 3.5), es muy grande el espacio de búsqueda de los resultados, la cantidad de variables y la cantidad de restricciones.

En (Brucker, et al., 1999) se comparan los resultados computacionales de diferentes instancias del problema llegando a la conclusión de que a medida que se va aumentando el número de actividades del proyecto los algoritmos de aproximación son más eficaces que los exactos (estos no llegan a resolver instancias de tamaño mediano y grande), en consecuencia es preferible utilizar los procedimientos heurísticos a los exactos.

Sin embargo, (Álvarez-Valdés Olaguibel, et al., 2009) proponen un procedimiento de reducción de las dimensiones del RCPS al que llaman **preproceso** que es válida para utilizar tanto en métodos exactos como aproximados, mediante el cual antes de empezar a resolver un problema se identifica si es un problema trivial, se trata de reducir el horizonte de la planificación, se aplican filtros sobre los recursos eliminando los ociosos y los no restrictivos, así mismo se aplican filtros sobre las variables y tras las fases anteriores se construye una cota inferior para la duración del problema con lo cual se puede resolver este como si se tratase de un problema lineal.

Entre los algoritmos exactos destacan por ser los más utilizados, los Branch and Bound o de Ramificación y Poda. Utilizan reglas de dominancia para podar las ramas que no llevan a la solución óptima y por tanto se va reduciendo el número de alternativas hasta alcanzar la solución. El éxito de estos algoritmos está en la determinación de la cota inferior<sup>26</sup>, pues cuanto más se relajan las restricciones más fácilmente se resuelve el problema pero peor es el valor de la cota que se obtiene.

### 4.4. Algoritmos Heurísticos

El primero en utilizar el término heurístico fue el matemático George Polya en su libro "How to solve it" al tratar de encontrar soluciones a determinados problemas. Los heurísticos son algoritmos basados en nuestro conocimiento del problema y en el sentido común, formados por una secuencia lógica de pasos que encuentran soluciones a nuestro problema, no necesariamente óptimas pero si razonables.

Son intuitivos y fáciles de implementar, con ellos se obtendrán secuencias de buena calidad y rápidas. Esta flexibilidad puede llevar a errores, siendo una de las dificultades de estos métodos. Además, los algoritmos heurísticos, necesitan de especialistas, es conveniente que sea más de uno, con conocimientos y que realicen previamente un trabajo de ajuste de los parámetros para que estos puedan llevar a buen término las instancias del problema que se están analizando, puesto que los resultados han de ser validados. La justificación del uso de la

<sup>26</sup> Se busca minimizar el makespan.



secuenciación se establece mediante la investigación de cómo se comportan en los problemas grandes.

Otro de los inconvenientes de las heurísticas constructivas es que algunas se pueden contradecir al aplicarse al mismo problema y esto origina confusión y pérdida de credibilidad de las heurísticas. Es imposible tener una garantía provisional de que una heurística en particular o una combinación de heurísticas, producen los mejores resultados para un problema. Así mismo, las sucesivas mejoras locales ocasionadas por las heurísticas, pueden trucar el camino que conduce al óptimo. Tampoco es adecuado estudiar el comportamiento y la complejidad del algoritmo siempre con respecto al peor caso. A pesar de estos inconvenientes las heurísticas son ampliamente utilizadas y en algunos casos se ven mejoradas por metaheurísticas e hiperheurísticas.

Pese a no garantizar la optimalidad, los métodos heurísticos de resolución (incluyendo entre ellos los procesos de mejora local y los algoritmos meta-heurísticos) son básicos por varios motivos (García Sabater & Maheut, 2011). En primer lugar son capaces de generar soluciones lo que generalmente es mejor que no tener solución alguna. En segundo lugar se puede decir que alcanzar el óptimo de un modelo que tampoco es exactamente el problema real no es esencialmente grave. Por último ser capaz de diseñar una buena heurística, exige un conocimiento del problema que puede conducir a mejoras de otro tipo.

### Esquemas de Generación de Secuencias o Esquemas de Secuenciación.

Se llama **secuencia** al conjunto de actividades ordenadas de tal modo que cada una de estas actividades está determinada por las anteriores. Una secuencia es **completa** cuando las actividades se realizan en el periodo fijado; además, una secuencia completa será **factible** cuando respete las relaciones de precedencia, las limitaciones de recursos y las fechas para completar la actividad final. Una secuencia **parcial** sería aquella en la que sólo se ha secuenciado una parte o subconjunto de esas operaciones, asignándosele un tiempo para finalizar.

Los algoritmos creados a partir de la secuenciación basada en reglas de prioridad quedan definidos por dos elementos: el esquema de generación de secuencias y una o varias reglas de prioridad. El Esquema de Generación de Secuencias o Esquema de Secuenciación, en inglés denominado Schedul Generation Scheme o Scheduling Scheme, sigue siendo el método heurístico constructivo más utilizado para resolver el RCPS. Para (Ballestín González, 2002), los esquemas de generación de secuencias construyen una secuencia factible, desarrollando progresivamente una secuencia parcial y partiendo de una secuencia vacía que será la actividad de inicio; al conjunto de actividades secuenciables se le aplica la regla o reglas de prioridad las cuales determinan el orden en que las actividades deben realizarse.

Los esquemas de secuenciación se pueden aplicar en dos direcciones, planificación o cálculo hacia delante y hacia detrás (forward and backward recursion) (Cervantes Posada, 2010). Existen dos grupos de esquemas de secuenciación: los esquemas de secuenciación en serie, a los cuales nombraremos desde este momento como SSS (Serial Scheme Scheduling), y los esquemas de secuenciación en paralelo a los que llamaremos PSS (Parallel Scheduling Scheme). A grandes rasgos, según (Navascués Fernández, s.f.) y (Kolisch & Hartman, 1999) los



SSS partiendo de una secuencia vacía van iterando e incorporando actividades y los PSS van realizando iteraciones temporales. A continuación se explican ambos tipos de esquemas siguiendo lo indicado por (Ballestín González, 2002) , (Cervantes Posada, 2010), (Klein, 2000) y (Kelley, 1963).

● Esquema de secuenciación en serie (SSS):

Una serie se compone de  $g=1,\dots,n-2$  etapas, en cada una de las cuales una actividad es seleccionada y secuenciada lo más temprano posible, respetando las precedencias y las restricciones de los recursos. Existen dos conjuntos disjuntos asociados a cada etapa  $g$ , siendo  $Sec_g$  el conjunto de actividades que ya han sido secuenciadas y que por tanto pertenecen a una secuenciación parcial, y el conjunto de actividades  $D_g$  las que son elegibles. Una actividad es elegible si no ha sido secuenciada y todas sus antecesoras sí y será seleccionada mediante una regla de prioridad. También puede haber actividades no secuenciadas y no elegibles.

Algoritmo SSS

- Se denominará secuencia o solución,  $Sol$ , a cualquier vector de tiempos de comienzo,  $Sol = \{s_1, s_2, \dots, s_n\}$  o idénticamente a cualquier vector de tiempos de finalización,  $Sol = \{f_1, f_2, \dots, f_n\}$ .
- Dada una secuencia  $Sol$ , sea  $A(t) = \{j \in V / s_j \leq t \leq f_j\} (= \{j \in V / f_j - d_j \leq t \leq f_j\})$  el conjunto de actividades que se están procesando, que están activas, en el instante  $t$ .

Definiendo  $\bar{R}_k(t)$  como la disponibilidad restante del recurso de tipo  $k$  en el instante  $t$

$\left( \bar{R}_k(t) = R_k - \sum_{j/s_j \leq t, f_j > t} r_{j,k}, k \in K \right)$  y  $F_g = \{f_j / j \in Sec_g\}$  como el conjunto de tiempos de completación, el algoritmo sería:

1. Se incorpora la actividad ficticia inicial  $j=1$ .  $s_1 = f_1 = 0, S_1 = \{1\}$
2. Para  $g = 2$  hasta  $n-1$ , hacer:
  - 2.1. Determinar  $D_g, F_g, \bar{R}_k(t) (k \in K; t \in F_g)$ .
  - 2.2. Seleccionar una actividad elegible  $j \in D_g$ .
  - 2.3. Determinar el tiempo más pronto de iniciar  $j$  compatible con las precedencias.  $ES_j = \max_{h \in Pred_j} \{f_h\}$
  - 2.4. Determinar el tiempo más pronto de iniciar  $j$  compatible con la disponibilidad de recursos.  $s_j = \min \{t / t \geq ES_j; t \in F_g; r_{j,k} \leq \bar{R}_k(\tau) \forall k \in K, \tau \in [t, t + d_j] \cap F_g\}$ .
  - 2.5.  $f_j = s_j + d_j$ .
  - 2.6.  $Sec_g = Sec_{g-1} \cup \{1\}$ .
3.  $f_n = \max_{h \in Pred_n} \{f_h\}$ .



En la primera orden se asigna el tiempo de inicio y de finalización 0 a la actividad ficticia  $j=1$  y se la incluye en la secuencia parcial. Al principio de cada etapa  $g$  se calcula el conjunto de elegibles  $D_g$  que serán todas las actividades factibles en función de las restricciones de precedencia, el conjunto de tiempos de finalización  $F_g$  y las disponibilidades restantes de recursos  $\bar{R}_k(t)$ . Después se selecciona una actividad  $j$  del conjunto de elegibles en función de una o varias reglas de prioridad. El tiempo de inicio de  $j$ , llamado  $s_j$ , se calcula: primero determinando el máximo de los tiempos más pronto de terminar de las antecesoras ( $ES_j$ ), y después fijando  $s_j$  como el mínimo entre este  $ES_j$  y el que se obtenga de considerar la disponibilidad restante de los recursos. El tiempo de finalización de  $j$ , llamado  $f_j$ , se determinará sumando al tiempo de inicio de la actividad la duración de la misma. El algoritmo termina cuando todas las actividades se encuentran en el conjunto de programadas  $Sec_g$ . Según indican (Kolisch & Padman, 2001), el conjunto de soluciones generadas siempre contará con la solución óptima.

- Esquema de secuenciación en paralelo (PSS):

El PSS intenta programar tantas actividades como sea posible en cada tiempo de decisión o secuenciación, este tiempo está relacionado con la terminación de las actividades en ejecución, no existiendo ninguna actividad factible en un periodo de tiempo que no pueda ser programada en dicho periodo. El PSS itera sobre  $n$  diferentes puntos de decisión en el que las actividades se pueden añadir a la programación, a cada iteración  $g$  le corresponde un tiempo de secuenciación  $t_g$ . Estos puntos de iteración están relacionados con los plazos de ejecución de las actividades ya programadas.  $C_g = \{j \in V / f_j \leq t_g\}$  representa el conjunto de actividades completadas o programadas hasta  $t_g$  y  $A_g = A(t_g) = \{j \in V / s_j \leq t_g \leq f_j\}$  el de activas que se están secuenciando para  $g \neq 0$ .

$A_0 = \{1\}$  y  $D_g$  es el conjunto de actividades elegibles siempre que cumplan las restricciones de la precedencias y de los recursos en un cierto tiempo  $t_g$ , es decir,  $D_g = \{j \in V \setminus (C_g \cup A_g); \text{Pred}_j \subseteq C_g \text{ y } r_{j,k} \leq \bar{R}_k(t_g) \forall k \in K\}$ . Para seleccionar una actividad se utiliza una regla de prioridad, que a su vez es programada para que se inicie en  $t_g$ . Cuando el algoritmo termina el conjunto de soluciones no siempre contiene una solución óptima.

Algoritmo PSS

1. Inicialización:  $g = 0, f_1 = 0, A_0 = C_0 = \{1\}, \bar{R}_k(0) = R_k, \forall k \in K$ .
2. Mientras  $|C_g \cup A_g| \leq n - 2$ , hacer:
  - 2.1.  $g = g + 1$ .
  - 2.2.  $t_g = \min_{j \in A_{g-1}} \{f_j\}$ .
  - 2.3. Calcular  $C_g, A_g, \bar{R}_k(t_g), D_g$ .



- 2.4. Mientras  $D_g \neq \emptyset$ , hacer
  - 2.4.1. Seleccionar  $j \in D_g$ .
  - 2.4.2.  $s_j = t_g$ .
  - 2.4.3.  $f_j = s_j + d_j$ .
  - 2.4.4. Actualizar  $\bar{R}_k(t_g), A_g$  y  $D_g$ .
3.  $f_n = \max_{h \in \text{Pred}_n} \{f_h\}$ .

### **Reglas de prioridad.**

Las reglas de prioridad se utilizan dentro de un esquema de generación de secuencias para determinar ordenadamente las actividades a elegir que serán programadas con posterioridad. Dependiendo de la información que se utilice las reglas de prioridad se clasifican en las siguientes categorías, (Álvarez-Valdés & Tamarit, 1989), (Kolisch, 1995b), (Klein, 2000) :

- Reglas de prioridad basadas en las redes, desechando la información sobre los recursos.
- Reglas de prioridad basadas en la ruta crítica, fundadas en los resultados de un pase hacia delante y hacia atrás, consideran los tiempos más pronto y más tarde de empezar o de terminar.
- Reglas de prioridad basadas en los recursos, en las limitaciones o requerimientos de estos.
- Reglas compuestas, tienen en cuenta más de un tipo de regla de prioridad para tener más información a la hora de combinarlas.

### **4.5. Algoritmos Metaheurísticos**

El término metaheurística fue utilizado por primera vez por F. Glover en el año 1986 cuando propuso el método de la búsqueda tabú. Los algoritmos metaheurísticos son métodos generales aproximados diseñados para resolver amplios problemas de optimización combinatoria difíciles en los que los heurísticos han fracasado. Las meta-heurísticas proporcionan marcos generales que permiten la creación de nuevos híbridos, que emulan y combinan diferentes conceptos derivados de las heurísticas clásicas, la inteligencia artificial, la naturaleza, la evolución biológica y la mecánica de estadística (Osman & Kelly, 1996). Exactamente (Osman & Kelly, 1996) definen como metaheurístico a un procedimiento iterativo, con una estructura y unas reglas generales de funcionamiento que lo caracterizan, que guía un método (normalmente un heurístico) subordinado combinando inteligentemente diversos conceptos para explorar los espacios de búsqueda utilizando estrategias aprendidas para conseguir soluciones cuasi-óptimas de manera eficiente.

Los algoritmos metaheurísticos tratan de diversificar las soluciones de forma que permiten la evasión de los óptimos locales, además intensifican la búsqueda guiándola en el espacio, prefiriendo los movimientos que mejoren el valor de la función objetivo. Van desde simples





procedimientos de búsqueda local a procesos de aprendizaje complejos. Por lo general son deterministas, robustos y fáciles de modificar. Son iterativos con paradas programadas por el usuario, ya que de no ser así continuarían ininterrumpidamente al no conocerse el óptimo y el proceso sería interminable. Trabajan guiando a los heurísticos para que escapen de los óptimos locales que es el principal problema con el que se encuentran los heurísticos. Posibles criterios de parada pueden ser: (1) alcanzar una cota previamente establecida, (2) una determinada cantidad de iteraciones sin mejora, (3) un tiempo máximo de proceso o (4) un tiempo máximo sin mejora. Actualmente se han desarrollado trabajos para establecer criterios de parada basados en la probabilidad de haber alcanzado el óptimo haciendo uso de la función de Weibull, ver (Yepes, et al., 2010), (Martí & Moreno Vega, 2003), (Crespo Robles, 2008).

Los metaheurísticos más tratados en la literatura son los algoritmos genéticos, introducidos por Holland en 1975 quien utilizó conceptos de la evolución biológica y su base genético-molecular para construir los algoritmos. Los algoritmos evolutivos hacen evolucionar una población de individuos sometiéndola a acciones aleatorias y a una selección de acuerdo con algún criterio, en función del cual se decide cuáles son los individuos más adaptados, que sobreviven, y cuáles los menos aptos, que son descartados. De acuerdo a (Kolisch & Hartman, 1999) el proceso es el siguiente, un algoritmo genético considera simultáneamente un conjunto o población de soluciones en lugar de sólo uno. Después de haber generado una población inicial, son producidas soluciones nuevas por acoplamiento de dos ya existentes (cruzado) y / o mediante la alteración de una existente (mutación). Después de producir nuevas soluciones, las soluciones más aptas "sobreviven" y constituyen la próxima generación, mientras que se eliminan los demás. El valor de aptitud mide la calidad de una solución, por lo general está basado en el valor de la función objetivo del problema de optimización que hay que resolver.

Según concluyen (Liao, et al., 2011) tras analizar la historia del arte de los algoritmos metaheurísticos, estos tienen un gran potencial en la programación de proyectos como técnicas de optimización para determinar soluciones casi óptimas en las distintas fases del ciclo de vida de un proyecto, pero de momento no ha sido aprovechado adecuadamente, afirmando literalmente que cualquier problema de optimización que no se ha podido solucionar con técnicas exactas o heurísticas es un firme candidato para las metaheurísticas. Así mismo, los distintos estudios que han analizado plantean ejemplos pequeños y en consecuencia aplicables sólo en un mundo ideal; además las metaheurísticas no se han utilizado todavía en paquetes de software de gestión de proyectos; en resumen, queda mucho camino por recorrer.

#### 4.6. Algoritmos Hiperheurísticos

Los algoritmos hiperheurísticos son un método que controla la elección de la heurística que debiera ser aplicada en cada punto de decisión de manera inteligente, dependiendo de las características del problema, la adecuación de las diferentes heurísticas y de la región del espacio de la solución que se está explorando, es decir, del estado del problema.



Las heurísticas clásicas y las metaheurísticas requieren una considerable experiencia en el dominio del problema y de las técnicas heurísticas adecuadas y por lo tanto son costosas de implementar. La hiperheurística requiere tener un conocimiento básico de las heurísticas y ha de ser lo suficientemente robusta como para manejar una amplia gama de problemas. Una hiperheurística es capaz de elegir entre heurísticas de bajo nivel sin la necesidad de utilizar el conocimiento del dominio, mediante el uso de indicadores de rendimiento que no son específicos para el problema. La hiperheurística elige entre todas las heurísticas, una o varias, pero siempre la que ofrezca mejores resultados. Si se encuentra una solución óptima local, diversifica la búsqueda a otras áreas del espacio de soluciones realizando una nueva y adecuada selección. (Cowling, et al., 2001).



## 5. El Problema de la Imprecisión en la Estimación de los Valores

Los términos desconocimiento, inseguridad, duda, inexactitud, confusión, equívoco, ambigüedad, indeterminado, difuso, vago, no son más que sinónimos del término **imprecisión** el cual queda definido por el la RAE como la “**falta de precisión**”. La precisión desde el punto de vista filosófico, es una abstracción o separación mental que hace que el entendimiento o pensamiento de dos cosas que se han identificado realmente, se conciben la una como distinta de la otra; si existe imprecisión es porque la información no es cuantificable, la podemos acotar pero dentro de un intervalo. Por otro lado el término **incertidumbre** se puede definir como la “**falta de certeza**”, es decir, la información es cuantificable pero no podemos estar seguros de cuáles son los parámetros porque dependen del azar, las probabilidades o la aleatoriedad.

Volviendo al tema de la imprecisión, para que dos cosas sean distintas entre ellas, poseerán unas particularidades en las que no serán coincidentes, dependerá de que tengan la menor cantidad posible de características en común y además de la valoración, del peso, o de la importancia que se les dé a dichas características. Pero ¿en qué grado serán diferentes?, ¿cómo se valorarán esas características?, ¿se pueden definir con exactitud?, pues depende. Todo lo que de alguna manera se pueda medir se conocerá con mayor o menor precisión. La temperatura de un objeto o la altura de una persona se sabrán con **precisión**, aunque con un margen de error que estará en función de la **exactitud** del aparato utilizado, eso sí, los datos obtenidos serán válidos. El calificar a los objetos como calientes o no, o a las personas como altas o no, sí será impreciso, porque con respecto a que patrón o a partir de cuándo se les otorgarán esos adjetivos.

Por ejemplo, si nos proporcionan tres huevos, uno de avestruz, uno de gallina y otro de perdiz, ¿los tres son iguales?. Desde el punto de vista de la Biología los tres son el mismo elemento, ya que a grandes rasgos los tres son unicelulares, formados por una célula eucariótica con un núcleo interior que es la yema, rodeada por el albumen o clara y todo ello protegido mediante una membrana o cáscara. Si con uno de ellos hay que dar de comer a 8 comensales, aunque biológicamente los tres tipos sean semejantes, es evidente que ya no nos parecerán iguales; se podría decir con bastante exactitud que con el de gallina o el de perdiz no sería posible, en cambio el de avestruz sí podría ser una solución factible; pero, ¿si tuviésemos 8 huevos de gallina? o ¿si tuviésemos sólo 1 comensal?, entonces seguramente sí podríamos cumplir nuestro objetivo. Luego para que el problema tenga una solución viable, depende de los gramos de huevo que el anfitrión **estime** necesarios por comensal, no de dónde procedan ni del número disponible de ellos. En consecuencia, como podemos ver en este ejemplo siempre habrá un grado de imprecisión porque no se puede medir con exactitud el estado de satisfacción de cada individuo; el estimar una determinada cantidad no asegurará la precisión en las cuantías, por tanto la **evaluación** de los **valores** será **imprecisa**, todo dependerá de la bondad de las decisiones que tomemos. Como veremos más adelante, existen criterios para valorar nuestras decisiones y así poder compararlas con otras.





## 6. La Limitación de Recursos con Valores Imprecisos

### 6.1. Imprecisión vs Incertidumbre

En este apartado partiendo de las definiciones iniciales del anterior, se intenta argumentar la diferencia entre imprecisión e incertidumbre, además de cómo ambas se complementan; estos dos términos serán de utilidad para abordar los conceptos de la lógica fuzzy que se analizarán más adelante en el [apartado 6.2](#). Básicamente se puede decir que la lógica fuzzy utiliza valores imprecisos emitiendo juicios convencionales para instancias que no quedan solucionadas con un simple verdadero/falso, si/no, etc., pues requieren de calificaciones intermedias.

Para definir un problema es necesario determinar unos parámetros a partir de unas variables o constantes que han de ser medibles. En función de los distintos valores que estos parámetros tengan se obtendrán distintos casos o instancias de un problema. Cuando no puede darse un valor exacto a esos parámetros, la información no es cuantificable, no es completa o no puede obtenerse, se dice que se está en un contexto de **imprecisión** o de falta de precisión (ver [apartado 5](#)). De acuerdo a ([Ponz Tienda, 2010](#)) para la programación de un proyecto se harán estimaciones que estén muy próximas a la realidad sin embargo siempre habrá un margen de imprecisión; como ejemplo, si tenemos que realizar la losa de cimentación de un edificio se puede estimar que se tardarán 40 días en realizarla, no obstante, no se puede afirmar que exactamente y sin ningún margen de error será ese el periodo de tiempo necesario, ¿por qué no podrían ser 38 o 41 días?.

Cuando los parámetros que intervienen son estocásticos<sup>27</sup>, es decir, cuando están sometidos a influencias aleatorias, nos encontramos ante un ambiente de **incertidumbre o riesgo**. Las variables aleatorias toman valores en un espacio probabilístico o conjunto en el que se encuentran todas las posibilidades. Si por ejemplo disponemos de dos equipos (equipo A y equipo B) para realizar las seis plantas de la tabiquería de un edificio; supongamos que decidimos que trabajarán en plantas diferentes y que la primera planta la realizará el primer equipo que llegue a obra el día en que se inicie el trabajo y la segunda planta, la comenzará ese mismo día el que llegue después; en consecuencia se decidirá aleatoriamente que equipo trabajará en la primera planta, existiendo una probabilidad del 50% de que sea el equipo A, o un 50% de que se trate del equipo B y dependerá de variables que nosotros no podemos controlar como la hora de salida hacia el puesto de trabajo, el tráfico con el que se encuentren, la avería de un semáforo, etc., es decir, la decisión estará regida por el azar.

La imprecisión en la toma de decisiones suele modelarse con **la teoría de conjuntos difusos o teoría fuzzy**; mientras que la incertidumbre o riesgo se tratan **con la teoría de probabilidades**, la cual permite modelar el conocimiento incompleto del ambiente. Por

<sup>27</sup> Lo contrario de los parámetros estocásticos serían los parámetros deterministas.



ejemplo, la optimización no es más que la toma de unas decisiones entre unas alternativas factibles, y una toma correcta de decisiones es elegir entre lo posible lo mejor.

También debe distinguirse entre incertidumbre y riesgo, pues no son lo mismo. Incidiendo en lo comentado con anterioridad, a la hora de tomar decisiones para solucionar un problema en donde intervienen probabilidades y resultados aleatorios, ver (Vitoriano Villanueva, 2009), estos no están controlados debido al azar o la falta de información, en consecuencia, nos hallamos ante un entorno de optimización regido por “la teoría de la decisión bajo incertidumbre o riesgo”, en este contexto los efectos de una decisión no están determinados de antemano; por tanto se está hablando de optimización estocástica o programación bajo incertidumbre. Las probabilidades para que se dé un determinado conjunto de escenarios en ocasiones **no son conocidas**, se trata pues de un proceso de **decisiones bajo incertidumbre**; si las probabilidades son **conocidas** o se pueden **estimar**, entonces se habla de **decisiones bajo riesgo**. Lógicamente la incertidumbre y el riesgo afectarán a los modelos utilizados para resolver los problemas y por tanto habrá que estudiar cómo solventarlos.

Tradicionalmente la teoría de la probabilidad se ha considerado como el método adecuado para tratar la incertidumbre y la imprecisión, en (Zadeh, 1996a) L.A. Zadeh defiende que aunque la teoría de probabilidades y la lógica difusa son diferentes, la teoría de probabilidades no es suficiente por sí sola para hacer frente a la incertidumbre, y que la teoría de probabilidades y la lógica difusa se complementan y no se contraponen. Ambas tienen programas y dominios de aplicación diferentes, si bien la teoría de probabilidades puede representar de forma restringida un conjunto difuso como un conjunto aleatorio o combinación de conjuntos no-fuzzy; asimismo hay muchas aplicaciones de la teoría de probabilidades donde la lógica fuzzy se ha utilizado con éxito. No obstante, la teoría de probabilidades tiene limitaciones para analizar los problemas descritos con términos fuzzy, no es lo suficientemente expresiva para representar el lenguaje humano, no posee métodos para estimar probabilidades o cuantificadores fuzzy; sin embargo es mucho más eficaz cuando las dependencias entre variables no están bien definidas. En estas cuestiones es donde deben complementarse.

En el enfoque descrito en (Zadeh, 1996a), se demuestra cómo las probabilidades se pueden estimar a partir de datos difusos. La manipulación de las probabilidades difusas en general requiere el uso de la aritmética difusa, y muchas de las propiedades de las probabilidades difusas son generalizaciones simples de las propiedades de las probabilidades de valor real<sup>28</sup>.

---

<sup>28</sup> Los números reales incluyen los números enteros, los racionales y los irracionales.



## 6.2. La lógica Fuzzy

“No hay nada fuzzy en la Lógica Fuzzy”

**Lofti A. Zadeh** *profesor Universidad de Berkeley (California)*

### 6.2.1. Introducción a la Lógica Fuzzy.

Porque las cosas no son simplemente blancas o negras, todo depende del cristal con que se las mire, la capacidad de pensar del hombre y su interés por reflejar el mundo real ha contribuido a buscar una lógica que fuese más allá del clasicismo aristotélico en el cual los umbrales de decisión quedan bien definidos. Esta es la llamada LÓGICA FUZZY, LÓGICA BORROSA o LÓGICA DIFUSA.

La lógica fuzzy se encarga de representar las proposiciones del lenguaje humano con innumerables matices que no tienen un significado preciso, que no responden a un SI/NO o a un VERDADERO/FALSO. Trata de llevar la riqueza de las expresiones semánticas a ámbitos científicos como la informática, las matemáticas, la física, etc., con la intención de describir las situaciones complejas y dinámicas, haciendo uso de razonamientos cualitativos; de este modo la ciencia se acerca al pensamiento humano.

**No hay nada difuso en la Lógica Difusa**, simplemente permite que un elemento tenga un grado de pertenencia dentro de un intervalo cerrado  $[1,0]$  en lugar de sólo dos opciones o valores absolutos, el “0” y el “1” ( $\{0,1\}$  falso y verdadero), por tanto, se puede representar con precisión el elemento asociándole un número que se encontrará entre un límite inferior 0 y un límite superior 1. ¿Se puede decir cuál es el límite entre la locura y la cordura de una persona?, ¿dónde está la línea que las separa?, ¿cómo de loca está una persona?; podríamos hacer una encuesta entre cientos de individuos y nunca obtendríamos una respuesta perfectamente definida, para llegar a una conclusión imprecisa pero válida podemos tratar el tema con las herramientas de la lógica fuzzy.

Consecuentemente, la lógica fuzzy hace uso de la imprecisión para enfrentarse a los problemas de la realidad logrando una mayor eficiencia que la lógica clásica. Como en su día comentó el profesor Zadeh ([Zadeh, 1979](#)):

*“Uno de los objetivos de la teoría de conjuntos difusos es el desarrollo de una metodología para la formulación y solución de los problemas que son demasiado complejos o que están mal definidos y que no se pueden analizar mediante técnicas convencionales... La teoría de los conjuntos borrosos se reconocerá como un desarrollo natural en la evolución del pensamiento científico. El escepticismo sobre su utilidad será visto como una manifestación del apego humano a la tradición y la resistencia a la innovación”.*

No vivimos en un mundo ideal y este ha cambiado bastante desde que Aristóteles, padre de la lógica, escribió sus cerca de 200 magníficos tratados relacionados con la ética, la filosofía y



las ciencias. La lógica fuzzy ofrece la oportunidad de complementar y mejorar la lógica clásica o nítida<sup>29</sup>, ¿por qué no aprovecharla?.



Figura 6: Esquema Lógica. Fuente (Trillas, et al., 1994)

La lógica fuzzy es muy útil cuando se necesita la experiencia del hombre para conseguir un objetivo. Hoy en día está reconocida y es aplicada con éxito, centrándose sobre todo en tres campos, (Trillas, et al., 1994):

- Sistemas expertos: programas para la toma de decisiones, emulan a un experto en campos específicos, son una rama de la inteligencia artificial. Tratan de generalizar el modelo de los factores de certeza para permitir la imprecisión no probabilística. Una muestra son los programas de diagnóstico en medicina, reparación y diagnóstico de automóviles, etc.
- Control automático: es una aplicación especializada de los sistemas expertos. Manejan predicados imprecisos para la descripción de reglas. Se encuentran en domótica, electrónica, robótica....
- Reconocimiento de formas o de patrones: utilizado en previsiones meteorológicas, reconocimiento de voz, predicción de terremotos...

#### **Un poco de historia – el origen.**

La lógica de Aristóteles estuvo oculta durante cientos de años, comenzando a conocerse a partir del siglo XII y permaneciendo hasta la primera mitad del siglo XX. En la ciencia moderna,

<sup>29</sup> Para L.A. Zadeh el concepto “nítido”, sinónimo de “clásico”, es la contraposición al concepto de “fuzzy”, “borroso” o “difuso”.





después de que George Boole (1815-1864) se encargase de los razonamientos que presentan las formulaciones precisas (lógica bivalente), George Cantor (1845-1918) crea la teoría de conjuntos empezando entonces a desarrollarse las lógicas polivalentes, las cuales tienen en cuenta las probabilidades, y donde la precisión es sustituida por la aproximación (Trillas, et al., 1995).

En 1965 Lotfi Asker Zadeh (Zadeh, 1996a), matemático, ingeniero, informático y profesor de la Universidad de Berkeley pone los cimientos de la lógica borrosa. Europa comenzó a tener interés por la Lógica fuzzy a partir de los años 70, en donde se usó por primera vez para el control automático de un horno de cemento. El profesor Enric Trillas fue uno de los pioneros en España en iniciar la investigación de la lógica fuzzy y sus aplicaciones. Así mismo, unos de los primeros en aplicar los principios de la lógica fuzzy a los productos de consumo (curiosamente fue algo tan sencillo como un cabezal de ducha) fueron los japoneses en 1987 que paralelamente implantaron la lógica fuzzy en el Sistema de Metro de Sendai, (Zadeh, 1996b).

### **Definiciones.**

#### **6.2.1.1. Lógica fuzzy, borrosa o difusa.**

La Lógica fuzzy, borrosa o difusa es el razonamiento matemático que permite calcular de forma exacta las magnitudes correspondientes a conceptos vagos o situaciones poco previsibles para poder tener control sobre ellas, (Hernández Negrín & I., 1997). La lógica difusa facilita la modelización de situaciones relacionales, que presentan vaguedad de forma intrínseca; es la modelización del razonamiento del sentido común, reflejando su carácter cualitativo, (Trillas, et al., 1994). Según la RAE la lógica difusa es la que admite una cierta incertidumbre entre la verdad o la falsedad de sus proposiciones para la toma de decisiones a semejanza del raciocinio humano.

#### **6.2.1.2. Conjunto borroso o difuso.**

Conjunto de objetos en donde la transición de la pertenencia es gradual y no brusca, es decir, es continua. Se escribe como  $\tilde{A}$ .

#### **6.2.1.3. Grado de pertenencia.**

Se considera que es un número entre 0 y 1 de manera que dado un elemento de un conjunto difuso, el 1 representa la pertenencia total al conjunto y el 0 la no pertenencia total. Entre el 1 y el 0 va reduciéndose de manera continua el grado de pertenencia que pueden tener los elementos. Cualquier elemento puede pertenecer a varios conjuntos en distintos grados.

#### **6.2.1.4. Función de pertenencia.**



La que asocia a cada elemento su grado de pertenencia a un subconjunto difuso. Se denota como  $\tilde{A}(x)$ .

#### 6.2.1.5. Número difuso.

Un número difuso es un conjunto difuso, convexo y normal, con función de pertenencia continua a trechos.

#### 6.2.1.6. Alfa-corte ( $\alpha$ ).

Determina el rango o conjunto de valores que conforman un determinado grado de dependencia, por tanto establece la precisión o robustez del resultado al determinar por ejemplo entre que valores se encuentran los costes, el consumo de recursos, la producción, los plazos de las tareas, etc. Lo que se hace es ir tomando distintas “rebanadas” de alfa-cortes para aproximarse al valor 1, en el cual no hay incertidumbre. Se escribe como  $\tilde{A}[\alpha]$ .

#### 6.2.1.7. Granulación.

El proceso de formar clases borrosas de objetos que están agrupados por semejanza.

#### 6.2.1.8. Variable difusa.

La relacionada con una medida de posibilidad.

#### 6.2.1.9. Instrucción difusa.

Aquella orden que contiene un concepto que debe ser representado por medio de un subconjunto difuso.

#### 6.2.1.10. Algoritmo difuso.

Conjunto ordenado de instrucciones, alguna de las cuales o todas son difusas, y cuya ejecución proporciona una solución aproximada a un problema específico.

#### Formulación matemática.

Siendo  $\Omega$  un conjunto, un subconjunto fuzzy  $\tilde{A}$  de  $\Omega$  queda definido por una función de pertenencia designada como  $\tilde{A}(x)$  que tomará valores en el intervalo  $[0, 1]$ . Si  $\tilde{A}(x)$  siempre es igual a 1 o a 0, entonces se obtiene un subconjunto nítido (no-fuzzy) de  $\Omega$ .

$$\tilde{A} = \{(x, \tilde{A}(x)), x \in \Omega\} \quad [6.1]$$



$\tilde{A}(x): \Omega \rightarrow [0,1]$	$\forall x \in \Omega$
Si $\tilde{A}(x_0) = 1$	$x_0 \in \tilde{A}$
Si $\tilde{A}(x_1) = 0$	$x_1 \notin \tilde{A}$
Si $\tilde{A}(x_2) = 0.6$	$x_2$ tiene un grado de pertenencia a $\tilde{A}$ de 0.6

Un número difuso casi siempre es de forma triangular o trapezoidal. Un número triangular difuso  $\tilde{A}$  queda definido por tres valores  $a_1 \leq a_2 \leq a_3$ , siendo la base el intervalo  $[a_1, a_3]$  (con un grado de pertenencia de 0) y su vértice está en  $a_2$  (con un grado de pertenencia 1). El núcleo del número difuso sería  $a_2$ , el soporte inferior  $a_1$  y el superior  $a_3$ .

$$\tilde{A}(x) = (a_1, a_2, a_3) \tag{6.2}$$

Se llama Alfa - corte ( $\alpha$ ) de un subconjunto difuso  $\tilde{A}$  al subconjunto de  $\tilde{A}$  en el que sus elementos tienen una función de pertenencia mayor o igual que  $\alpha$ . La fórmula [6.4] muestra con se obtiene el Alfa-corte de un número triangular difuso.

$$\tilde{A}[\alpha] = \{x \in \Omega \mid \tilde{A}(x) \geq \alpha\} \quad 0 \leq \alpha \leq 1 \tag{6.3}$$

$$\tilde{A}[\alpha] = (Si[\alpha], a_2, Ss[\alpha]) = (a_1 + \alpha(a_2 - a_1), a_2, a_3 - \alpha(a_3 - a_2)) \tag{6.4}$$

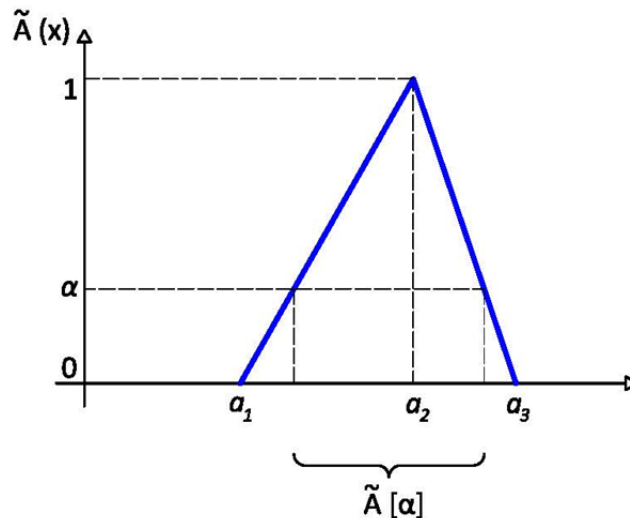


Figura 7: Número Difuso Triangular  $\tilde{A}$  y Alfa-corte  $\tilde{A}[\alpha]$  del número difuso.

Un número difuso trapezoidal  $\tilde{M}$  queda definido por cuatro valores  $a_1 \leq a_2 \leq a_3 \leq a_4$ , donde la base mayor del trapecio es el intervalo  $[a_1, a_4]$  (con un grado de pertenencia 0) y la base menor es el intervalo  $[a_2, a_3]$  (con un grado de pertenencia 1).

$$\tilde{M}(x) = (a_1, a_2, a_3, a_4) \tag{6.5}$$

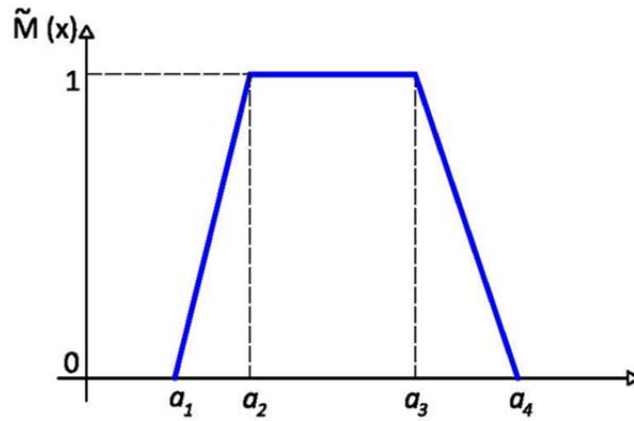


Figura 8: Número Difuso Trapezoidal  $\tilde{M}$

### 6.2.2. Aritmética y ordenación de los números difusos.

Con los números difusos se pueden hacer las mismas operaciones aritméticas que con los números nítidos para resolver problemas haciendo uso del procedimiento de los Alfa-cortes e intervalos aritméticos (Moore, 1979) (Neumaier, 1990), ya que con el método del “principio de extensión” de Zadeh resulta bastante complicado realizar los cálculos. Las operaciones aritméticas se representan con el símbolo #. (Ponz Tienda, et al., 2012) (Ponz Tienda, 2010).

$$\begin{aligned} \tilde{A}(\alpha) \# \tilde{B}(\alpha) &= \tilde{C}(\alpha) = (c_1, c_2, c_3) = \\ &= \begin{cases} c_1 = \min(a_1 \# b_1, a_1 \# b_3, a_3 \# b_1, a_3 \# b_3), \\ c_2 = a_2 \# b_2, \\ c_3 = \max(a_1 \# b_1, a_1 \# b_3, a_3 \# b_1, a_3 \# b_3). \end{cases} \end{aligned} \quad [6.6]$$

La fórmula [6.6] se puede simplificar como se indica a continuación siempre que  $\{a_1, a_2, a_3, b_1, b_2, b_3\} \geq 0$  :

$$\tilde{A}(\alpha) \oplus \tilde{B}(\alpha) = (a_1 + b_1, a_2 + b_2, a_3 + b_3), \quad [6.7]$$

$$\tilde{A}(\alpha) \ominus \tilde{B}(\alpha) = (a_1 - b_3, a_2 - b_2, a_3 - b_1), \quad [6.8]$$

$$\tilde{A}(\alpha) \otimes \tilde{B}(\alpha) = (a_1 \times b_1, a_2 \times b_2, a_3 \times b_3), \quad [6.9]$$

$$\tilde{A}(\alpha) \oslash \tilde{B}(\alpha) = (a_1 \div b_3, a_2 \div b_2, a_3 \div b_1). \quad [6.10]$$

Se debe tener en cuenta que los resultados aritméticos de la lógica difusa, nos pueden confundir puesto que  $\tilde{A} - \tilde{A} \neq 0$  y  $\tilde{A} \div \tilde{A} \neq 1$ . Véase a continuación en la Figura 9 la suma de dos números difusos cualesquiera  $\tilde{A} = (2, 5, 6)$  y  $\tilde{B} = (3, 4, 7)$  que confirma la sencillez de operar con este tipo de números; en el cuadro se han obtenido alfa-cortes de los dos números a sumar y posteriormente se ha sumado cada uno de ellos; de igual forma se podría haber calculado el resultado de la suma  $\tilde{S}$ , utilizando sólo los soportes y núcleo de los números sin necesidad de obtener alfa-cortes; estos alfa cortes son necesarios para poder definir el número cuando el



resultado que se obtiene no es un número triangular sino un número más complejo como es el caso por ejemplo del producto o el cociente de dos números difusos.

	$\tilde{A}(\alpha)$			$\tilde{B}(\alpha)$			$\tilde{A}(\alpha) \oplus \tilde{B}(\alpha)$		
$\alpha$	i	n	s	i	n	s	i	n	s
0	2	5	6	3	4	7	5	9	13
0,1	2,3	5	5,9	3,1	4	6,7	5,4	9	12,6
0,2	2,6	5	5,8	3,2	4	6,4	5,8	9	12,2
0,3	2,9	5	5,7	3,3	4	6,1	6,2	9	11,8
0,4	3,2	5	5,6	3,4	4	5,8	6,6	9	11,4
0,5	3,5	5	5,5	3,5	4	5,5	7	9	11
0,6	3,8	5	5,4	3,6	4	5,2	7,4	9	10,6
0,7	4,1	5	5,3	3,7	4	4,9	7,8	9	10,2
0,8	4,4	5	5,2	3,8	4	4,6	8,2	9	9,8
0,9	4,7	5	5,1	3,9	4	4,3	8,6	9	9,4
1	5	5	5	4	4	4	9	9	9

i=Soporte Inferior; n=Núcleo; s=Soporte Superior

Figura 9: Suma de dos números difusos

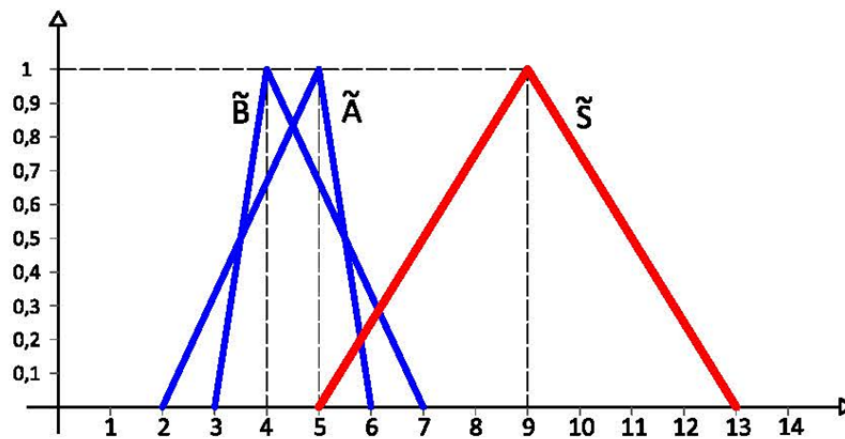


Figura 10: Representación gráfica de la suma de números difusos

Una dificultad a la hora de trabajar con números difusos es su ordenación. Por ejemplo si se dispone de dos números difusos triangulares  $\tilde{A}$  y  $\tilde{B}$  para decidir cuál es mayor y cuál menor, se calcularán los baricentros<sup>30</sup> de cada triángulo, determinándose que aquel cuyo baricentro sea mayor será así mismo el mayor de los dos números.

<sup>30</sup> El baricentro de un triángulo es el centro de gravedad del triángulo o punto en donde se cortan sus tres medianas.



Dados dos números difusos  $\tilde{A} (a_1, a_2, a_3)$  y  $\tilde{B} (b_1, b_2, b_3)$ , sus baricentros se hallarán haciendo la media ponderada de los extremos superior, inferior y el núcleo:

$$Bar(\tilde{A}) = \frac{a_1 + 2a_2 + a_3}{4}$$

$$Bar(\tilde{B}) = \frac{b_1 + 2b_2 + b_3}{4}$$

Obtenidos los baricentros de los triángulos se aplicarán las siguientes condiciones:

1.  $Bar(\tilde{A}) < Bar(\tilde{B})$ , entonces  $\tilde{A} < \tilde{B}$ ,
2.  $Bar(\tilde{A}) > Bar(\tilde{B})$ , entonces  $\tilde{A} > \tilde{B}$ ,
3.  $Bar(\tilde{A}) = Bar(\tilde{B})$ ; entonces si
 

{	$a_3 > b_3$ , entonces $\tilde{A} > \tilde{B}$ , $a_3 = b_3$ y $a_1 < b_1$ entonces $\tilde{A} < \tilde{B}$ , $a_3 < b_3$ , entonces $\tilde{A} < \tilde{B}$ .
---	--

Veamos un ejemplo con los números anteriormente utilizados  $\tilde{A} = (2, 5, 6)$  y  $\tilde{B} = (3, 4, 7)$ :

$$Bar(\tilde{A}) = \frac{2 + 2 \cdot 5 + 6}{4} = 4,5$$

$$Bar(\tilde{B}) = \frac{3 + 2 \cdot 4 + 7}{4} = 4,5$$

Como los baricentros son iguales y  $a_3 < b_3$  entonces  $\tilde{A} < \tilde{B}$

Al llegar a este punto nos preguntamos cómo se compararía un número nítido con un número difuso. Los números nítidos no son más que un caso especial de números difusos en donde el soporte inferior, el núcleo y el soporte superior son coincidentes, por tanto se pueden representar gráficamente mediante un segmento; comparando los segmentos de números nítidos con el baricentro de los triángulos que forman los números difusos se consigue determinar la ordenación de mayor a menor entre ellos. En este caso se supone que se dispone de un número nítido  $\tilde{C} = (12, 12, 12)$  y un número difuso  $\tilde{D} = (6, 11, 13)$ :

$$Bar(\tilde{D}) = \frac{6 + 2 \cdot 11 + 13}{4} = 10,45$$

Como  $10,45 < 12$  se concluye que  $\tilde{D} < \tilde{C}$ . [6.11]

6.3. El Estado del Arte de la Limitación de Recursos con Valores Imprecisos

Inicialmente (Prade, 1979) aplica la teoría de conjuntos difusos para el desarrollo de un programa de un trimestre académico en una escuela francesa, utilizando la lógica fuzzy en un problema de programación. (Chanas & Kamburowski, 1981) incorporan por primera vez los



números difusos en un RCPSP. (Lorterapong, 1994) presenta una programación de proyectos con recursos limitados utilizando un método que se centra en el tiempo estimado de terminación del proyecto, la utilización de recursos y la interrupción de los recursos. (Hapke & Slowinski, 1996) muestran una generalización del método heurístico de reglas de prioridad con parámetros de tiempo imprecisos. (Kim, et al., 2003) desarrollan un algoritmo genético híbrido con control de lógica difusa para resolver un problema RCPSP, su enfoque se basa en el diseño de los operadores genéticos con control de lógica difusa y la utilización de una heurística constructiva en serie. (Dubois, et al., 2003) explican cómo el utilizar restricciones flexibles en la programación es una manera de hacer frente a las limitaciones de los problemas de programación y aplica esta flexibilidad en el caso en que se tienen recursos limitados. (Chen & Huang, 2007) hacen uso de un método representando las duraciones de las actividades como variables lingüísticas, teniendo en cuenta el tiempo de finalización y el grado de criticidad. (Liberatore, 2008) desarrolla un algoritmo de tres pasos para la programación de proyectos aplicándolo a un conjunto de 110 problemas de gestión de proyectos, en particular a problemas de recursos limitados, utiliza una nueva metodología para el análisis de la ruta crítica difusa coherente con el principio de extensión de Zadeh. (Bhaskar, et al., 2011) presentan una heurística para RCPSP basada en una regla de prioridad para el esquema de generación de programación paralela donde para el cálculo de la ruta crítica se requiere la comparación de números difusos. (Masmoudi & Haït, 2013) ofrecen en ambientes bajo incertidumbre una nueva técnica para tratar con recursos limitados en la programación de tareas; sin aplicar alfa-cortes en un Gantt para obtener planes de recursos deterministas y considerando como difusos tanto el Gantt como el plan de carga de trabajo a nivel de programación.

### 6.4. Los valores difusos para el RCPSP

Antes de pasar al siguiente punto del trabajo en donde se analizará un caso de estudio del problema tipo RCPSP, es preciso definir como se deben tratar algunos de los valores que son necesarios para poder resolver el problema mediante la utilización de la lógica fuzzy:

- Duración Difusa ( $\tilde{d}_j$ ).

La duración difusa normal de una tarea o  $\tilde{d}_j$ , está formada por un núcleo  $d_{2j}$  en donde la función de pertenencia de los elementos del conjunto difuso es la unidad, es decir, tiene un grado de pertenencia de 1, y dos soportes  $d_{1j}$ ,  $d_{3j}$ , que respectivamente son el soporte inferior y el superior en donde la función de pertenencia tiene un valor de 0. La duración se denota como  $\tilde{d}_j = (d_{1j}, d_{2j}, d_{3j})$ .

- Tiempo difuso más pronto y más tarde de empezar una tarea ( $\tilde{ES}_j$  y  $\tilde{EF}_j$ ).

El tiempo difuso más pronto de empezar una tarea  $j$  será el mayor de los tiempos más pronto de terminar de sus actividades predecesoras. El tiempo difuso más pronto de terminar una tarea  $j$  se obtendrá sumándole al tiempo difuso más pronto de empezar la duración difusa de dicha actividad:



$$\widetilde{ES}_j = (es_{1j}, es_{2j}, es_{3j}) = \max \widetilde{EF}_i; \quad \forall i \in \text{Pred}_j \quad [6.12]$$

$$\widetilde{EF}_j = (ef_{1j}, ef_{2j}, ef_{3j}) = \widetilde{ES}_j \oplus \tilde{d}_j \quad [6.13]$$

- Tiempo difuso más tarde y más pronto de terminar una tarea  $(\widetilde{LF}_i$  y  $\widetilde{LS}_i)$ .

El tiempo difuso  $(\widetilde{LF}_i)$  más tarde de terminar una tarea  $i$  será el menor de los tiempos más tarde de terminar de sus actividades sucesoras. El tiempo difuso  $(\widetilde{LS}_i)$  más tarde de empezar una tarea  $i$  se obtendrá restándole al tiempo difuso más tarde de terminar la actividad la duración difusa de dicha actividad:

$$\widetilde{LF}_i = (lf_{1i}, lf_{2i}, lf_{3i}) = \min \widetilde{LS}_j; \quad \forall j \in \text{Suc}_i \quad [6.14]$$

$$\widetilde{LS}_i = (ls_{1i}, ls_{2i}, ls_{3i}) = \widetilde{LF}_i \ominus \tilde{d}_i \quad [6.15]$$





## 7. Aplicación de la Lógica Fuzzy al RCPSP, un Caso de Estudio

### 7.1. La PSPLIB – “Project Scheduling Library”-Biblioteca de Problemas Programación

Los investigadores fueron desarrollando sus propias instancias de problemas RCPSP para poder comparar entre si los distintos procedimientos exactos y heurísticos que iban surgiendo, pero llegó un momento en el que el número de estos algoritmos era tan elevado que se hacía muy difícil el poder evaluarlos y confrontarlos. Era complicado para los expertos el tener conocimiento de todas las mejoras de las heurísticas que aparecían al encontrarse dispersas por toda la bibliografía, pero había que aprovechar todo el potencial que estas presentaban.

De ahí surgió la necesidad de crear librerías en las que se reúnen y catalogan distintas instancias que sirven como patrones sobre los que aplicar los diferentes procedimientos, no son más que generadores de problemas test. De este modo los investigadores, pueden exponer y valorar la suficiencia de los algoritmos que proponen a través de instancias que proceden de las mismas fuentes, facilitándose de este modo la experimentación.

(Kolisch & Sprecher, 1996) presentan la **PSPLIB**, un **conjunto de instancias de referencia** para la evaluación de los procedimientos que solucionan los RCPSP de un solo modo o multi-modos, pero sus ideas pueden aplicarse además a otros problemas de optimización. La PSPLIB cuenta con problemas de diferentes tamaños, de 30, 60, 90 y 120 actividades; cada uno de estos distintos tamaños crea un conjunto de instancias dentro de la PSPLIB. Los problemas de 30 actividades se pueden resolver con métodos exactos, mientras que los otros deben ser resueltos mediante procedimientos aproximados.

Los casos son **creados por el generador estándar ProGen** el cual concibe proyectos de forma aleatoria pero de manera que la instancia resultante cumple una serie de restricciones introducidas a priori (Ballestín González, 2002). Con ProGen los investigadores pueden descargar las instancias que necesitan para testear sus algoritmos y además pueden añadir a la librería sus resultados para que estén a disposición de la comunidad según se va mejorando la calidad de los resultados obtenidos. Así es como la librería va aumentando y enriqueciéndose progresivamente. Existen variantes de ProGen, conocidas como ProGen/max y ProGen/ $\pi$ x, también se conocen otros generadores como DAGEN, RanGen y RanGen2 (Hartmann & Briskorn, 2010). ProGen utiliza un diseño factorial de experimentos que está basado en **dos subconjuntos de parámetros** para cada conjunto de pruebas (Cervantes Posada, 2010) (Kolisch, et al., 1995a) (Hekimoglu, 2007) ; se necesita de estos parámetros para clasificar el problema y ver la tratabilidad de la solución (Kolisch, 1999), (Ballestín González, 2002):

a) **Parámetros base** (no cambian de instancia a instancia):

- Número de actividades. En ProGen se encuentran 4 tipos de niveles en los problemas, proyectos con 30, 60, 90 y 120 actividades.
- Número de modos en que cada actividad puede realizarse.
- Número y tipo de recursos necesarios para el procesamiento de las actividades.



- Disponibilidad máxima de cada tipo de recurso en toda la duración del proyecto por periodo de tiempo.
- Número de sucesoras de la actividad ficticia de inicio.
- Número de predecesoras de la actividad ficticia de fin.
- Número de sucesoras y predecesoras del resto de actividades.
- Duración de las actividades.

b) **Parámetros variables** (cambian de instancia a instancia):

- La Complejidad de la red (Network Complexity o NC) mide el número medio de relaciones de precedencia no redundantes por cada actividad. Depende de la cantidad de relaciones de predecesoras de la red. Cuanto más bajo sea NC, menos relaciones de precedencia se tendrán. Los factores NC para el RCPSP son {1.5, 1.8, y 2.1}, NC se representa como:

$$NC = \frac{1}{n} \sum_{j=2}^n |\text{Pred}_j|$$

- El Factor de Recurso (Resource Factor o RF) representa la cantidad media de los recursos requeridos por actividad. Si RF=1 quiere decir que cada actividad utiliza todos los tipos de recursos, si RF =0 estamos ante un problema sin limitaciones de recursos. Los factores de RF para el RCPSP son {0.25, 0.5, 0.75, 1}, RF se representa:

$$RF = \frac{1}{(n-2) \cdot K} \sum_{j=2}^{n-2} \sum_{k \in K} \begin{cases} 1 & \text{si } r_{j,k} > 0 \\ 0 & \text{en otro caso} \end{cases}$$

- La Intensidad de los Recursos (Resource Strength RS) refleja la relación entre la demanda de recursos y la disponibilidad, es decir, el promedio disponible de cada tipo de recurso. Al igual que el factor RF, tiene un valor entre 1 y 0; influye en la dificultad de la instancia de manera que cuando más pequeño es el RS, mayor es su dificultad (Cervantes Posada, 2010). Un RS = 0 implica que por lo menos una actividad necesita la cantidad máxima disponible de un tipo de recurso, mientras que RS = 1 corresponde a la ausencia de la limitación de recursos. Se determina la demanda mínima de cada recurso  $R_k^{\min}$  para que exista una solución factible y se calcula como  $\max\{r_{j,k}, j \in V\}$ ; además se establece la demanda máxima  $R_k^{\max}$  en la secuencia ES (Early Start), donde cada actividad esta secuenciada lo más pronto posible, la secuencia ES se obtiene con cálculos hacia delante o forward pass. Los factores RS para el RCPSP en j120 son {0.1, 0.2, 0.3, 0.4, 0.5 }, en tanto que para j30, j60 y j90 son {0.2, 0.5, 0.7, 1}. RS se representa como:

$$RS_k = \frac{R_k - R_k^{\min}}{R_k^{\max} - R_k^{\min}} \quad K = 1, \dots, K$$



Por tanto, la disponibilidad  $R_k$  se obtiene despejando en la fórmula anterior.

$$R_k = R_k^{\min} + RS_k (R_k^{\max} - R_k^{\min})$$

Dando unos valores mínimos y máximos a los parámetros base, un valor a cada uno de los parámetros variables (ya indicados anteriormente) y un margen de error para estos valores, ProGen intenta generar una instancia en donde los parámetros variables sean igual a los solicitados teniendo en cuenta los márgenes de error, enviando un mensaje de error si no lo consigue. Todas las instancias de los 4 conjuntos de problemas de ProGen, j30, j60, j90 y j120 son generadas a partir de los siguientes máximos y mínimos de los parámetros base (Kolisch, 1999), (Ballestín González, 2002) :

	$K$	$d_j, j \neq 1, n$	$Q$	$r_{j,k}, j \neq 1, n$	$ Suc_1 $	$ Pred_n $	$ Suc_j , j \neq 1, n$	$ Pred_j , j \neq 1, n$
mín	4	1	1	1	3	3	1	1
máx	4	10	4	10	3	3	3	3

Figura 11: Valores de los parámetros para generar j30, j60, j90 y j120

- $K$  Cantidad de tipos de recursos.
- $d_j$  Duración de la actividad  $j$ .
- $Q$  Número de tipos de recursos que precisa una actividad.
- $r_{j,k}$  Cantidad de unidades del recurso tipo  $k$  que necesita la actividad  $j$ .
- $|Suc_j|$ , Conjunto de las sucesoras inmediatas de  $j$ .
- $|Pred_j|$ , Conjunto de las predecesoras o antecesoras inmediatas de  $j$ .

## 7.2. Diseño y Estrategia de la Experimentación

- Definición del problema.

Se toma una instancia cualquiera del conjunto de problemas RCPSp con 30 actividades de la PSPLIB, la designada como J301\_1.RCP. Puesto que lo habitual en la literatura existente es trabajar con datos difusos en los tiempos y duraciones de las actividades, en esta ocasión nos centramos en el tratamiento difuso de los datos de la instancia relacionados con las cargas y cantidades admisibles de recursos. Partiendo de la misma instancia se plantean, desarrollan y



solucionan cuatro supuestos, comentándose posteriormente en el [apartado 7.3](#) los resultados obtenidos:

1. Solución al problema de forma clásica, suponiendo solamente las relaciones de precedencia entre las distintas actividades.
2. Solución nuevamente con datos nítidos contemplando además el cumplimiento de las disponibilidades máximas de los recursos.
3. Solución teniendo en cuenta las restricciones de las actividades y los recursos, igual que en supuesto 2º, pero considerando como difusos los datos del consumo de recursos de cada una de las actividades por periodo de tiempo.
4. Solución con los mismos condicionantes que el caso 3º, valorando también que la disponibilidad máxima por cada tipo de recurso y periodo de tiempo es difusa.

Para resolver los distintos casos se trabaja con una metodología heurística basada en reglas de prioridad, partiendo de los datos de la PSPLIB aplicando en primer lugar la lógica clásica y posteriormente los conceptos de la lógica fuzzy o difusa. Se utilizará un único esquema de secuenciación en serie pasando el algoritmo hacia delante en una sola ocasión y aplicando en el forward pass una regla de prioridad muy habitual basada en la red, la SPT (Sortest Processing Time rule) o regla del tiempo de ejecución más pequeño ( $\min d_j$ ), en caso de empate se tomará la decisión de avance empleando la regla del mínimo valor de la etiqueta que identifica a cada actividad.

Se cumplen las condiciones características del RCPSP explicadas en el [apartado 3.6](#). Se dispone de 32 actividades, 30 del problema más *la actividad ficticia inicial 1 y la actividad ficticia final 32*; se cuenta con 4 tipos de recursos  $k_1, k_2, k_3$  y  $k_4$ . La disponibilidad máxima de recursos durante todo el proyecto por periodo de tiempo, es decir por día, es para cada tipo de recurso la siguiente:

$$R_{k_1}^{\max} = 12, R_{k_2}^{\max} = 13, R_{k_3}^{\max} = 4 \text{ y } R_{k_4}^{\max} = 12 \quad [7.1]$$

Además en el J301\_1.RCP, cada una de las actividades, exceptuando las ficticias inicial y final que no consumen ni tiempo ni recursos, se sirven durante su procesamiento de un único tipo de recurso. El resto de datos que ofrece la PSPLIB se encuentran en la [Figura 12](#)<sup>31</sup>. Debe tenerse en cuenta que no es objeto de este proyecto de Máster tratar de optimizar las soluciones, dejando este asunto para una investigación posterior.

<sup>31</sup> Obsérvese que se cumplen los valores máximos y mínimos indicados en la Figura 11.



Actividad	Dur	Sucesoras	Recursos	
			Tipo (k)	ud. por día
1	0	2, 3 y 4	0	0
2	8	6, 11 y 15	1	4
3	4	7, 8 y 13	1	10
4	6	5, 9 y 10	4	3
5	3	20	1	3
6	8	30	4	8
7	5	27	1	4
8	9	12, 19 y 27	2	1
9	2	14	1	6
10	7	16 y 25	4	1
11	9	20 y 26	2	5
12	2	14	2	7
13	6	17 y 18	1	4
14	3	17	2	8
15	9	25	1	3
16	10	21 y 22	4	5
17	6	22	4	8
18	5	20 y 22	4	7
19	3	24 y 29	2	1
20	7	23 y 25	2	10
21	2	28	4	6
22	7	23	1	2
23	2	24	1	3
24	3	30	2	9
25	3	30	1	4
26	7	31	3	4
27	8	28	4	7
28	3	31	2	8
29	7	32	2	7
30	2	32	2	7
31	2	32	3	2
32	0	-	0	0

Figura 12: Datos PSPLIB para el caso J301\_1.RCP

- Solución del problema J301-1.RCP considerando solo las relaciones de precedencia.

Si únicamente se consideran las actividades y sus relaciones de precedencia y no se tienen en cuenta las restricciones de los recursos, se obtiene una planificación con un makespan de 38 días. A continuación se muestra dicha planificación incorporada en un diagrama de Gantt (ver [Figura 13](#), [Figura 14](#), [Figura 15](#) y [Figura 16](#)). Los números del 1 al 38 son los días del calendario del proyecto; en la columna “Job” (actividad en inglés) aparecen las distintas actividades designadas con un número o etiqueta; la columna “Dur” indica la duración en días de cada una de las tareas; “ES” (Early Start) es el tiempo más pronto de empezar de la actividad; “EF” se corresponde con el tiempo más tarde de terminar la actividad; la columna “Res” (Resources) hace referencia a  $r_{j,k}$  o cantidad de recursos necesarios para poder ejecutar la actividad. Las celdas de color verde muestran los días del calendario en que cada actividad debe desarrollarse. Cada uno de los diagramas representa a uno de los 4 tipos de recursos existentes en el proyecto, mostrando el orden en que cada actividad debe procesarse para cumplir las relaciones de precedencia e incluyéndose dentro de cada celda la cantidad de recursos que necesita dicha actividad en cada periodo de tiempo o día. En la última fila se realiza la suma de los recursos por días, destacándose en color amarillo aquellos en los que se sobrepasa la cantidad de recursos disponibles. Así mismo la última fila se inicia con una celda en la que aparece la disponibilidad máxima por día del recurso tipo o  $R_k$  (recogida en [\[7.1\]](#)) y la máxima cantidad diaria de recurso encontrada en los 38 días del calendario.



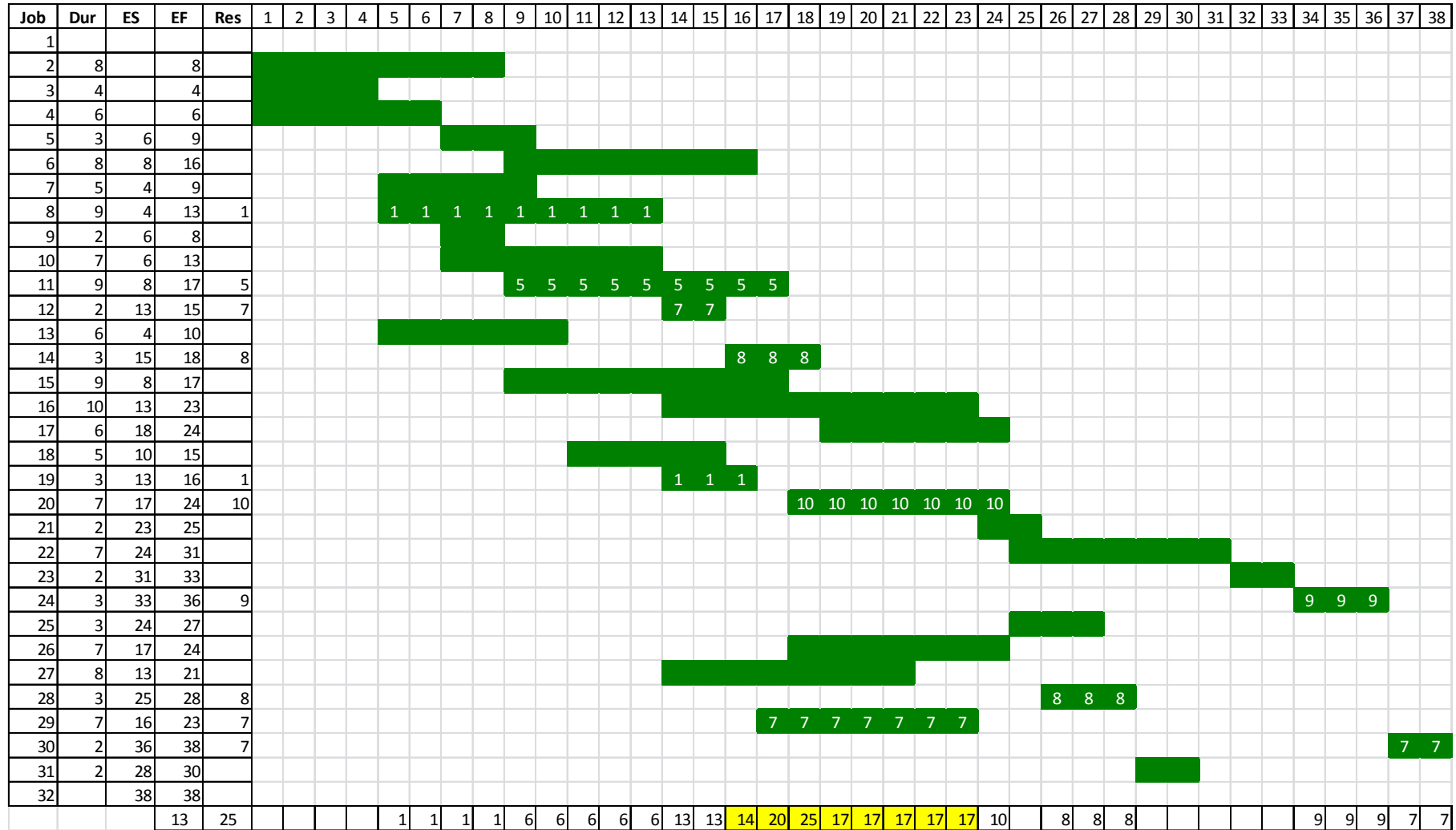


Figura 14: Gantt inicial del proyecto J301\_1.RCP, respetando sólo precedencias e indicando los recursos tipo  $k_2$ .









● Solución con datos nítidos según la lógica clásica respetando precedencias y limitación de recursos.

Se secuencian nuevamente las actividades teniendo en cuenta las restricciones de los recursos y las reglas de prioridad consideradas en la definición del problema. El makespan del proyecto aumenta de 38 a 57 días. En la **Figura 18** se han reunido en un solo grafo los cuatro tipos de recursos, representando en el calendario las distintas actividades secuenciadas que realizan cada uno de ellos y donde se aprecia la cantidad de recursos necesarios para su ejecución. Las cuatro últimas filas del cuadro recogen la carga diaria de cada recurso procedente del sumatorio de las distintas actividades que se deben procesar. Al comparar con la cantidad admisible de cada uno de los tipos se comprueba que en ningún momento se superan las disponibilidades máximas tolerables. En la **Figura 17** se recoge paso a paso cada una de las 31 etapas que han de realizarse para encontrar esta 2ª solución al problema.

<i>g</i>	1	2	3	4	5	6	7	8				
<i>Sec<sub>g</sub></i>	1	1,3	1,3,7	1,3,4,7	1,3,4,7,9	1,3,4,5,7,9	1,3,4,5,7,9,13	1,3,4,5,7,9,13,18				
<i>F<sub>g</sub></i>	0	0,4	0,4,9	0,4,6,9	0,4,6,8,9	0,4,6,8,9,11	0,4,6,8,9,11,14	0,4,6,8,9,11,14,19				
<i>D<sub>g</sub></i>	2,3,4	2,4,7,8,13	2,4,8,13	2,5,8,9,10,13	2,5,8,10,13	2,8,10,13	2,8,10,18,17	2,8,10				
<i>j</i>	3	7	4	9	5	13	18	10				
<i>g</i>	9	10	11	12	13	14	15	16				
<i>Sec<sub>g</sub></i>	1,3,4,5,7,9,10,13,18	1,2,3,4,5,7,9,10,13,18	1,2,3,4,5,6,7,9,10,13,18	1,2,3,4,5,6,7,8,9,10,13,18	1,2,3,4,5,6,7,8,9,10,12,13,18	1,2,3,4,5,6,7,8,9,10,12,13,14,18	1,2,3,4,5,6,7,8,9,10,12,13,14,18,19	1,2,3,4,5,6,7,8,9,10,12,13,14,17,18,19				
<i>F<sub>g</sub></i>	0,4,6,8,9,11,13,14,19	0,4,6,8,9,11,13,14,17,19	0,4,6,8,9,11,13,14,17,19,27	0,4,6,8,9,11,13,14,17,19,27	0,4,6,8,9,11,13,14,15,17,19,27	0,4,6,8,9,11,13,14,15,17,18,19,27	0,4,6,8,9,11,13,14,15,16,17,18,19,27	0,4,6,8,9,11,13,14,15,16,17,18,19,27,33				
<i>D<sub>g</sub></i>	2,8,16	6,8,11,15,16	8,11,15,16	11,12,15,16,19,27	11,14,15,16,19,27	11,15,16,17,19,27	11,15,16,17,27,29	11,15,16,27,29				
<i>j</i>	2	6	8	12	14	19	17	29				
<i>g</i>	17	18	19	20	21	22	23	24				
<i>Sec<sub>g</sub></i>	1,2,3,4,5,6,7,8,9,10,12,13,14,17,18,19,29	1,2,3,4,5,6,7,8,9,10,12,13,14,17,18,19,27,29	1,2,3,4,5,6,7,8,9,10,11,12,13,14,17,18,19,27,29	1,2,3,4,5,6,7,8,9,10,11,12,13,14,17,18,19,20,27,29	1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,17,18,19,20,27,29	1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,17,18,19,20,25,27,29	1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,17,18,19,20,25,26,27,29	1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,25,26,27,29				
<i>F<sub>g</sub></i>	0,4,6,8,9,11,13,14,15,16,17,18,19,25,27,33	0,4,6,8,9,11,13,14,15,16,17,18,19,25,27,33,41	0,4,6,8,9,11,13,14,15,16,17,18,19,25,26,27,33,41	0,4,6,8,9,11,13,14,15,16,17,18,19,25,26,27,33,41	0,4,6,8,9,11,13,14,15,16,17,18,19,25,26,27,33,41	0,4,6,8,9,11,13,14,15,16,17,18,19,25,26,27,33,36,41	0,4,6,8,9,11,13,14,15,16,17,18,19,25,26,27,33,36,41	0,4,6,8,9,11,13,14,15,16,17,18,19,25,26,27,33,36,41,43				
<i>D<sub>g</sub></i>	11,15,16,27	11,15,16	15,16,20,26	15,16,26	16,25,26	16,26	16	21,22				
<i>j</i>	27	11	20	15	25	26	16	21				
<i>g</i>	25	26	27	28	29	30	31					
<i>Sec<sub>g</sub></i>	1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,25,26,27,29	1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,25,26,27,28,29	1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,25,26,27,28,29,31	1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,25,26,27,28,29,31	1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,25,26,27,28,29,31	1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,31	1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31					
<i>F<sub>g</sub></i>	0,4,6,8,9,11,13,14,15,16,17,18,19,25,26,27,33,36,41,43,45	0,4,6,8,9,11,13,14,15,16,17,18,19,25,26,27,33,36,41,43,45,48	0,4,6,8,9,11,13,14,15,16,17,18,19,25,26,27,33,36,41,43,45,48,50	0,4,6,8,9,11,13,14,15,16,17,18,19,25,26,27,33,36,41,43,45,48,50	0,4,6,8,9,11,13,14,15,16,17,18,19,25,26,27,33,36,41,43,45,48,50,52	0,4,6,8,9,11,13,14,15,16,17,18,19,25,26,27,33,36,41,43,45,48,50,52,55	0,4,6,8,9,11,13,14,15,16,17,18,19,25,26,27,33,36,41,43,45,48,50,52,55,57					
<i>D<sub>g</sub></i>	22,28	22,31	22	23	24	30	32					
<i>j</i>	28	31	22	23	24	30	32					
<i>g</i>	: Secuencia o etapa		<i>Sec<sub>g</sub></i>			: Conjunto actividades ya secuenciadas			<i>F<sub>g</sub></i>		: Conjunto tiempos de completación	
<i>D<sub>g</sub></i>	: Conjunto actividades elegibles		<i>j</i>			: Actividad elegida						

Figura 17: Secuenciación en Serie\_Datos Nítidos.





● Solución con cargas de recursos por día difusas y disponibilidad máxima nítida.

Las actividades y sus precedencias como es lógico, siguen siendo las mismas que en las hipótesis anteriores, considerándose las duraciones de las actividades como números nítidos. Igualmente se mantienen el tipo de recurso que utiliza cada actividad pero ahora las unidades de recurso por día se representan mediante un número difuso triangular ( $\widetilde{r}_{j,k}$ ) en donde el núcleo son las unidades iniciales, y los soportes inferior y superior se obtienen aplicando a ese núcleo un coeficiente de minoración del 40% y de mayoración del 20%. La disponibilidad máxima de cada tipo de recurso ( $R_{k_n}^{\max}$ ) continúa siendo nítida tal y como aparece en [7.1].

Actividad	Dur	Sucesoras	Recursos			
			Tipo (k)	ud. por día		
				sop.inf	núcleo	sop.sup
1	0	2, 3 y 4	0	0	0	
2	8	6, 11 y 15	1	2,4	4,8	
3	4	7, 8 y 13	1	6	12	
4	6	5, 9 y 10	4	1,8	3,6	
5	3	20	1	1,8	3,6	
6	8	30	4	4,8	9,6	
7	5	27	1	2,4	4,8	
8	9	12,19 y 27	2	0,6	1,2	
9	2	14	1	3,6	7,2	
10	7	16 y 25	4	0,6	1,2	
11	9	20 y 26	2	3	6	
12	2	14	2	4,2	8,4	
13	6	17 y 18	1	2,4	4,8	
14	3	17	2	4,8	9,6	
15	9	25	1	1,8	3,6	
16	10	21 y 22	4	3	6	
17	6	22	4	4,8	9,6	
18	5	20 y 22	4	4,2	8,4	
19	3	24 y 29	2	0,6	1,2	
20	7	23 y 25	2	6	12	
21	2	28	4	3,6	7,2	
22	7	23	1	1,2	2,4	
23	2	24	1	1,8	3,6	
24	3	30	2	5,4	10,8	
25	3	30	1	2,4	4,8	
26	7	31	3	2,4	4,8	
27	8	28	4	4,2	8,4	
28	3	31	2	4,8	9,6	
29	7	32	2	4,2	8,4	
30	2	32	2	4,2	8,4	
31	2	32	3	1,2	2,4	
32	0	-	0	0	0	

Figura 19: Datos del proyecto J301\_1.RCP con la cantidad de recursos por actividad representada mediante un número difuso.

Como se ha decidido que la disponibilidad máxima por cada tipo de recurso es un número nítido, para poder confrontarlas con las cargas diarias se sigue el procedimiento indicado en el apartado 6.2.2, por tanto las disponibilidades máximas quedarían según se indica a continuación:

$$\widetilde{R}_{k_1}^{\max} = (12,12,12), \widetilde{R}_{k_2}^{\max} = (13,13,13), \widetilde{R}_{k_3}^{\max} = (4,4,4) \text{ y } \widetilde{R}_{k_4}^{\max} = (12,12,12) \quad [7.2]$$



Se deben comparar las disponibilidades necesarias por día con las cargas máximas disponibles de acuerdo a [7.2]. En base a la Figura 18, las distintas cuantías por periodo de tiempo o día de cada tipo de recurso quedan reflejadas en el siguiente cuadro, considerándose como núcleo de los números difusos la cantidad correspondiente a la suma de recursos por días y calculando el soporte inferior y el superior (ver [6.7]) teniendo en cuenta los coeficientes del 40% y del 20%:

		Tipo de Recurso											
		$k_1$			$k_2$			$k_3$			$k_4$		
		i	n	s	i	n	s	i	n	s	i	n	s
Cantidades de Recursos	Totales por día	6,6	11	13,2	7,8	13	15,6	2,4	4	4,8	7,2	12	14,4
		6	10	12	7,2	12	14,4	1,2	2	2,4	4,8	8	9,6
		4,8	8	9,6	6	10	12				4,2	7	8,4
		2,4	4	4,8	5,4	9	10,8				3,6	6	7,2
		1,8	3	3,6	4,8	8	9,6				3	5	6
		1,2	2	2,4	4,2	7	8,4				1,8	3	3,6
					3	5	6				0,6	1	1,2
					0,6	1	1,2						

i = soporte inferior      n = núcleo      s = soporte superior

Figura 20: Cantidades de recursos totales por día en función del tipo de recurso.

Según se estudió en [6.11], comparando  $\tilde{C}$ , que representa a  $\widetilde{R}_{k_1}^{\max} = 12$ , con la mayor cantidad necesaria por ejemplo de  $k_1$ , es decir  $\tilde{D} = (6,6, 11, 13,2)$ , se afirma que  $\tilde{D} < \widetilde{R}_{k_1}^{\max}$ . Al representar gráficamente ambos números, por semejanza de triángulos se obtiene un alfa-corte  $\alpha = 0.54$  donde estos interseccionan, observando que entre los alfa-cortes de 0 a 0.54 existe una zona sombreada que pertenece al intervalo  $[12, 13,2]$  de cantidad de recursos en donde se sobrepasa la carga máxima del recurso  $k_1$ , incumpliendo por tanto la limitación del recurso  $k_1$  al no permitir la nitidez de  $\widetilde{R}_{k_1}^{\max}$  ningún grado de imprecisión. Lo mismo ocurriría para el resto de cantidades marcadas en gris en la Figura 20.

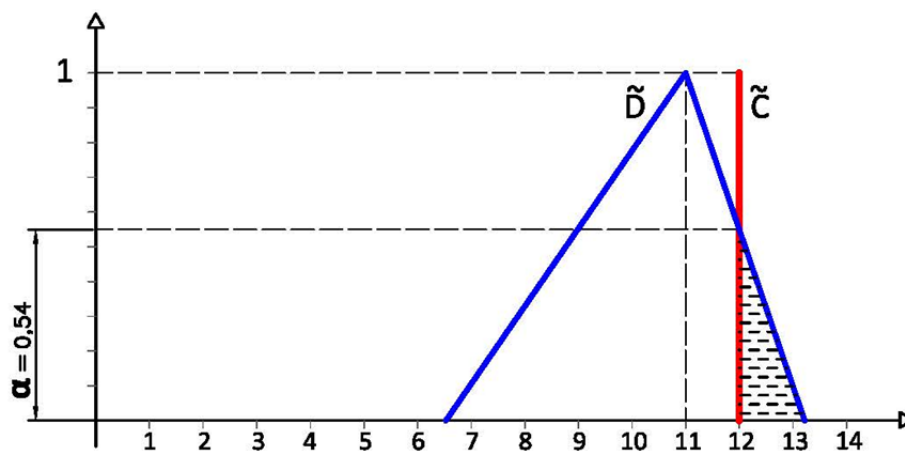


Figura 21: Representación gráfica Número Nítido  $\tilde{C}$  y Número Difuso  $\tilde{D}$ .

Se va a continuar trabajando en este 3º supuesto secuenciándose nuevamente el proyecto pero decidiendo que la suma de las cargas diarias no superen el límite rígido de la



disponibilidad máxima siendo para cada uno los tipos  $k_1, k_2, k_3$  y  $k_4$  respectivamente (6,10,12), (6,10,12), (1'2,3,3'6) y (6,10,12). Obviamente se está cambiando el caso porque lo que realmente se hace es marcar como punto de referencia unos límites inferiores a los que se tenían inicialmente, pero vamos a ver qué sucede para mostrar cual sería el procedimiento a seguir manteniendo la consideración de unas disponibilidades máximas nítidas. Por una parte disponemos de las etapas que se van realizando y por otra del Gantt donde estas quedan reflejadas.

$g$	1	2	3	4	5	6	7	8
$Sec_g$	1	1,3	1,3,7	1,3,4,7	1,3,4,7,9	1,3,4,5,7,9	1,3,4,5,7,9,13	1,3,4,5,7,9,13,18
$F_g$	0	0,4	0,4,9	0,4,6,9	0,4,6,8,9	0,4,6,8,9,11	0,4,6,8,9,11,15	0,4,6,8,9,11,15,20
$D_g$	2,3,4	2,4,7,8,13	2,4,8,13	2,5,8,9,10,13	2,5,8,10,13	2,8,10,13	2,8,10,18,17	2,8,10
$j$	3	7	4	9	5	13	18	10
$g$	9	10	11	12	13	14	15	16
$Sec_g$	1,3,4,5,7,9,10,13,18	1,2,3,4,5,7,9,10,13,18	1,2,3,4,5,6,7,9,10,13,18	1,2,3,4,5,6,7,8,9,10,13,18	1,2,3,4,5,6,7,8,9,10,12,13,18	1,2,3,4,5,6,7,8,9,10,12,13,14,18	1,2,3,4,5,6,7,8,9,10,12,13,14,18,19	1,2,3,4,5,6,7,8,9,10,12,13,14,17,18,19
$F_g$	0,4,6,8,9,11,13,15,20	0,4,6,8,9,11,13,15,19,20	0,4,6,8,9,11,13,15,19,20,28	0,4,6,8,9,11,13,15,19,20,28	0,4,6,8,9,11,13,15,19,20,28	0,4,6,8,9,11,13,15,18,19,20,28	0,4,6,8,9,11,13,15,16,18,19,20,28	0,4,6,8,9,11,13,15,16,18,19,20,28,34
$D_g$	2,8,16	6,8,11,15,16	8,11,15,16	11,12,15,16,19,27	11,14,15,16,19,27	11,15,16,17,19,27	11,15,16,17,27,29	11,15,16,27,29
$j$	2	6	8	12	14	19	17	29
$g$	17	18	19	20	21	22	23	24
$Sec_g$	1,2,3,4,5,6,7,8,9,10,12,13,14,17,18,19,29	1,2,3,4,5,6,7,8,9,10,12,13,14,17,18,19,27,29	1,2,3,4,5,6,7,8,9,10,11,12,13,14,17,18,19,27,29	1,2,3,4,5,6,7,8,9,10,11,12,13,14,17,18,19,20,27,29	1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,17,18,19,20,27,29	1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,17,18,19,20,25,27,29		
$F_g$	0,4,6,8,9,11,13,15,16,18,19,20,25,28,34	0,4,6,8,9,11,13,15,16,18,19,20,25,28,34,42	0,4,6,8,9,11,13,15,16,18,19,20,25,28,34,42	0,4,6,8,9,11,13,15,16,18,19,20,25,28,34,41,42	0,4,6,8,9,11,13,15,16,18,19,20,25,28,34,41,42	0,4,6,8,9,11,13,15,16,18,19,20,25,28,34,41,44		
$D_g$	11,15,16,27	11,15,16	15,16,20,26	15,16,26	16,25,26	16,26		
$j$	27	11	20	15	25	¿?		
$g$	: Secuencia o etapa		$Sec_g$			: Conjunto tiempos de completación		
$D_g$	: Conjunto actividades elegibles		$j$			: Actividad elegida		

Figura 22: Secuenciación en Serie\_Carga de recursos difusa sin superar las disponibilidades máximas nítidas.





Al introducir la Actividad 26, la cual necesita 4 recursos para su ejecución volvemos a tropezar con otro problema, para poder seguir se debería reducir el número de recursos de la Actividad 26 a 3 ud/día como máximo y en consecuencia se alargaría la duración de dicha actividad; otra posibilidad sería el aumentar la carga máxima admisible del recurso  $k_4$  como mínimo a 4,8 ud/día ( $4 + 20\%$  de 4). De todas formas se estarían cambiando los datos de la instancia y por tanto el problema no es factible como se comentaba al inicio de este apartado.

- Solución con cantidades de recursos por día difusas y disponibilidades difusas.

Las cantidades de recursos no varían, son las mismas de la Figura 20 y las disponibilidades máximas por tipo de recurso calculando los soportes inferiores y superiores aparecen en la Figura 24, por consiguiente, tanto las cargas diarias como las limitaciones de los recursos se tratan como números difusos. Comparando los baricentros de los números difusos triangulares que representan las cantidades más altas diarias de cada tipo de recurso con su disponibilidad se advierte que no se crean conflictos y que por tanto la solución en este 4º caso es de 57 días, es decir, coincide con la del 2º caso en donde cargas y disponibilidades se analizaban como números nítidos.

Tipo de recurso		$k_1$			$k_2$			$k_3$			$k_4$		
nº difuso más alto	(i,n,s)	6,6	11	13,2	7,8	13	15,6	2,4	4	4,8	7,2	12	14,4
	baricentro	10,45			12,35			3,8			11,4		
$\widetilde{R}^{\max}$	(i,n,s)	7,2	12	14,4	7,8	13	15,6	2,4	4	4,8	7,2	12	14,4
	baricentro	11,4			12,35			3,8			11,4		
Ordenación		nº difuso $< \widetilde{R}_{k_1}^{\max}$			nº difuso $= \widetilde{R}_{k_2}^{\max}$			nº difuso $= \widetilde{R}_{k_3}^{\max}$			nº difuso $= \widetilde{R}_{k_4}^{\max}$		
i = soporte inferior		n = núcleo			s = soporte superior								

Figura 24: Solución caso 4º.

7.3. Discusión de Resultados

- Según se van añadiendo restricciones a los problemas la duración final de estos o makespan se ve aumentada.
- La disposición de los recursos es una parte fundamental a la hora de planificar un proyecto.
- La disposición de los recursos es una parte fundamental a la hora de planificar un proyecto.





- Tratar la imprecisión de las cargas de los recursos mediante la lógica fuzzy combinándola con unas limitaciones o restricciones de recursos nítidas genera conflictos en la planificación que afectan a las decisiones a tomar para poder solucionar el problema.
- No se encuentra inconveniente en considerar las limitaciones y cantidades de recursos como números difusos, consiguiéndose los mismos resultados que con los números nítidos, lo cual es importante puesto que la lógica difusa nos permite acercarnos más a la realidad que la lógica clásica.





## 8. Conclusiones y Futuras Líneas de investigación

Cada proyecto de construcción es único, las actividades que lo forman deben ser gestionadas desde su origen cuando son planificadas hasta su finalización. Esta gestión requiere de la experiencia de los jefes de proyecto para conseguir con éxito los objetivos que se persiguen, pero su dificultad va aumentando en función de la cantidad de trabajos a realizar, las combinaciones y secuenciaciones se pueden llegar a complicar tanto para encontrar una solución adecuada que las herramientas tradicionales no son suficientes, debiéndose recurrir además de a los conocimientos en construcción, a técnicas matemáticas e informáticas, de esta manera los problemas son tratados como modelos matemáticos intentando solucionarlos mediante programas informáticos.

Los recursos necesarios para la ejecución de las actividades son una parte importantísima a considerar dentro de la gestión del proyecto, las limitaciones de estos dan lugar a una diversidad de problemas que pueden reunirse bajo el nombre de Resource Constrained Project Scheduling Problem o RCPSp. En este trabajo se ha aplicado la lógica difusa sobre la limitación de recursos, pues si bien se conocen estudios de este tipo de problemas que experimentan con la imprecisión aplicada a las duraciones de las tareas o con la incertidumbre sobre recursos, no se tiene conocimiento de experiencias sobre la imprecisión en los recursos.

Tras profundizar en los supuestos aplicados a un problema estándar RCPSp, se llega a la conclusión de que la combinación de la lógica clásica y la difusa puede llevarnos a soluciones que aun no siendo las óptimas pueden servir para enriquecer los resultados al unir la rigidez de la primera con la realidad que ofrece la lógica fuzzy, ya que el mundo real es precisamente "impreciso". La aplicación de la lógica difusa en Sistemas Expertos<sup>32</sup> y Automatismos está a la orden del día, pero en temas usuales y no por ello sencillos como por ejemplo la ejecución de un edificio, queda todavía mucho por investigar.

Esta memoria deja bastantes puertas abiertas a trabajos posteriores sobre los que se puede desarrollar una tesis doctoral, como por ejemplo:

- Estudio detallado de las distintas variantes y extensiones del problema RCPSp y aplicación de estos al sector de la construcción.
- Profundizar en la notación de estos problemas para mejorar las clasificaciones que se manejan actualmente para facilitar el entendimiento entre los investigadores.
- Clasificación detallada e íntegra de los algoritmos existentes en la literatura para resolver los problemas RCPSp relacionados con el sector de la construcción.
- Elaboración y análisis de distintos algoritmos para optimizar las soluciones obtenidas tras la aplicación de la lógica fuzzy a la limitación de recursos.
- Aplicación de la lógica difusa en la nivelación de recursos.
- Realización de estudios comparativos sobre el tratamiento de los recursos considerando las diferencias entre la optimización determinista y la optimización

<sup>32</sup> Sistemas Expertos: Programas para la toma de decisiones en campos específicos.



estocástica, es decir, entre la teoría de las posibilidades y la teoría de las probabilidades.

- Investigación de la limitación de recursos mediante el Método del Principio de Extensión de Zadeh, justificando la dificultad que presenta la programación al aplicar este método.
- Implementación de las investigaciones a un caso real de un proyecto de construcción, con la utilización de la imprecisión en las restricciones sobre las duraciones y los recursos. Validación del modelo empleado.



**9. Índice de Figuras**

Figura 1: Funcionamiento del razonamiento humano..... 7

Figura 2: Cronología de Gestión y Limitación de Recursos. .... 15

Figura 3: Objetivos básicos en la gestión de proyectos. .... 19

Figura 4: Jerarquía científico-técnica para resolución de problemas. .... 31

Figura 5: Tipos algoritmos para resolución problema RCPSP. .... 43

Figura 6: Esquema Lógica. Fuente (Trillas, et al., 1994)..... 56

Figura 7: Número Difuso Triangular  $\tilde{A}$  y Alfa-corte  $\tilde{A}[\alpha]$  del número difuso. .... 59

Figura 8: Número Difuso Trapezoidal  $\tilde{M}$  ..... 60

Figura 9: Suma de dos números difusos ..... 61

Figura 10: Representación gráfica de la suma de números difusos ..... 61

Figura 11: Valores de los parámetros para generar j30, j60, j90 y j120 ..... 67

Figura 12: Datos PSPLIB para el caso J301\_1.RCP ..... 69

Figura 13: Gantt inicial del proyecto J301\_1.RCP, respetando sólo precedencias e indicando los recursos tipo  $k_1$  ..... 70

Figura 14: Gantt inicial del proyecto J301\_1.RCP, respetando sólo precedencias e indicando los recursos tipo  $k_2$  ..... 71

Figura 15 : Gantt inicial del proyecto J301\_1.RCP, respetando sólo precedencias e indicando los recursos tipo  $k_3$  ..... 72

Figura 16: Gantt inicial del proyecto J301\_1.RCP, respetando sólo precedencias e indicando los recursos tipo  $k_4$  ..... 73

Figura 17: Secuenciación en Serie\_Datos Nítidos. .... 74

Figura 18: Solución al J301\_1.RCP con datos nítidos, respetando precedencias y limitaciones de recursos..... 75

Figura 19: Datos del proyecto J301\_1.RCP con la cantidad de recursos por actividad representada mediante un número difuso..... 76

Figura 20: Cantidades de recursos totales por día en función del tipo de recurso. .... 77

Figura 21: Representación gráfica Número Nítido  $\tilde{C}$  y Número Difuso  $\tilde{D}$ . .... 77

Figura 22: Secuenciación en Serie\_Carga de recursos difusa sin superar las disponibilidades máximas nítidas. .... 78

Figura 23: Grafo considerando las cargas de recursos difusas y la disponibilidad máxima nítida. .... 79

Figura 24: Solución caso 4º. .... 80





## 10. Bibliografía

Agirre Pérez, I., 2007. *Sistema de planificación estocástico de proyectos: Implicaciones en la gestión de riesgos*. Logroño: Universidad de la Rioja, Servicio de Publicaciones.

Al-Anzi, F. S., Al-Zamel, K. & Allahverdi, A., 2010. Weighted Multi-Skill Resources Project. *J. Software Engineering & Applications*, Volumen 3, pp. 1125-1130.

Alcaraz Soria, J. & Maroto Álvarez, C., 2009. Genetic Algorithms for the Resource-Constrained Project Scheduling Problem. *Boletín de Estadística e Investigación Operativa*, 25(1), pp. 22-31.

Álvarez-Valdés Olaguíbel, R., Crespo, E., Tamarit Goerlich, J. M. & Fulgencia, V. J. M., 2009. Un procedimiento de fuerte reducción de las dimensiones del RCPS/. *Revista de Matemática: Teoría y Aplicaciones*, 16(1), pp. 188-198.

Álvarez-Valdés, R. & Tamarit, J., 1989. Heuristic Algorithms for Resource-Constrained Project Scheduling: A Review and an Empirical Analysis.. *Slowinski, R. and J. Weglarz (Eds.): Advances In Project Scheduling*. Elsevier, pp. 113-134.

Areces, P., Alba, C., Villanueva, J. & Alberto, G., 2011. *XV Congreso Intenacional de Ingeniería de Proyectos*. Huesca, Universidad de Oviedo.

Ballestín González, F., 2002. *Nuevos Métodos de Resolución del Problema de Secuenciación de Proyectos con Recursos Limitados*. Valencia: Universitat de València, Servicio de Publicaciones.

Bhaskar, T., Manabendra, N. P. & Asim, K. P., 2011. A heuristic method for RCPSP with fuzzy activity times. *European Journal of Operational Research*, Volumen 208, pp. 57-66.

Blazewicz, J., J.K., L. & Rinnooy Kan, A., 1983. Scheduling subjeçyt to Resource Constraints: classification and complexity. *Discrete Applied Mathematics*, Volumen 5, pp. 11-24.

Brucker, P., 1995. *Schedulings Algorithms*. 5ª ed. Berlin: Springer-Verlag.

Brucker, P. y otros, 1999. Resource-constrained project scheduling: Notation, classification, models and methods. *European Journal of Operational Research*, Volumen 112, pp. 3-41.

Brucker, P. & Knust, S., 2012. *Complex Scheduling*. 2ª ed. Berlin: Springer.

Buckley, J. J., Eslami, E. & Feuring, T., 2002. *Fuzzy Mathemtics in Economics and Engineering*. Heidelberg: Physica-Verlag.

Carrascosa Martínez, J., Durán Álvarez, J., Martínez Aires, M. & Martínez Rueda, M., 2007. *Organización, Programación y Control de Obra*. 2ª ed. Granada: Proyecto Sur Industrias Gráficas.

Cervantes Posada, M., 2010. *Nuevos Métodos Metaheurísticos para la Asignación Eficiente, Optimizada y Robusta de Recursos Limitados*. Valencia: Universidad Politécnica de Valencia, Servicio de Publicaciones.



Chanas, S. & Kamburowski, 1981. The use of Fuzzy variables in PERT. *Fuzzy Sets Systems*, 5(1), pp. 11-19.

Chen, C. & Huang, S., 2007. Applying fuzzy method for measuring criticality in project network. *Information Sciences*, Volumen 177, pp. 2248-2458.

Chevalier, J., 1960. *Organización de empresas*. 2ª ed. Barcelona: Palestra.

Cowling, T., Kendall, G. & Soubeiga, E., 2001. A Hyperheuristic Approach to Scheduling a Sales Summit. *Lecture Notes in Computer Science*, Volumen 2079, pp. 176-190.

Crespo Escobar, E. & Villa Juliá, M. F., 2005. Secuenciación heurística de un proyecto con restricciones temporales. *Revista Electrónica de Comunicaciones y Trabajos de ASEPUMA*, Volumen 6, pp. 5-28.

Crespo Robles, E., 2008. *Optimización estocástica del mantenimiento de sistemas de transporte*. Madrid: Universidad Politécnica de Madrid.

de la Peña Esteban, D., García de Frutos, D. & Simón Rodríguez, M. A., 2005. Formulación de un algoritmo para la asignación y nivelación de recursos en la programación de Proyectos con recursos limitados y distintas duraciones posibles para las actividades, utilizando el Método de Roy. *Tecnología y Desarrollo*, Volumen 3, p. Separata.

Demeulemeester, E. & Herroelen, W., 2002. *Project scheduling: a research handbook*. Boston: Kluwer Academic.

Didier, D. & Prade, H., 2000. *"Fundamentals of Fuzzy Sets"*. Massachusetts: Kluwer.

Dubois, D., Fargier, H. & Fortemps, P., 2003. Fuzzy scheduling: modelling flexible constraints vs. coping with incomplete knowledge. *European Journal of Operational Research*, Volumen 147, pp. 231-252.

Fernández-Viagas Escudero, V., 2012. *Estudio de la programación de tareas y asignación de personal en empresas de servicio*. Universidad de Sevilla. [En línea]  
Available at: <http://bibing.us.es/proyectos/abreproy/70317/>  
[Último acceso: mayo 2013].

García Sabater, J. & Maheut, J., 2011. *Modelado y Resolución de Problemas de Organización Industrial mediante Programación Matemática*. [En línea]  
Available at: [personales.upv.es/jpgarcia/LinkedDocuments/modeladomatematico.pdf](http://personales.upv.es/jpgarcia/LinkedDocuments/modeladomatematico.pdf)  
[Último acceso: junio 2013].

González Rodríguez, P. L., s.f. *Origen y Evolución de la Organización del Trabajo*. Sevilla: Universidad de Sevilla, Servicio de Publicaciones.

Graham, R., Lawler, E., J.K., L. & Rinnooy Kan, A., 1979. Optimization and Approximation in Deterministic Sequencing and Scheduling: a Survey. *Annals of Discrete Mathematics*, Volumen 5, pp. 287-326.





Hapke, M. & Slowinski, R., 1996. Fuzzy priority heuristics for project scheduling. *Fuzzy Sets and Systems*, 83(3), pp. 291-299.

Hartmann, S. & Briskorn, D., 2010. A survey of variants and extensions of the resource-constrained project scheduling problem. *European Journal of Operational Research*, Volumen 207, pp. 1-14.

Hekimoglu, Ö., 2007. *Comparison of the resource allocation capabilities of project management software packages in resource constrained project scheduling problems*. Ankara: Middle East Technical University.

Hernández Negrín, I. & I., L. C., 1997. "Lógica Fuzzy para Principiantes: cuando la máquina se acerca al pensamiento humano". Madrid: Omrom Electronics.

Herroelen, W., Demeulemeester, E. & De Reyck, B., 1996. *Resource-Constrained Project Scheduling: A survey of recent developments*, Leuven: Katholieke Universiteit Leuven.

Herroelen, W., Demeulemeester, E. & De Reyck, B., 1998. A classification scheme for project scheduling problems. *Kluwer Academics Publishers*, Volumen 1, pp. 1-26.

Herroelen, W., Demeulemeester, E. & De Reyck, B., 2001. A note on the paper "Resource-Constrained Project Scheduling: Notation, Classification, Models, and Methods" by Brucker et al. *European Journal of Operational Research*, 128(3), pp. 679-688.

Instituto Universitario Investigación, 2012. *Centro de Investigación Operativa. Universidad Miguel Hernández*. [En línea]  
Available at: <https://sites.google.com/site/gdpumh/>

Kelley, J. J., 1963. The critical-path method: Resources planning and scheduling. *Industrial Scheduling*, Muth, J. and G. Thompson (eds.). Prentice-Hall, New Jersey., pp. 347-365.

Kerzner, H., 2009. *Project Management: a Systems Approach to Planning, Scheduling and Control*. 10ª ed. Hoboken (New Jersey): John Wiley & Sons, Inc..

Kim, K., Gen, M. & Yamazaki, G., 2003. Hybrid genetic algorithm with fuzzy logic for resource-constrained project scheduling. *Applied Soft Computing*, 2(3), pp. 174-188.

Klastorin, T., 2010. *Gestión de proyectos*. Barcelona: Profit Editorial I., S.L..

Klein, R., 2000. *Scheduling of Resource Constrained Projects*. Massachusetts: Kluwer Academic.

Kolisch, R., 1995b. *Project scheduling under resource constraints: efficient heuristics for several problem classes*. Michigan: University of Michigan.

Kolisch, R., 1999. Resource Allocation Capabilities of Commercial Project Management Software Packages. *Interface*, Volumen 29, pp. 19-31.

Kolisch, R. & Hartmann, S., 2006. Experimental Investigation of Heuristics for Resource-Constrained Project Scheduling: An Update. *European Journal of Operational Research*.



- Kolisch, R. & Hartman, S., 1999. Heuristic Algorithms for Solving the Resource-Constrained Project Scheduling Problem: Classification and Computational Analysis. *Project scheduling: Recent models, algorithms and applications*, pp. 147-178.
- Kolisch, R. & Padman, R., 2001. An integrated survey of deterministic project scheduling. *Omega*, Volumen 29, pp. 249-272.
- Kolisch, R. & Sprecher, A., 1996. PSPLIB-A project Scheduling problem library. *European Journal of Operational Research*, Volumen 96, pp. 205-216.
- Kolisch, R., Sprecher, A. & Drexel, A., 1995a. Characterization and generation of a general class of resource-constrained project scheduling problems. *Management Science*, 41(10), pp. 1693-1703.
- Lawler, E., Lenstra, J., Rinnooy Kan, A. & Shomoys, D., 1989. *Sequencing and Scheduling: Algorithms and Complexity*. Eindhoven: Eindhoven University of Technology.
- Liao, T., Egbelu, P., Sarker, B. & Leu, S., 2011. Metaheuristics for project and construction management – A state of the art review. *Automation in Construction*, Volumen 20, pp. 491-505.
- Liao, T., Egbelu, P., Sarker, B. & Leu, S., 2011. Metaheuristics for project and construction management – A state-of-the-art review. *Automation in Construction*, Volumen 20, pp. 491-505.
- Liberatore, M., 2008. Critical Path Analysis With Fuzzy Activity Times. *IEEE TRANSACTIONS ON ENGINEERING MANAGEMENT*, 55(2), pp. 329-337.
- Lorterapong, P., 1994. A fuzzy heuristic method for resource-constrained project scheduling. *Project Management Journal*, 25(4), pp. 12-18.
- Lova, A., Tormos, P. & Barber, B., 2006. Multi-Mode Resource Constrained Project Scheduling: Scheduling, Schemes, Priority Rules and Mode Selection Rules. *Inteligencia Artificial*, 10(30), pp. 69-86.
- Mañas, J. A., 1997. *Análisis de Algoritmos: Complejidad*. Departamento de Ingeniería de Sistemas Telemáticos. U.P. de Madrid. [En línea]  
Available at: <http://www.lab.dit.upm.es/~lprg/material/apuntes/o/index.html>
- Martín Martín, Q., 2010. *Curso de Estadística (Investigación Operativa I)*, Universidad de Salamanca. [En línea]  
Available at: <http://ocw.usal.es/enseñanzas-tecnicas/investigacion-operativa-i>  
[Último acceso: mayo 2013].
- Martí, R. & Moreno Vega, J., 2003. MultiStart Methods. *Inteligencia Artificial, Revista Iberoamericana de Inteligencia Artificial*, Volumen 19, pp. 49-60.
- Masmoudi, M. & Haït, A., 2013. Project scheduling under uncertainty using fuzzy modelling. *Engineering Applications of Artificial Intelligence*, Volumen 26, pp. 135-149.



Medina Ramón, J., 2008. *Diseño Óptimo de Redes para la Programación de Obras de Edificación, para una Nivelación y Distribución de Recursos Personales Constante*. Valencia: Universidad Politécnica de Valencia, Servicio de Publicaciones.

Meléndez Julio, I., s.f. *La programación lineal y la asignación de recursos limitados*. [En línea] Available at: <http://www.monografias.com/trabajos96/programacion-lineal-y-asignacion-recursos-limitados/programacion-lineal-y-asignacion-recursos-limitados.shtml#bibliograa> [Último acceso: mayo 2013].

Montoya Molina, P., s.f. *Manual para la Gestión de Proyectos*. Almería: Universidad de Almería, Servicio de Publicaciones.

Moore, R., 1979. *Methods and Applications Of Interval Analysis*.. Philadelphia: SIAM, Studies in Applied Mathematics.

Morillas, A., 2006. *Eumed.net*. [En línea] Available at: <http://www.eumed.net/libros-gratis/2006b/amr/> [Último acceso: junio 2013].

Navascués Fernández, J., s.f. *Técnicas avanzadas para la gestión de proyectos de software*. Sevilla: Universidad de Sevilla, Servicio de Publicaciones.

Neumaier, A., 1990. *Interval Methods for System of Equations*. Cambridge: Cambridge University Press.

Osman, I. & Kelly, J., 1996. *Meta-heuristics: Theory & Application*. Massachusetts: Kluwer Academic Publishers.

Pellicer Armiñana, E. & Yepes Piqueras, V., 2005. *Consideraciones sobre la función de control aplicada a la gestión de proyectos de construcción*. Málaga, s.n.

Peña, V. & Zumelzu, L., 2006. *Estado del Arte del Job Shop Scheduling Problem*, Valparaíso: Universidad Técnica Federico Santa María.

PMI, 2004. *Project Management Body of Knowledge Guide*. Pennsylvania: Project Management Institute.

PMI, 2012. *Project Management Institute*. [En línea] Available at: <http://www.pmi.org/> [Último acceso: 2012].

Ponz Tienda, J. L., 2008. *Project Management con Redes Pert*. Valencia: Universidad Politécnica de Valencia.

Ponz Tienda, J. L., 2010. *"GRCPS Robusto basado en Producción para Proyectos de Edificación y Construcción"*. Valencia: Universidad Politécnica de Valencia, Servicio de Publicaciones.

Ponz Tienda, J. L., 2011. *"Gestión de Proyectos con Excel 2010"*. Madrid: Anaya Multimedia.



Ponz Tienda, J. L., Pellicer Armiñana, E. & Yepes Piqueras, V., 2012. Complete fuzzy scheduling and fuzzy earned value management in construction projects. *Journal of Zhejiang University SCIENCE A*, 13(1), pp. 56-58.

Ponz Tienda, J. L., Yepes, V., Pellicer, E. & Moreno Flores, J., 2013. The Resource Leveling Problem with Multiple Resources using an Adaptive Genetic Algorithm. *Elsevier\_Automation in Construction*, Volumen 29, pp. 161-172.

Prade, H., 1979. Using fuzzy set theory in a scheduling problem: A case study. *Fuzzy Sets and Systems*, 2(2), pp. 153-165.

Pritsker A.A.B, L.J., W. & P.M., W., 1969. Multiproject scheduling with limited resources: a zero-one programming approach. *Management Science*, 16(1), pp. 93-108.

Sipser, M., 2006. *Introduction to the Theory of Computation*. 2ª ed. Boston: Thomson Course Technology.

Soler Fernández, D., 2009. *Apuntes Optimización (Máster de Gestión en la Edificación, UPV)*. Valencia: s.n.

Trillas, E., Alsina, C. & Terricabras, J., 1995. *"Introducción a la Lógica Borrosa"*. Barcelona: Ariel.

Trillas, E. y otros, 1994. *"Fundamentos e Introducción a la Lógica Fuzzy"*. Madrid: Omrom Electronics.

Uher, T. E., 2003. *Programming and Scheduling Techniques*. 1ª ed. Sydney: University of New South Wales.

Vitoriano Villanueva, B., 2009. *Apuntes sobre Modelos Operativos de Gestión*. Madrid: Universidad Complutense de Madrid, Servicio de Publicaciones.

Yepes Piqueras, V., 2002. *Optimización Heurística Económica aplicada a las redes del transporte VRPTW*. Valencia: Universidad Politécnica de Valencia, Servicio de Publicaciones.

Yepes Piqueras, V., 2012. *Bloque 3 de Optimización Heurística: Modelo matemático de optimización*. [En línea]

Available at: <http://victoryepes.blogs.upv.es/category/modelo-matematico/>

Yepes, V., Carbonell, A. & González-Vidasoa, 2010. *Teoría del valor extremo como criterio de parada en la optimización heurística de bóvedas de hormigón estructural\_ "VII Congreso Español sobre Metaheurísticas, Algoritmos Evolutivos y Bioinspirados"*. Valencia: Universidad Politécnica de Valencia.

Zadeh, L., 1979. Fuzzy Sets versus Probability. *Proceedings of the IEEE*, 68(3), p. 421.

Zadeh, L., 1996a. *"Fuzzy Sets, Fuzzy Logic, and Fuzzy Systems: Selected Papers by Lofti A. Zadeh"*. Singapore: World Scientific.

Zadeh, L., 1996b. Nacimiento y evolución de la lógica borrosa, el soft computing y la computación con palabras: Un punto de vista personal. *Psichothema*, 8(2), pp. 421-429.



Zhi-Jie, C. & Chiu-Cheng, C., 2010. An Evolutionary Algorithm with Multi-Local Search for the Resource-Constrained Project Scheduling Problem. *Intelligent Information Management*, Volumen 2, pp. 220-226.