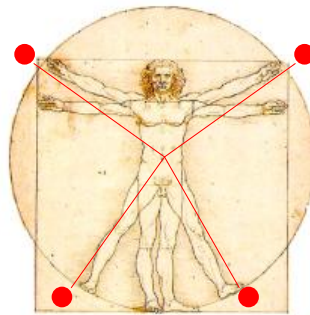


# **TECNOLOGÍ@ y DESARROLLO**

*Revista de Ciencia, Tecnología y Medio Ambiente*

VOLUMEN XII. AÑO 2014

SEPARATA



## **ANÁLISIS DE LOS MÉTODOS HEURÍSTICOS DE RESOLUCIÓN DEL PROBLEMA DE PROGRAMACIÓN DE PROYECTOS CON RECURSOS LIMITADOS (RCPSP: RESOURCE CONSTRAINED PROJECT SCHEDULING PROBLEM)**

**M<sup>a</sup> Antonia Simón Rodríguez, Fco. David de la Peña Esteban,  
Pedro Rincón Rincón**



UNIVERSIDAD ALFONSO X EL SABIO  
Escuela Politécnica Superior  
Villanueva de la Cañada (Madrid)

© Del texto: M<sup>a</sup> Antonia Simón Rodríguez, Fco. David de la Peña Esteban , Pedro Rincón Rincón  
Julio, 2014.

<http://www.uax.es/publicacion/analisis-de-los-metodos-heuristicos-de-resolucion-del-problema-de-programacion.pdf>

© De la edición: *Revista Tecnol@ y desarrollo*

Escuela Politécnica Superior.

Universidad Alfonso X el Sabio.

28691, Villanueva de la Cañada (Madrid).

ISSN: 1696-8085

Editor: Javier Morales Pérez – tecnologia@uax.es

No está permitida la reproducción total o parcial de este artículo, ni su almacenamiento o transmisión ya sea electrónico, químico, mecánico, por fotocopia u otros métodos, sin permiso previo por escrito de la revista.

# **ANÁLISIS DE LOS MÉTODOS HEURÍSTICOS DE RESOLUCIÓN DEL PROBLEMA DE PROGRAMACIÓN DE PROYECTOS CON RECURSOS LIMITADOS (RCPSP: RESOURCE CONSTRAINED PROJECT SCHEDULING PROBLEM)**

**M<sup>a</sup> Antonia Simón Rodríguez<sup>a)</sup>, Fco. David de la Peña Esteban<sup>b)</sup>, Pedro Rincón Rincón<sup>c)</sup>**

a) Doctora Ing. Industrial, Profesora Asociada del Departamento de Tecnología Industrial, Escuela Politécnica Superior, Universidad Alfonso X el Sabio. Av de la universidad nº1, CP 28691, Villanueva de la Cañada, Madrid. España. Tlf.: 9181597583 email:asimorod@uax.es

b) Doctor Ing. Industrial, Profesor Asociado del Departamento de Ingeniería, Universidad a Distancia de Madrid (Udima)

c) Doctor Ing. Industrial, Profesor Asociado del Departamento de Tecnología Industrial, Escuela Politécnica Superior, Universidad Alfonso X el Sabio.

**RESUMEN:** Para que un proyecto finalice en plazo, es de crucial importancia que la programación de los recursos disponibles a las actividades se haga de forma rigurosa, sobre todo cuando los recursos no son infinitos. Este tema es el que aborda la Programación de Proyectos con Recursos Limitados (RCPSP: Resource Constrained Project Scheduling Problem). En este artículo se ha hecho un análisis de las distintas técnicas heurísticas que solucionan este problema, viendo las reglas de prioridad más usadas y sus características, así como de las técnicas metaheurísticas más utilizadas para el problema del RCPSP. Estas técnicas, aunque dan una perspectiva mucho mayor que las tradicionales adolecen también en ciertos aspectos que no se tienen normalmente en consideración.

**PALABRAS CLAVE:** programación de proyectos con recursos limitados, técnicas heurísticas, regla de prioridad, técnicas metaheurísticas, nivelación

***ABSTRACT:** For the the success of the project planning it is essential that the allocation and resource leveling is performed in a accurate way, especially when resources are not infinite. This issue is addressed by the RCPSP (Resource Constrained Project Scheduling Problem). This paper has made an analysis of the different heuristics techniques that solve this problem, the priority rules used and their characteristics, and the metaheuristics techniques commonly used for RCPSP. These techniques, although they give a much higher performance than traditional techniques also suffer from certain aspects that are not usually taken into consideration.*

**KEY-WORDS:** Resource-Constrained Project Scheduling, Heuristics, priority rule, Metaheuristics, leveling.

**SUMARIO:** 1. Introducción 2. Objetivos 3. Programación de Proyectos con recursos limitados 4. Métodos heurísticos para el problema RCPSP 5. Conclusiones 6. Futuras líneas de investigación 7. Bibliografía

## 1. Introducción

Una vez que se ha decidido la realización de un proyecto, y antes de comenzar la ejecución física del mismo, es necesario planificar dicho proyecto en función de las necesidades y recursos disponibles. Para planificar un proyecto es necesario definir cada una de las tareas a realizar y establecer el orden de las mismas, esto es, su secuenciación. Habrá que estimar la duración y la necesidad de recursos de dichas actividades así como la previsión de los recursos disponibles para la realización del proyecto.

Existen dos aproximaciones a la planificación. La *planificación libre*, que busca que el proyecto se realice lo antes posible, en función de las duraciones de las actividades y de las precedencias existentes entre las mismas. Esta planificación dará la fecha de finalización óptima. También está la *planificación objetivo*, en la cual se parte de la fecha en que debe estar finalizado el proyecto, y se programan las actividades “hacia atrás”, de manera que se pueda cumplir esa fecha objetivo. En muchos casos este tipo de planificaciones da lugar a que las tareas deban realizarse en menos tiempo del que sería deseable.

La programación del proyecto parte de la planificación, y consiste en adjudicar los recursos a cada actividad, así como poner fechas concretas a todas las actividades, para poder obtener una fecha de realización total del proyecto. Existen diversos métodos para poder realizar esto, como son el método CPM, el método PERT, el método ROY y el método de las precedencias. Todos se basan en una serie de particularidades comunes de todos los proyectos.

Para poder cumplir con las fechas previstas, un tema crucial en la programación de proyectos será la distribución de los recursos disponibles de cada momento. Una mala asignación y programación de estos recursos hará que el proyecto no esté finalizado en los plazos y costes establecidos.

Los recursos que generalmente deben incluirse en un proyecto:

**HUMANOS:** son aquellos factores productivos que realizan el trabajo directo apoyados en sus habilidades y capacidades. En principio, utilizando una mayor cantidad de los mismos supondría disminuir el tiempo asociado a una tarea y por lo tanto a un proyecto, aunque el exceso de los mismos podría hacer inadecuado el trabajo. Los recursos humanos se rigen por una jornada y un calendario laboral dependiente de cada Comunidad Autónoma donde se pone de manifiesto cuántas horas disponibles tiene ese recurso al año.

**FINANCIEROS:** Recursos de capital necesarios para el pago de los factores productivos del proyecto. Reflejan las necesidades de caja o flujo financiero de un proyecto, derivados de los egresos e ingresos, con relación al tiempo. En este apartado podrían considerarse los costes de transporte (cambiar de un proyecto a otro cuando hay que desplazar continuamente la maquinaria o los recursos de un lugar para otro puede no ser rentable).

- **MATERIALES:** Recursos marcados en todo momento por las existencias almacenadas del mismo.
- **OTROS:** Tienen que ver en factores ajenos al proyecto, como puedan ser los equipos de que se disponen, la limpieza de la fábrica, la forma de almacenaje, dependencia de empresas externas...

Una vez definidas las actividades del proyecto se determinará, con ayuda de las diversas técnicas de redes, el camino crítico del proyecto así como las holguras para cada una de las actividades. Se elaborará el diagrama de Gantt del proyecto, quedando reflejadas en el mismo todas las actividades, su duración, así como las necesidades de recursos materiales, humanos y financieros para cada día del proyecto.

A partir de esta información se tendrá un diagrama de ocupación para cada uno de los recursos humanos del proyecto, que mostrará la carga de trabajo asignada por día a cada recurso. Teniendo en cuenta las disponibilidades de cada día, habrá que reajustar los recursos disponibles. El hecho de que los recursos sean tan limitados puede afectar a la planificación inicial del proyecto y en muchas ocasiones será necesaria una reprogramación de los recursos, intentando siempre que no afecte a la duración total del proyecto.

Entre las distintas alternativas al problema, podría recurrirse a la incorporación de nuevos trabajadores en momentos puntuales del proyecto, o modificarse las actividades no críticas que al disponer de holgura pueden desplazarse dentro de la misma cuando interese. Las actividades críticas por su parte, no pueden alterarse, pues un retraso en la misma produce un retraso en el proyecto.

Existen diversos métodos para la programación de los recursos en un proyecto, que van desde la programación lineal, como por ejemplo ZARE et al (2012) hasta diversos algoritmos heurísticos, que se analizarán en este artículo.

## **2. Objetivos**

A partir de la planificación y programación de un proyecto, se procede a determinar qué actividades son críticas, y cuáles son las holguras de cada una de ellas, obteniéndose el calendario óptimo de ejecución del proyecto. Esto se logra con la ayuda del diagrama de red del proyecto.

Llegados a este punto surge el problema de cómo lograr cumplir este calendario, teniendo en cuenta que cada actividad requiere una serie de recursos, tanto humanos como físicos, que no son indefinidos, sino en muchas ocasiones escasos. Este problema se ha llamado de Programación de Proyectos con Recursos Limitados (RCPSP: Resource Constrained Project Scheduling Problem).

El objetivo primordial de este artículo es realizar un análisis de las distintas técnicas heurísticas que solucionan este problema, desde las basadas en reglas de prioridad hasta las técnicas metaheurísticas. A partir de este análisis se van a determinar qué factores no

se tienen en cuenta, y cuáles se deberían incluir en el planteamiento del problema para acercarse lo más posible a la realidad.

### **3. Programación de Proyectos con Recursos Limitados (RCPS: Resource Constrained Project Scheduling Problem)**

#### **3.1. Introducción**

Antes de comenzar la ejecución del proyecto, se ha realizado la asignación de recursos de las diferentes actividades del proyecto. Esta asignación puede ser impuesta por la empresa o por nosotros mismos, obedecer a históricos existentes en cuanto a la realización de las actividades o bien el resultado de un exhaustivo estudio sobre cómo habría de realizarlas. Sin embargo, puede que dichas predicciones no sean las más adecuadas y exista una mala asignación.

Posibles causas de una mala asignación de recursos son:

- La imposibilidad para alcanzar los objetivos propuestos en el tiempo estimado, que harán necesario un replanteamiento del proyecto para reducir dicha duración. De cara a atajar este problema se puede disminuir el alcance de los objetivos (objetivos más modestos), o aumentar los recursos destinados a las actividades críticas para poder así reducir la duración de las mismas y por tanto del proyecto.
- La duración del proyecto es idónea, pero las curvas de carga muestran que, en ciertos instantes los recursos necesarios son mayores a los disponibles. Debería establecerse un orden en la realización de las tareas para que no se produzcan desequilibrios en la curva de carga, intentando no aumentar la duración del proyecto.
- En otras ocasiones, no hay problema alguno en cuanto a la posibilidad de alcanzar los objetivos en el plazo previsto ni situaciones de picos de carga, sin embargo las necesidades de los recursos de las diferentes tareas en el tiempo son muy variables. Habrá que plantearse la necesidad de eliminar recursos del proyecto o trasladar recursos de unas tareas a otras.

Posibles soluciones a los problemas anteriores pueden obtenerse habitualmente:

- Modificando las asignaciones y las duraciones de las distintas actividades, desplazando las tareas dentro de su holgura,
- Cambiando la estructura del proyecto, pudiendo dividir las tareas en subtareas, o modificar el orden de las actividades, etc.
- Una vez realizados los cambios oportunos se comprobará que el nuevo planteamiento del proyecto cumple mejor las expectativas.

A este tipo de problemas se les suele llamar RCPSP (Resource Constrained Project Scheduling Problem). La tabla 3.1.1 muestra un ejemplo de RCPSP, dónde sólo se necesita un tipo de recurso, del que se dispone una capacidad de 5 unidades.

Tarea	Duración	Nº recursos necesarios/día	Predecesoras
A	4	3	-
B	3	3	-
C	2	4	A
D	3	1	B
E	1	3	C
F	2	2	D

Tabla 3.1.1: Ejemplo RCPSP

Dos posibles soluciones al sencillo problema planteado se muestran a continuación [Figura 3.1.1 y Figura 3.1.2]

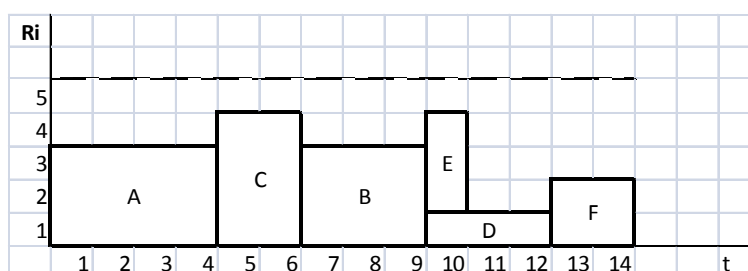


Figura 3.1.1: Diagrama Gantt del ejemplo RCPSP – posibilidad 1

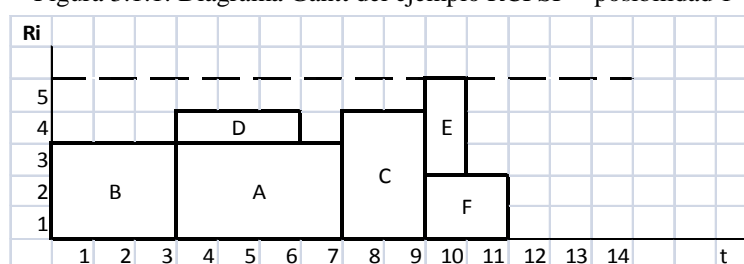


Figura 3.1.2: Diagrama Gantt del ejemplo RCPSP – posibilidad 2

Puede apreciarse que de una opción a otra hay una diferencia de 3 días en la programación del proyecto. Si en un proyecto que sólo tiene seis tareas y solo necesita un recurso hay una diferencia notable en el tiempo de realización del mismo, el problema se agrava cuando se trata de un proyecto de gran número de tareas y recursos quedando de manifiesto la gran importancia de un buen programa de cara a realizar los

Proyectos en el menor número de días y optimizando al máximo los recursos disponibles.

### 3.2. Punto de partida del problema de la nivelación de recursos

La nomenclatura de todos los factores a tener en cuenta en este tipo de problemas es la siguiente:

- J: Conjunto de todas las actividades del proyecto ( $J=\{1,2,\dots,n\}$ ).
- n: Número total de actividades que posee el proyecto.
- $d_j$ : Duración o tiempo de procesamiento de la actividad  $j$ . Es un valor entero, constante y no negativo.
- $F_j$  indica el instante final de la actividad  $j$ .
- H: Conjunto de todas las relaciones de precedencia existentes en el proyecto
- $P_j$ : Conjunto de todas las actividades predecesoras inmediatas de la actividad  $j$  ( $P_j \in J \setminus \{j\}$ ).
- K: Conjunto de todos los tipos de recursos requeridos por el proyecto ( $K=\{1,2,\dots,m\}$ ).
- m: Número total de tipos de recursos requeridos por el proyecto.
- $r_{jk}$ : Cantidad de unidades requeridas del tipo de recurso  $k$  para realizar la actividad  $j$  ( $r_{jk} \leq R_k$ ). Es un valor entero, constante y no negativo.
- $R_k$ : Representa la disponibilidad (capacidad) total del tipo de recurso  $k$ . Es un valor entero, constante y no negativo.
- $R_k(t)$ : Representa la suma de las unidades del tipo de recurso  $k$  utilizadas durante el Período  $t$  ( $R_k(t) \leq R_k$ ).
- G: Grafo que representa a todas las actividades con sus respectivas relaciones de precedencia ( $G=(J,H)$ ).

Con la resolución del problema, se intenta determina un vector de tiempos finales ( $F_1, F_2, \dots, F_n$ ) denominado secuencia, que respeta las relaciones de precedencia, la limitación de recursos y que hace que el tiempo de ejecución del proyecto sea mínimo.

Al conjunto de actividades que están en proceso (activas) en un momento del tiempo  $t$ , se le llama  $A(t) = \{j \in J \mid F_j - d_j \leq t < F_j\}$ .

### 3.3. Criterios a seguir para nivelar los recursos.

De cara a su realización, en un momento dado, cada tarea consume una cantidad determinada de recursos. A partir del diagrama de red del proyecto, y de las fechas de inicio y fin de cada actividad establecidas, se obtendrán las curvas de carga de cada uno de los recursos aplicados en un proyecto. La comparación de las curvas de carga de recursos necesarios con las curvas de carga de recursos disponibles puede poner de manifiesto la existencia de áreas de sobrecarga y subcarga que obstaculicen el logro de los objetivos prefijados en el plan.



Generalmente los recursos se nivelan en función de su coste y del diagrama de carga de los mismos. En primer lugar se ajustarán los recursos con mayor precio que posean un diagrama de carga menos uniforme.

Cuando se quiere reducir el nivel de ocupación de un recurso y existen varias actividades que utilizan el mismo, en primer lugar serán seleccionadas aquellas tareas que lo precisan en el momento en el que se produce la sobrecarga.

Posibles soluciones al problema de nivelación son:

- De entre las actividades que utilizan el recurso en sobrecarga, se tendrán en cuenta en primer lugar aquellas que puedan modificarse sin transformar dicha actividad en actividad crítica.
- A continuación las actividades que tienen más opciones a ser replanteadas son aquellas que más holgura posean, pudiendo mover dicha tarea dentro de la libertad que nos permita dicha holgura, teniendo en cuenta que el desplazamiento de una actividad puede nivelar la carga de un recurso pero desnivelarlo en otro o compromisos establecidos que impidan dichos movimientos de tareas.
- Otra posibilidad para solucionar el gran problema de la nivelación de recursos sería variar la intensidad en la programación de las tareas, alargando la duración de las actividades que tienen holgura estableciendo un orden prioridad generalmente consistente en alargar primero las actividades más baratas.
- Se podría recurrir al uso de recursos suplementarios en caso de necesidad, llegando a un compromiso entre el coste asociado al uso de los mismos y el coste que supondría un retraso en el proyecto.
- La programación intermitente de actividades: En ciertos casos existe la posibilidad de programación intermitente en una actividad y hace posible la descomposición en sus diversas partes. Estas subtareas pueden ser, así, ejecutadas separadamente y a intensidad constante, lo que da una mayor agilidad a la distribución de recursos.
- La modificación del programa en el tiempo. A esta medida solo se recurre cuando, tras probar con las demás, no se consiguen superar las desviaciones de carga existentes.

Su empleo conduce a un cambio radical en el concepto de camino crítico. La ruta crítica deja de estar formada por una sucesión continua de actividades y pasa a ser una combinación de tareas del camino de tiempo crítico y tareas con recursos limitados. Este nuevo camino se denomina "*camino crítico compuesto*".

Cada actividad, podrá desplazarse, aumentar su duración o ambas cosas a la vez, siempre bajo un criterio económico.

La problemática que se ha venido analizando hasta ahora sería una problemática acumulativa. No se puede superar el máximo de recursos disponibles en cada momento

y deberán ser distribuidos de la forma más óptima posible entre las actividades que deben realizarse de forma simultánea.

Otro tipo de problemática más compleja de analizar, sería la problemática de tipo disyuntivo, que plantean, por la razón que sea, la imposibilidad de realizar dos o más actividades a la misma vez. Las restricciones disyuntivas corresponden a fenómenos rígidos, con muy poco margen de maniobra, en contraposición a las restricciones acumulativas. Este tipo de restricciones, dan lugar a problemas combinatorios de muy difícil solución.

#### **4. Métodos heurísticos para el problema RCPSP**

Existen dos claras tipologías de métodos. Los basados en reglas de prioridad, y los basados en técnicas metaheurísticas.

##### ***4.1. Métodos heurísticos basados en reglas de prioridad***

Las reglas de prioridad se usan para determinar qué actividad es seleccionada del conjunto de actividades que son elegibles (Dg).

Las reglas de prioridad tienen tres componentes:

- Función  $v(j)$ : función que asigna a cada actividad  $j$  del conjunto de elegibles (Dg) un valor real.
- Método de selección: hay que describir el extremo del conjunto, bien la actividad que presente mayor valor  $v(j)$  ó la que posea menor valor  $v(j)$
- Desempate: regla para desempatar entre las actividades que tienen igual valor  $v(j)$ .

Los tipos más comunes de reglas de prioridad son:

- Mínimo tiempo de finalización más tardío de una tarea, descrito por Davis et al (1975).
- Holgura mínima, calculada como el tiempo de finalización más tardío menos el tiempo de comienzo más tardío. También se puede apreciar en Davis et al (1975).
- Tiempo de procesamiento más pequeño de las tareas, según Alvarez-Valdés et al (1989).
- Mayor número de tareas sucesoras, también según Alvarez-Valdés et al (1989).
- Peso posicional con mejor ranking: se calcula como la duración de una tarea más la duración de todas las tareas sucesoras de ésta. Descrito en Alvarez-Valdés et al (1989).
- Mínimo tiempo de comienzo más tardío de una tarea, que se calcula como el tiempo de finalización más tardío menos su duración. Expuesto por Kolisch (1995)

Los tipos de métodos heurísticos basados en reglas de prioridad se pueden clasificar en:

#### 4.1.1. *Métodos de un paso*

Son métodos que generan una única secuencia, empleando sólo una regla de prioridad para obtener una solución factible.

Fueron los primeros métodos heurísticos utilizados para resolver el problema de RCPSP debido a su facilidad de implementación y de comprensión, así como su rapidez.

Ejemplos de estos métodos son los propuestos por Valls et al (1992), Kolish (1996), Alvarez-Valdés y Tamarit (1989), Patterson (1973, 1976), Boctor (1990), Davis (1973) y en compañía de Patterson (1975), Cooper (1976), y Gasparotti (2011)

#### 4.1.2. *Métodos multipaso*

- Métodos con múltiples reglas de prioridad

Emplean múltiples secuencias así como diferentes reglas de prioridad. Un ejemplo de este tipo se puede ver en Boctor (1990), que empleando 7 reglas de prioridad, puede llegar a generar múltiples programas mediante el uso de combinaciones de esas reglas.

Otro ejemplo se tiene en Ulusoy y Özdamar (1989) donde emplean una combinación de 2 reglas para obtener 10 programas.

Un último ejemplo de este tipo lo muestra Baradaran et al (2009) donde plantea una regla de prioridad híbrida para los diagramas tipo PERT.

- Métodos de secuenciación adelante-atrás

En estos métodos no sólo se secuencia en el orden lógico del programa, sino que se combina con la secuenciación hacia atrás. Consiste en aplicarlo a la red de precedencias inversa. Ejemplos de estos métodos son los aportados por Li y Willis (1992), así como por Ulusoy y Özdamar (1989)

- Métodos de muestreo

Diferentes programas son obtenidos a partir de una serie de probabilidades y una regla de prioridad. Se trata de calcular una probabilidad de selección  $p(j)$  que representa la probabilidad de que se seleccione la actividad  $j$  del conjunto de actividades elegibles  $D_g$ , cuando se selecciona la actividad a secuenciar.

Como ejemplos de muestreo aleatorio están Alvarez-Valdés y Tamarit (1989), así como Cooper (1976).

Una modificación del mismo fue introducido por Schirmer y Riesenbergr (1997), llamada muestreo aleatorio sesgado normalizado. También está el muestreo aleatorio sesgado basado en la peor elección, que utiliza las prioridades de una manera indirecta a través de unos determinados valores llamados “regret”. Este último tipo de muestreo ha sido utilizado por Schirmer y Riesenbergr (1997) y Kolish (1995) entre otros.

## **4.2. Métodos heurísticos basados en técnicas metaheurísticas**

### **4.2.1. Introducción**

Las técnicas metaheurísticas son algoritmos de optimización de propósito general, capaces de proporcionar muy buenas soluciones en tiempo y con recursos razonables. Se basan en procesos iterativos, persiguiendo siempre la mejor solución en el mejor ámbito de búsqueda. Las propiedades de las técnicas metaheurísticas podrían resumirse en las siguientes:

- El objetivo es una búsqueda eficiente que encuentre soluciones casi óptimas.
- Son estrategias generales que guían el proceso de búsqueda.
- El procedimiento de cualquier metaheurística es genérico, no depende del problema.
- Intentan evitar la exploración en regiones del espacio de búsqueda no óptimas.
- Los métodos heurísticos empleados son específicos para cada caso.
- Una vez determinada una solución se desarrollarán funciones de bondad para valorar los resultados y poder estimar lo óptima que es la solución encontrada.

Dentro de estas técnicas, los métodos constructivos suelen ser los más rápidos, aunque las soluciones suelen ser de muy baja calidad. Añadiendo componentes a una solución inicial, se generará una solución completa.

### **4.2.2. Clasificación de las técnicas metaheurísticas**

En función del espacio de búsqueda utilizado, las técnicas metaheurísticas pueden clasificarse en dos tipos. Metaheurísticas basadas en trayectoria si se utiliza un único punto del espacio de búsqueda o basadas en población si se trabaja sobre un conjunto o población.

- Metaheurísticas basadas en trayectoria

Se irá formando una trayectoria partiendo de un punto inicial y mejorando la solución mediante el análisis del entorno más cercano. La búsqueda se dará por concluida una vez encontrada una solución suficientemente buena, no se pueden realizar más iteraciones por haber alcanzado el número máximo establecido o se produce un anquilosamiento en el proceso de búsqueda.

Dentro de las técnicas metaheurísticas basadas en trayectoria se encuentran:

- El Enfriamiento ó temple Simulado (SA ó “Simulated Annealing”)
  - La Búsqueda Tabú (TS ó “Tabu Search”)
  - La Búsqueda Local (LS ó “Local Search”)
  - Metaheurísticas basadas en población
- Estas técnicas operan con cada una de las posibles soluciones de un conjunto de individuos denominado población. La eficiencia de esta técnica radica principalmente en el tratamiento o manipulación de dicha población en cada iteración.

Los principales tipos de técnicas metaheurísticas basadas en población son las siguientes:

- Los Algoritmos Evolutivos
- La Búsqueda Dispersa (SS ó “Scatter Search”)
- Los sistemas basados en Colonias de Hormigas, (“Ant Colony Optimization”)
- Algoritmos Basados en Nubes de Partículas (ó Particle Swarm Optimization)
- Metaheurísticas híbridas: GRASP

#### 4.2.3. El Enfriamiento ó temple Simulado (SA ó “Simulated Annealing”)

Esta técnica es una analogía al proceso de tratamiento térmico de metales llamado temple o recocido, basado en Kirkpatrick et al (1983). A partir de una solución actual,  $X_0$ , se elegirá en cada iteración una solución  $X_1$ . Esta solución,  $X_1$ , se tomará como solución actual y sustituirá a la solución  $X_0$  siempre que sea mejor que ella. En el caso de que la nueva solución,  $X_1$ , sea de peor calidad, no será descartada sino que se le asignará una determinada probabilidad a fin de no caer en un óptimo local.

Características:

- -Se va modificando paulatinamente una solución mientras se produce una mejora, de modo que el proceso acaba en un óptimo local.
- -De cara a evitar quedar atrapado en un óptimo local, se permiten movimientos hacia soluciones peores. Estos movimientos de escape deben controlarse adecuadamente para no desviar la búsqueda cuando se dirija hacia una buena solución. Para ello se utiliza una función de probabilidad que disminuye la probabilidad de movimientos de escape cuando la búsqueda avanza.

Ejemplos de estos algoritmos para resolver el problema del RCPSP son Bouleimen y Lecocq (2003), Cho y Kim (1997), Lee y Kim (1996) y Sampson y Weiss (1993).

#### 4.2.4. La Búsqueda Tabú (TS ó “Tabu Search”)

Fue desarrollada por Glover (1988 y 1990), Su estrategia está basada en la búsqueda de las mejores soluciones en distintas regiones de búsqueda, memorizando y guardando dichas soluciones en una lista denominada lista tabú. Estas soluciones serán excluidas de iteraciones posteriores.

Conceptos claves de TS:

- Principio de TS: Memoria + Aprendizaje = Búsqueda inteligente.
- Memoria Adaptativa: memoria selectiva, incluyendo olvido estratégico.
- Exploración sensible: se concentra en buscar buenas características de las soluciones
- Formas de elaborar la lista Tabú
  - Memoria Explícita: memoriza en la lista tabú toda la solución ya visitada, por ejemplo: Solución tabú = (1,0,1,1,1,0,1,1)
  - Memoria Atributiva: no se memoriza toda la solución, sino que se memorizan atributos, es decir, características importantes de las soluciones ya visitadas, por ejemplo: Atributo tabú: = (0, \*, \*, \*, \*, \*, \*, 1)
- La lista Tabú debe ser dinámica.
- Tipos de Estrategias: están asociadas en la memoria a largo plazo
  - Intensificar: consiste en regresar a buenas regiones ya exploradas para estudiarlas más a fondo. Para ello se favorece la aparición de aquellos atributos asociados a buenas soluciones encontradas.
  - Diversificar: consiste en visitar nuevas áreas no exploradas del espacio de soluciones. Para ello se modifican las reglas de elección para incorporar a las soluciones atributos que no han sido usados frecuentemente.
- Criterios de finalización del proceso:
  - cuando ya se ha realizado un número de iteraciones  $k$  sin mejorar el valor de nuestra función objetivo.
  - ó cuando se considere que el valor obtenido es cercano al óptimo esperado.

Para el problema del RCPSP diversos autores han buscado la solución mediante esta metodología: Baar et al (1998), Lee y Kim (1996), Pinson et al (1994) y Thomas y Salhi (1998).

#### 4.2.5. La Búsqueda Local (LS ó “Local Search”):

Los métodos de búsqueda local van explorando el entorno siempre dentro del espacio de búsqueda para encontrar un óptimo local a partir de una solución completa. Habrá un operador de movimiento que será determinante a la hora de explorar el

vecindario y poder encontrar el óptimo local perseguido en este método. Partiendo de una solución inicial, se elige una mejor solución, examinando su entorno o vecindario. El proceso continuará hasta que en la búsqueda no exista un vecino mejor o el proceso quede estancado.

Tras cada iteración, la solución actual será sometida a un cambio que la convierta en una solución intermedia. Para mejorarla, esta nueva solución será supeditada a una heurística. Tras el proceso de mejora y siempre que pase un test de aceptación, el nuevo óptimo local pasará a ser la nueva solución actual.

Como aplicaciones al problema del RCPSP están Leon y Ramamoorthy (1995) y Naphade et al (1997).

#### 4.2.6. Los Algoritmos Evolutivos

Los algoritmos evolutivos se basan en la posibilidad de que cada individuo represente una solución y en cómo cambian esas soluciones con los cambios de su entorno, se adaptan o mueren.

El proceso sería el siguiente: Los individuos que conforman la población serán elegidos aleatoriamente. Posteriormente a cada ser se le otorgará un valor en función del problema planteado, será un valor en función de sus capacidades, aptitudes, su capacidad para adaptarse.... Ese valor de cada individuo será el punto de partida del algoritmo para realizar la búsqueda. Seleccionar, recombinar o mutar la población son las posibles acciones a tener en cuenta cuando se desea modificar la población.

Fases del algoritmo evolutivo:

- Una primera fase donde se seleccionan buenas soluciones o individuos.
- Una segunda fase dónde se buscan nuevas soluciones en regiones no exploradas anteriormente.

Dentro de los algoritmos evolutivos el más utilizado es el Algoritmo Genético (AG ó Genetic Algorithm GA).

Los Algoritmos Genéticos, expuestos por Holland (1975), intentan simular la selección natural y la supervivencia del más fuerte desarrollados por Darwin. El Algoritmo Genético es capaz de crear soluciones y mejorarlas hacia valores óptimos.

Pueden ir perfeccionando su propia heurística en el proceso de ejecución, por lo que no requiere largos períodos de entrenamiento especializado.

Para la resolución del problema planteado del RCPSP se han propuesto diversas soluciones, como Alcaraz y Maroto (2001), Hartmann (1998, 2002), Lee y Kim (1996) y Leon y Ramamoorthy (1995).

#### 4.2.7. La Búsqueda Dispersa (BD ó SS “Scatter Search”)

Método desarrollado por Glover (1998). La Búsqueda Dispersa (BD) se basa en el principio de que la información sobre la calidad o el atractivo de un conjunto de reglas, restricciones o soluciones puede ser utilizado mediante la combinación de éstas. En concreto, dadas dos soluciones, se puede obtener una nueva mediante su combinación de modo que mejore a las que la originaron. Según esto, cuando por ejemplo se crean nuevas soluciones a partir de una combinación lineal de otras dos o más, el conjunto de referencia puede evolucionar.

Utilizando esta metodología Vanhoucke (2010) propone una solución para los problemas de RCPSPP intentando minimizar los costes asociados a cada tarea.

#### 4.2.8. Los sistemas basados en Colonias de Hormigas, (“Ant Colony Optimization”)

El método encuentra su inspiración en las hormigas y en su forma de buscar la comida: en un primer momento las hormigas eligen arbitrariamente una zona cercana al hormiguero. En el caso de encontrar comida, la llevaría a su hormiguero, dejando un rastro debido a una sustancia química denominada feromona, que les permitiría al resto de hormigas encontrar dicha comida. Las hormigas de las siguientes generaciones son atraídas para realizar su búsqueda en el espacio de solución cercano a las buenas soluciones previas, encontradas en generaciones anteriores. La feromona presente en los caminos sirve como mecanismo de comunicación, que las hormigas usan de forma probabilística para construir soluciones a un problema y que luego se adaptan para reflejar la experiencia de las hormigas. Se trata de una estrategia de búsqueda distribuida en la que un conjunto de agentes coopera en la búsqueda, transmitiendo e intercambiando información al resto.

Merkle et al (2002) aplicó las colonias de hormigas al problema del RCPSPP.

## 5. Conclusiones

Para planificar un proyecto es necesario definir cada una de las tareas a realizar y establecer el orden de las mismas, esto es, su secuenciación. Habrá que estimar la duración y la necesidad de recursos de dichas actividades así como la previsión de los recursos disponibles para la realización del proyecto.

Para que un proyecto finalice en plazo, es de crucial importancia que la programación de los recursos disponibles a las actividades se haga de forma rigurosa. La Programación de Proyectos con Recursos Limitados (RCPSPP: Resource Constrained Project Scheduling Problem) aborda estos temas y tiene en cuenta ciertos factores que los métodos tradicionales de nivelación de recursos adolecen, resultando ineficientes a la hora de realizar una buena programación del proyecto.

Las técnicas de programación lineal implican la creación de un modelo propio para cada proyecto, con sus restricciones particulares. La creación de este modelo puede en



muchos casos ser creación trivial, pero arduo y complicado en la mayoría de proyectos reales pues habría que parametrizar todas las actividades, duraciones, recursos y restricciones del proyecto.

Las técnicas tradicionales que se utilizan para la nivelación de recursos adolecen de no tener en cuenta muchas consideraciones. La mayoría se basan en diagramas de precedencia que únicamente tienen en cuenta relaciones de final de una actividad con el comienzo de otra, siendo este tipo de restricciones en ocasiones insuficiente.

Otro aspecto que no suelen considerar es la posibilidad de que una tarea puede tener diversas duraciones, en función de los recursos empleados por día en su realización. Esta realidad da mayores posibilidades para un mucho mejor aprovechamiento de los recursos disponibles.

Para solucionar algunos de estos temas se han desarrollado los métodos enumerados para la resolución de problemas RCPSP. Su planteamiento matemático se encuentra con grandes dificultades para encontrar una solución, por lo que se han desarrollado numerosas técnicas heurísticas. En este artículo se ha hecho un análisis de las distintas técnicas, viendo las reglas de prioridad más usadas y sus características, así como de las técnicas metaheurísticas más utilizadas para el problema del RCPSP.

Estas técnicas, aunque dan una perspectiva mucho mayor que las tradicionales adolecen también en ciertos aspectos, como es el caso de que no consideran la posibilidad de tener hitos intermedios a cumplir en los proyectos, ni permiten en general que una tarea pueda ser realizada en partes, aun cuando técnicamente sea posible.

## **6. Futuras líneas de investigación**

Se han estudiado las diversas técnicas de programación de proyectos con recursos limitados (RCPSP), y viendo las debilidades que presentan en algunos aspectos, una futura línea de investigación será el dotarlas de todo tipo de restricciones entre las diversas tareas de un proyecto, y que no sean exclusivamente del final de una actividad con el comienzo de otra. Así mismo los métodos vistos para la resolución del RCPSP no consideran la posibilidad de tener hitos intermedios a cumplir en los proyectos. Este sería otro factor a incluir.

La posibilidad de dividir una tarea en partes, si es viable técnicamente, también puede dar una gran flexibilidad a la asignación de los proyectos, aunque complica aún más su solución. A pesar de ello se considera que es una oportunidad que debe ser tenida en cuenta en nuestras futuras investigaciones.

Existe una gran variedad de métodos, como los vistos en este trabajo, para entornos de programación con un solo Proyecto. La adaptación de estos métodos a entornos multiproyecto prácticamente no se ha desarrollado, y hay muy pocas referencias. Las prioridades que tengan la realización de cada uno de los Proyectos involucrados será también un factor fundamental a considerar.

## 7. Bibliografía

- ALCARAZ, J. & MAROTO, C. (2001): *A robust genetic algorithm for resource allocation in project scheduling*. Annals of Operations Research 102, pp.83–109.
- ALCARAZ, J. & MAROTO, C. & Ruiz, R. (2004): *Improving the performance of genetic algorithms for the RCPS problem*. Proceedings of the Ninth International Workshop on Project Management and Scheduling, Nancy, pp 40–43, Nancy.
- ALVAREZ-VALDÉS, R. & TAMARIT, J.M. (1989): *Heuristic algorithms for resource-constrained project scheduling: A review and an empirical analysis*. In R. Slowinski and J. Weglarz, editors, Advances in Project scheduling, Elsevier, Amsterdam, the Netherlands, pp. 113–134.
- BAAR, T. & BRUCKER, P. & KNUST, S. (1998): *Tabu-search algorithms and lower bounds for the resourceconstrained project scheduling problem*. In S. Voss, S. Martello, I. Osman, and C. Roucairol, editors, Meta-heuristics: Advances and trends in local search paradigms for optimization, Kluwer Academic Publishers, pp. 1–8.
- BALLESTÍN, F. (1999): *Nuevos Procedimientos de búsqueda en el espacio de soluciones del problema de secuenciación de Proyectos con recursos limitados*. Trabajo de investigación. Universidad de Valencia.
- BARADARAN, S. & FATEMI GHOMI, S.M.T. (2009): *A hybrid heuristic rule for constrained resource allocation in PERT type network*. World Applied Sciences Journal 7 (10): 1324-1330, 2009.
- BOCTOR, F.F. (1990): *Some efficient multi-heuristic procedures for resource-constrained project scheduling*. European Journal of Operational Research, 49, pp. 3–13.
- BOZORG HADDAD, O. et al (2008): *Finding the shortest path with honey-bee mating optimization algorithm in project management problems with constrained unconstrained resources*. Comput Optim Appl 47, pp. 97–128.
- BOULEIMEN, K. & LECOCQ, H. (2003): *A new efficient simulated annealing algorithm for the resource-constrained project scheduling problem and its multiple modes version*. European Journal of Operacional Research, 149, pp. 268–281.
- CERTA, A. et al (2009): *Multi-objective human resources allocation in R&D projects planning*. International Journal of Production Research. Vol. 47, No. 13, pp. 3503–3523.
- CHO, J. H. & KIM, Y. D. (1997): *A simulated annealing algorithm for resource-constrained project scheduling problems*. Journal of the Operational Research Society, 48, pp. 736–744.
- COOPER, D. F. (1976): *Heuristics for scheduling resource-constrained projects: An experimental investigation*. Management Science, 22:1186–1194, 1976.
- DE COS CASTILLO, M. (1995), *Teoría general del proyecto*. Ed. Síntesis, Madrid.

- DAVIS, E. W. (1973): *An experimental investigation of resource allocation in multactivity projects*. Operations Research Quarterly, 24, pp. 587–591.
- DAVIS, E. W. & PATTERSON, J. H. (1975): *A comparison of heuristic and optimum solutions in resource–constrained project scheduling*. Management Science, 21, pp. 944–955.
- DE LA PEÑA ESTEBAN, F. D. & SIMÓN RODRÍGUEZ, M. et al (2005) [en línea]: *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 ROY*. Tecnología y desarrollo, ISSN 1696-8085, N.º. 3, [[http://www.uax.es/publicaciones/archivos/TECTIN05\\_001.pdf](http://www.uax.es/publicaciones/archivos/TECTIN05_001.pdf)]
- GASPAROTTI, CARMEN (2011): *Resource allocation within a project using heuristic algorithm*. Review of Management & Economic Engineering. 2011, Vol. 10 Issue 2, pp. 35-46.
- GLOVER, F. (1989): *Tabu Search – Part I*, ORSA Journal on Computing 1, pp. 190-206.
- GLOVER, F. (1990): *Tabu Search – Part II*, ORSA Journal on Computing 2, pp. 4-32.
- GLOVER F. (1998): *A Template for Scatter Search and Path Relinking*. Artificial Evolution. Springer LNCS 1363, pp. 13-64.
- HARTMANN, S. (1998): *A competitive genetic algorithm for resource-constrained project scheduling*. Naval Research Logistics, 45, 733–750.
- HARTMANN, S. (2002): *A self-adapting genetic algorithm for project scheduling under resource constraints*. Naval Research Logistics, 49, pp. 433–448.
- KIRKPATRICK, S. & GELATT JR., C. D. & VECCHI, M. P. (1983): *Optimization by Simulated Annealing*. Science 13. Vol. 220. no. 4598, pp. 671 – 680.
- HOLLAND, J. (1975): *Perspectives on Adaptation in Natural and Artificial Systems*. University of Michigan.
- HORBACH, A. (2010). *A Boolean satisfiability approach to the resource-constrained project scheduling problem*. Ann Oper Res 181, pp. 89-107.
- J. FERNÁNDEZ MARTÍNEZ, A. MARTÍNEZ ARGOTE, K. OLALDE AZKORRETA (1999). *Departamento de Expresión Gráfica y Proyectos de Ingeniería. Escuela Universitaria de Ingeniería Técnica Industrial e Ingeniería*.
- KOLISCH, R. (1995): *Project scheduling under resource constraints — Efficient heuristics for several problem classes*. Physica, Heidelberg.
- KOLISCH, R. (1996): *Efficient priority rules for the resource-constrained project scheduling problem*. Journal of Operations Management, 14, pp. 179–192.
- KOLISCH, R. & HARTMANN, S. (1998): *Heuristic Algorithms for Solving the Resource-Constrained Project Scheduling Problem*. Handbook on recent advances in Project scheduling.

- LEE, J. K. & KIM, Y. D. (1996): *Search heuristics for resource-constrained project scheduling*. Journal of the Operational Research Society, 47, pp. 678–689..
- LEON, V. J. & RAMAMOORTHY, B.(1995): *Strength and adaptability of problem-space based neighborhoods for resource-constrained scheduling*. OR Spektrum, 17, pp. 173–182.
- LI, K. Y. & WILLIS, R. J. (1992): *An iterative scheduling technique for resource-constrained project scheduling*. European Journal of Operational Research, 56, pp. 370–379.
- MERKLE, D. & MIDDENDORF, M. & SCHMECK, H. (2002): *Ant colony optimization for resource-constrained project scheduling*. IEEE Transactions on Evolutionary Computation, 6, pp. 333–346.
- NAPHADE, K. S. & WU, S. D. & STORER, R. H. (1997): *Problem space search algorithms for resource-constrained project scheduling*. Annals of Operations Research, 70, pp. 307–326.
- PADMAN, R. & ZHU, D. (2006): *Knowledge integration using problem spaces: a study in resource-constrained project scheduling*. J Scheduling 9, pp. 133-152.
- PATTERSON, J. H. (1973): *Alternate methods of project scheduling with limited resources*. Naval Research Logistics Quarterly, 20, pp. 767–784.
- PATTERSON, J. H. (1976): *Project scheduling: The effects of problem structure on heuristic performance*. Naval Research Logistics Quarterly, 23, pp. 95–123.
- PINSON, E. & PRINS, C. & RULLIER, F. (1994). *Using tabu search for solving the resource-constrained Project scheduling problem*. In Proceedings of the fourth international workshop on project management and scheduling, Leuven, Belgium, pp. 102–106.
- SAMPSON, S. E. & WEISS, E. N. (1993): *Local search techniques for the generalized resource-constrained project scheduling problem*. Naval Research Logistics, 40, pp. 665–675.
- SCHIRMER, A. & RIESENBERG, S. (1997): *Parameterized heuristics for project scheduling — Biased random sampling methods*. Manuskripte aus den Instituten für Betriebswirtschaftslehre 456, Universität Kiel, Germany-.
- VALLS, V. & PEREZ, M. A. & M. S. QUINTANILLA, M. S. (1992): *Heuristic performance in large resource-constrained projects*. Technical Report 92-2, Department of Statistics and Operations Research, University of Valencia.
- VANHOUCKE, M. (2010): *A scatter search heuristic for maximising the net present value of a resource-constrained project with fixed activity cash flows*. International Journal of Production Research. Vol. 48, No. 7.

ZARE, ZAHRA et al (2012): *Proposing a Model on Preemptive Multi-mode Resource-constrained Project Scheduling Problem*. International Journal of Business & Social Science. 2012, Vol. 3 Issue 4, pp. 126-130.