

OPTIMIZACIÓN DE RECURSOS MEDIANTE

COLONIA DE HORMIGAS

Paula Areces, Carlos Alba

ArcelorMittal Global R&D Asturias

Joaquín Villanueva

Universidad de Oviedo, Área de Proyectos de Ingeniería

Alberto Gómez

Universidad de Oviedo, Área de Organización de Empresas

Abstract

The planning of projects with limited resources is one of the classic problems in Project Management. Having a limited set of resources, the stages and tasks of the project, with their durations and precedence relations, the aim is to reach the best planning in terms of deadlines or cost.

In function of the number of tasks and resources, solving this problem can be trivial or something impossible to solve algorithmically in a reasonable time. This study proposes a solution based on metaheuristic techniques, particularly Ant Colony Optimization (ACO). In addition to the original problem, they will be permitted flexible and inflexible constraints, i.e. suggested end date for the project or to forbid enlarging the duration of the project to solve the overloaded resources, respectively. It will be possible to assign penalty costs due to delays, extra hours, transport costs, idle cost and relations among resources.

It is presented in this work a classification of the different resource scheduling problems and a general solution based in ACO. A simple example is included to easily understand the potential behind that could be used as a technique inside any stage of a methodology for resource leveling and project planning.

Keywords: *Resource Leveling, Project Planning, ACO*

Resumen

La planificación de proyectos con recursos limitados es uno de los problemas clásicos en la Dirección de Proyectos. A partir de un conjunto de recursos limitado, las fases y tareas del proyecto, con sus duraciones y relaciones de precedencia, se pretende obtener la mejor planificación en términos de plazo o coste.

En función del número de tareas y recursos, la resolución del problema puede ser trivial o no resoluble algorítmicamente en un tiempo razonable. Este estudio propone una solución basada en técnicas metaheurísticas, en particular, Optimización por Colonia de Hormigas (ACO). Añadido al problema original, se permitirán restricciones flexibles e inflexibles, como fecha sugerida de finalización de proyecto o no permitir alargar la duración del proyecto para resolver la sobreasignación de recursos, respectivamente. Se permitirá asignar costes de

penalización por retrasos, asignación de horas extra, costes por transporte, tiempo ocioso y relaciones entre recursos.

Se presenta en este trabajo una clasificación de los problemas de programación de recursos y una solución genérica basada en ACO. Se incluye un ejemplo sencillo para comprender fácilmente el potencial que podría ser usado como técnica dentro de alguna fase en metodologías para nivelación de recursos y planificación de proyectos.

Palabras clave: Nivelación de Recursos, Planificación de Proyectos, ACO

1. Introducción

La planificación es una de las competencias fundamentales en la dirección de proyectos, con gran impacto en muchos campos: aeronáutica, comunicaciones, enrutamiento, control industrial o planificación de la producción. Derivado de este problema y en relación con la asignación de recursos surgen gran variedad de familias de problemas de optimización. En este trabajo se estudia el caso de la planificación de proyectos con recursos limitados (Resource-Constrained Project Scheduling Problem - RCPSP). Esta planificación se realiza teniendo en cuenta restricciones temporales, de disponibilidad de recursos y de precedencia entre las tareas a planificar.

Aunque el RCPSP es un problema NP-duro, existen aproximaciones exactas como las de (Brucker et al, 1998) o (Sprecher, 2000). Sin embargo este enfoque no es viable en la mayoría de los casos por lo que es necesario recurrir a heurísticos. En la literatura pueden encontrarse muchas opciones para su resolución, siendo los metaheurísticos los más extendidos. En este artículo se estudia la utilización de los algoritmos de Optimización de Colonias de Hormigas (Ant Colony Optimization - ACO), desarrollado por Dorigo y Di Caro en 1999, que han demostrado ser una buena opción en la resolución de problemas de optimización combinatoria como el RCPSP.

El artículo se organiza de la siguiente forma. En primer lugar se describen brevemente los distintos problemas RCPSP derivados del mismo. A continuación se explica la técnica de resolución elegida y su funcionamiento, el metaheurístico ACO. En el siguiente apartado se describe un ejemplo sencillo que permite comprender fácilmente el funcionamiento del algoritmo, para terminar con las conclusiones obtenidas.

2. Planificación de Proyectos con Recursos Limitados

El problema de planificación de proyectos con recursos limitados (RCPSP) es un problema clásico dentro de la dirección de proyectos que considera un conjunto de tareas, con sus respectivas duraciones y necesidades de recursos, que deben ser planificadas sujetas a restricciones de precedencia y de disponibilidad de recursos, siendo el objetivo del problema la obtención de una planificación que minimice la duración (o makespan) del proyecto.

De forma más formal el problema RCPSP puede ser definido como un conjunto de tareas $A = \{1, \dots, i, \dots, n\}$ y un conjunto de recursos $R = \{1, \dots, k, \dots, m\}$. Cada tarea i tiene una duración conocida $d_i \in \mathbb{N}$ y requiere una cantidad de recursos $r_{ik} \in \mathbb{N}$ siendo $k \in R$ el tipo de recurso necesario. Cada recurso k tiene una disponibilidad por periodo a_k . Sea ρ el conjunto de precedencia entre tareas donde $p_{ij} \in \rho$ indica que la tarea j sucede a la tarea i . La solución de un RCPSP es una planificación que consiste en una lista de instantes de inicio de tareas $s = (s_0, s_1, \dots, s_n, s_{n+1})$ siendo s_0 y s_{n+1} dos tareas ficticias de duración igual a cero y sin necesidad de recursos que representan el inicio y el fin del proyecto.

Las asunciones que definen el problema RCPSP hacen que en la práctica se requiera añadir nuevas características para que se ajuste a las situaciones del mundo real. En los estudios de (Yang et al., 2001) y (Hartmann et al., 2010) se pueden encontrar dos recopilaciones de los tipos de RCPSP que pueden aparecer en función de diferentes aspectos. A continuación se describen algunas de estas principales variantes o extensiones del problema RCPSP que permiten comprender la variedad de problemas derivados del RCPSP básico.

2.1 En función de las tareas

En esta sección se describen brevemente las variantes del problema RCPSP en relación a cómo se comportan las actividades del proyecto a planificar.

2.1.1. RCPSP con tareas mono-modo (SM-RCPSP)

Es el caso básico en el que las tareas son mono-modo, es decir, su duración así como los recursos que requiere son fijos y no cambian a lo largo de la vida del proyecto.

2.1.2. RCPSP con tareas multi-modo (MM-RCPSP)

El problema RCPSP estándar considera que las tareas tienen una duración y unos recursos requeridos constantes, sin embargo en el RCPSP multimodo las tareas pueden tener asociados un conjunto de posibles modos de ejecución. Cada uno de estos modos se caracteriza por una duración y los recursos que requiere para poder completarse. Así una tarea t_i podría tener un conjunto de modos $M_i = \{(d_{i1}, R_{i1}), \dots, (d_{in}, R_{in})\}$.

(Jarboui et al., 2008) trata el problema MM-RCPSP con recursos no renovables. (Sabzehparvar et al., 2008) considera el problema multimodo con restricciones generales sobre los recursos. (Li et al., 2008) extienden el concepto de modo como medida de la calidad, es decir, el modo tiene una calidad asociada y la planificación debe alcanzar un mínimo de calidad, por tanto la función objetivo busca maximizar la calidad global del proyecto. (Tiwari et al., 2009) consideran la calidad también en sus estudios multimodo de forma que una tarea debe ser iniciada con un recurso con menos calidad de la requerida y finalizada por un recurso con la calidad necesaria.

2.1.3 RCPSP con tareas apropiativas

En el RCPSP clásico se asume que una tarea no puede ser interrumpida una vez comienza a ejecutarse. RCPSP con tareas apropiativas considera que una tarea sí puede ser interrumpida por otra tarea en determinados instantes durante su tiempo de ejecución.

(Davis et al., 1971), (Kaplan, 1988), (Bianco et al., 1999), (Brucker et al., 2001) y (Debels et al., 2008) han realizado estudios considerando actividades apropiativas.

2.1.4. RCPSP con tareas con necesidad de recursos variable en el tiempo

En el RCPSP básico se considera que los recursos que requiere una actividad son constantes a lo largo de su ejecución, es decir, si se denota con r_{ik} la cantidad de recursos de tipo k que requiere una tarea i y la tarea tiene una duración d_i , entonces en cada uno de esos periodos la tarea requerirá r_{ik} unidades.

En el RCPSP con necesidad de recursos variable en el tiempo, el uso de los recursos durante la ejecución de la tarea puede variar. Así continuando con la nomenclatura anterior y generalizándola (se denota por r_{ikt} la cantidad de recursos de tipo k que requiere una

tarea i en el instante t), se tiene una solicitud de recursos variable durante la ejecución de la tarea.

Esta variación tiene una extensión que consiste en una acotación del consumo por periodo, es decir, que en lugar de tener un consumo exacto por unidad de tiempo se tiene un consumo mínimo y un consumo máximo.

(Cavalcante et al, 2001) y recientemente (Drezet et al, 2008) han realizado estudios considerando la variabilidad en la demanda de recursos por tarea. El estudio de Drezet et al desarrolla el caso en que los recursos son desarrolladores software donde se manejan restricciones de número de recursos máximo y mínimo por periodo y el número máximo de horas que legalmente puede trabajar el desarrollador por día.

2.1.5 RCPSP con tareas con tiempos de set-up

En algunos casos puede suceder que un recurso deba ser preparado antes de que una tarea comience, a este intervalo se le denomina tiempo de set-up. Un ejemplo que puede ilustrar esta variación es el caso en el que el recurso es una máquina.

(Mika et al, 2008) considera varios tipos de tiempos de set-up y los introduce en el RCPSP multimodo en el que estos tiempos de preparación se representan añadiendo tareas con modos específicos.

(Nonobe et al., 2002) introducen los tiempos de set-up también como tareas extra que además deben ser completadas antes del inicio de la tarea que requiere la preparación.

2.2. En función de las restricciones temporales

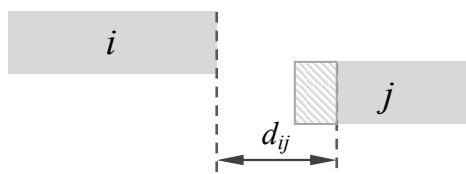
En este apartado se describen brevemente las variantes con diferentes restricciones relativas al tiempo.

2.2.1. RCPSP con periodos máximos

En el RCPSP básico las relaciones de precedencia entre tareas son de tipo inicio-fin, esto es, que una tarea debe haber finalizado antes de que sus sucesoras puedan comenzar.

En el RCPSP con periodos máximos la precedencia de tareas se extiende de modo que se acota el momento en que comienzan las tareas sucesoras. Así se denota por TF_i el instante de finalización de la tarea i , por TI_j el instante de inicio de la tarea j y por d_{ij} el tiempo máximo entre TF_i y TI_j . Entonces $TF_i + d_{ij} \leq TI_j$, es decir, el intervalo máximo entre la finalización de la tarea i y el inicio de la tarea j es d_{ij} .

Figura 1 Intervalo máximo entre 2 tareas con relación de precedencia

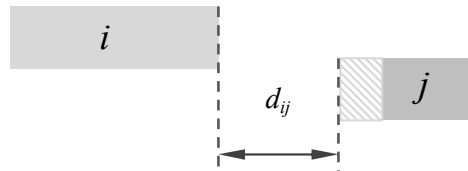


(Heilmann et al., 2003), (Sabzehparvar et al., 2008) y (Barrios et al., 2009) consideran los intervalos máximos ligados al problema MM-RCPSP.

2.2.2. RCPSP con periodos mínimos

Del mismo modo, existe el RCPSP con periodos mínimos. Se denota por TF_i el instante de finalización de la tarea i , por TI_j el instante de inicio de la tarea j y por d_{ij} el tiempo máximo entre TF_i y TI_j . Entonces $TF_i + d_{ij} \leq TI_j$, es decir, el intervalo mínimo entre la finalización de la tarea i y el inicio de la tarea j es d_{ij} .

Figura 2 Intervalo mínimo entre 2 tareas con relación de precedencia



(Drexler et al., 2000) consideran un problema multimodo con intervalos mínimos que dependen de los modos. Recientemente, (Chassiakos et al., 2005) y (Vanhoucke et al., 2006) también los han considerado en sus estudios.

2.2.3. RCPSP con periodos prohibidos

El RCPSP con periodos prohibidos introduce periodos en los que una actividad no puede ser ejecutada.

2.2.4. RCPSP con duraciones continuas

El RCPSP clásico asume que las duraciones de las tareas son valores enteros. Una extensión de esta idea consiste en permitir que las duraciones de los trabajos sean de tipo continuo. (Rom et al., 2002) presentan un modelo matemático con una línea temporal que no se encuentra partida en periodos de igual duración.

2.3. En función de los recursos

A continuación se describen las variantes sobre el problema RCPSP básico en relación a los recursos.

2.3.1. RCPSP con recursos renovables

En el RCPSP básico los recursos son renovables, es decir, los recursos que requiere la actividad están completamente disponibles en cada periodo de ejecución de la tarea. Recursos renovables como la mano de obra o las máquinas están limitados por periodo.

(Demeulemeester et al., 1997) presentan un procedimiento de ramificación y poda basado en el concepto de combinación de actividad-modo maximal. Una combinación actividad-modo es un subconjunto de tareas ejecutadas en un modo específico y es maximal si ninguna otra tarea puede añadirse sin causar un conflicto de recursos. (Sprecher, 1998) desarrolla también un procedimiento de ramificación y poda basado en el concepto de árbol de precedencias. (Hartmann, 1997) aplica un algoritmo genético para su resolución.

2.3.2. RCPSP con recursos no renovables

En el RCPSP con recursos no renovables estos tienen una limitación global al proyecto no por periodo.

Los recursos no renovables sólo deben considerarse en RCPSP multi-modo.

(Kelly, 1963), (Fulkerson, 1961), (Hindelang et al., 1997), (Harvey et al., 1979) y (Skutella, 1997) han realizado estudios sobre este problema desde dos perspectivas: el problema del presupuesto y el problema de fecha límite.

2.3.3. RCPSP con recursos doblemente limitados

En el caso de recursos doblemente limitados, éstos se encuentran restringidos a dos ámbitos, por periodo y a nivel global del proyecto.

2.3.4. RCPSP con recursos parcialmente renovables

El RCPSP con recursos parcialmente renovables introduce la idea de que cada recurso tiene definido un conjunto de sub-periodos para los que se indica la disponibilidad del recurso.

Como ejemplo de un recurso parcialmente renovable, podríamos tener un empleado que trabaja de lunes a viernes y el sábado o el domingo, pero no ambos. De este modo cada día del lunes a viernes sería un periodo con disponibilidad del empleado igual a 1 y el sábado y el domingo compondrían conjuntamente un único periodo con disponibilidad 1.

2.3.5. RCPSP con producción/eliminación de recursos

En el RCPSP con producción/eliminación de recursos los recursos pueden ser creados (aumenta la disponibilidad del recurso) o retirados (disminuye la disponibilidad del recurso) por las tareas.

2.3.6. RCPSP con recursos continuos

En el RCPSP básico la disponibilidad de los recursos se considera únicamente con valores de tipo discreto (por ejemplo, mano de obra o máquinas). Los recursos continuos permiten dividir su disponibilidad de manera continua, lo cual resulta de gran importancia en recursos como la energía o las materias primas.

2.3.7 RCPSP con recursos dedicados

Los recursos dedicados son aquellos que pueden ser únicamente asignados a una actividad al mismo tiempo y pueden ser representados como recursos renovables con disponibilidad por periodo igual a 1.

2.3.8. RCPSP con disponibilidad de recursos dependiente del tiempo

En el RCPSP clásico se asume que los recursos renovables tienen una disponibilidad constante, es decir, la disponibilidad del recurso es la misma en todos los periodos de tiempo. Sin embargo esto supone una gran limitación que en muchos casos no se ajusta a la realidad.

Esta variación sería un caso especial de los recursos parcialmente renovables en el que se define un sub-periodo para cada intervalo con una capacidad propia.

2.4. En función del objetivo

El RCPSP básico considera como función objetivo la minimización de la duración del proyecto, sin embargo existen diferentes funciones objetivo que pueden tenerse en cuenta en la resolución del problema.

2.4.1. Objetivo basado en tiempo

- Función objetivo basada en el retraso (lateness y tardiness)

Lateness es la diferencia entre el instante previsto de finalización de una tarea y el instante real. Tardiness es similar a lateness pero con la diferencia de que no puede ser negativo.

- Función objetivo basada en tiempos tempranos (earliness)

Earliness es la diferencia entre el instante real de finalización de una tarea y el instante previsto. Debe ser mayor que cero.

2.4.2. Objetivo basado en recursos no renovables

En el caso de querer establecer la función objetivo basada en recursos no renovables es posible minimizar el consumo de este tipo de recursos.

2.4.3. Objetivo basado en costes

El coste puede ser un factor que se desee incluir en la función objetivo. Algunas aproximaciones sobre el coste:

- Minimizar la suma de los costes de las actividades dependiendo éstos del instante de inicio de la tarea
- Minimizar el coste total del proyecto incluyendo costes por adelantos y retrasos, así como costes asociados a la duración de las tareas (acortar la duración de una tarea supone un coste adicional).
- Minimizar los costes relativos a la duración de las tareas cuando éstas están acotadas con una duración máxima y una duración mínima.
- Minimizar el coste del proyecto siendo éste proporcional a la duración del proyecto. Además las tareas pueden ser combinadas para reducir la duración del proyecto aunque con costes asociados. Esta función objetivo se emplea en problemas sin restricciones en los recursos.

2.4.4. Objetivo basado en planificación proactiva

Cuando un proyecto se lleva a cabo probablemente se produzcan retrasos que no podrán ser previstos durante la fase de planificación del proyecto. Por tanto, un director de proyecto estaría interesado en que la planificación fuera robusta para que cuando se produzcan dichos retrasos estos afecten lo menos posible a la ejecución del proyecto. Bajo este contexto surgen las denominadas funciones objetivo basadas en planificación proactiva.

Una opción consiste en minimizar el makespan y maximizar las holguras libres de las tareas. La holgura libre de una tarea es el intervalo de tiempo que puede retrasarse una tarea sin que afecte al resto de tareas. Existen otras propuestas en las que se maximiza la holgura libre mínima como único objetivo.

Otras aproximaciones añaden medidas de robustez a través de la ponderación de la holgura libre de una tarea con el número de sus sucesores inmediatos y/o la suma de sus solicitudes de recursos. La idea es evitar valores grandes de holguras libres ya que con holguras más pequeñas probablemente se obtenga un búfer suficiente.

2.4.5. Objetivo basado en planificación reactiva

Cuando un proyecto en ejecución sufre situaciones inesperadas, la programación realizada antes del inicio del proyecto deja de ser válida. En este punto, el problema a resolver se ha

visto modificado: algunas tareas estarán finalizadas, algunas estarán parcialmente realizadas, otras estarán en progreso, algunos recursos habrán cambiado su disponibilidad, etcétera.

En la planificación proactiva que se explicó anteriormente, el objetivo es adelantarse a los imprevistos. En el caso de la planificación reactiva ésta debe proporcionar una nueva planificación considerando los cambios que se han producido desde la planificación inicial.

2.4.6. Objetivo basado en el valor actual neto (NPV-RCPSP)

El Valor Actual Neto o VAN (Net Present Value, NPV) es un procedimiento que permite calcular el valor actual de un determinado número de flujos de caja futuros originados por una inversión.

Si durante la ejecución del proyecto se producen flujos de caja entrantes (debidos al pago por la finalización de tareas) o flujos salientes (debidos a la ejecución de tareas y al uso de recursos), entonces la función objetivo minimizará el VAN del proyecto sujeto a las restricciones RCPSP.

(Demeulemeester, 1997) desarrolla un algoritmo basado en el concepto de árbol temprano el cuál comprende todas las tareas planificadas en sus instantes más tempranos de finalización y que se corresponde con una solución factible con una duración de proyecto igual a la longitud del camino crítico.

2.5. Multi-objetivo

Si se combinan algunos de los diferentes objetivos comentados, puede obtenerse un problema multi-objetivo. En la práctica esto se lleva a cabo a través de una única función objetivo en la que se ponderan cada uno de los objetivos considerados.

2.6. Otros

Además de todas las variantes y extensiones que se han tratado en las secciones anteriores existen algunos subtipos que se muestran a continuación.

2.6.1. RCPSP estocástico

El RCPSP clásico es de tipo determinístico, esto es, la duración de cada tarea es conocida de antemano. En el RCPSP estocástico el tiempo de ejecución de cada tarea es una variable aleatoria que sigue una función de distribución. Este tipo de problemas es más realista aunque incrementa notablemente la complejidad. Si la función objetivo fuera minimizar la duración del proyecto, la solución que nos daría sería una duración estimada.

Normalmente es muy difícil poder obtener la función de distribución del proyecto ya que las tareas son interdependientes.

2.6.2. RCPSP relativos al problema bin-packing

El problema bin-packing consiste en, dado un conjunto de objetos, generalmente de forma rectangular, maximizar la cantidad de objetos que se pueden colocar en una superficie finita también rectangular (bin), minimizando el número de superficies necesarias.

La analogía entre el problema del bin-packing y el problema RCPSP consiste en que:

- La disponibilidad del recurso representa el tamaño del bin.
- El consumo de un recurso de una tarea sería el tamaño de un objeto.
- Cada periodo de tiempo se puede ver como un bin dentro del cual podemos empaquetar tareas.

- Minimizar la duración del proyecto representa minimizar el número de bins necesarios.

Es posible adaptar heurísticos desarrollados para el problema RCPSP para que resuelvan el problema del bin-packing simplemente con unas pequeñas modificaciones.

2.6.3. Multi-resource-constrained project scheduling problem (MRCPSP)

En el RCPSP clásico una tarea puede necesitar recursos de varios tipos, pero una tarea sólo requiere ejecutar una operación.

En el problema MRCPSP una tarea puede estar compuesta por un conjunto de operaciones o un conjunto de recursos sucesivos. Para una operación dada los recursos pueden ser empleados en paralelo o en serie. Este tipo de problemas también son denominados problemas de planificación de máquinas (máquinas en paralelo, Flow-shop, Job shop).

En (Pinedo, 1995) puede obtenerse más información sobre el problema de planificación de máquinas.

3. Técnica de Resolución

El RCPSP es un problema de tipo NP-duro por lo que la obtención de soluciones mediante el empleo de métodos exactos se vuelve inaplicable rápidamente en cuanto el tamaño del problema a resolver aumenta, siendo necesario recurrir a método heurísticos.

En el estudio de (Kolisch et al., 2006) se presenta un análisis sobre los heurísticos que vienen empleándose en los últimos tiempos como método de resolución del RCPSP. Dentro de los heurísticos, las técnicas metaheurísticas juegan un importante papel ya que se utilizan frecuentemente para la obtención de soluciones en un tiempo razonable. Entre los metaheurísticos que se pueden encontrar en la literatura están:

- Recocido Simulado (Simulated Annealing): (Valls et al., 2004), (Guan et al. 2009).
- Búsqueda tabú (Tabu Search): (Artigues et al., 2003), (Nonobe et al., 2002), (Lambrechts et al., 2007), (Deblaere et al., 2011).
- Algoritmos Genéticos (Genetic Algorithms): (Alcaraz et al., 2004), (Coelho et al, 2003), (Gonçalves et al., 2003), (Ashtiani et al, 2008), (Kuster et al, 2008).
- Algoritmos de Colonias de Hormigas (Ant Colony Optimization, ACO).

Este artículo se centra en la optimización con algoritmos de colonias de hormigas. El algoritmo ACO ha demostrado ser un medio efectivo para la resolución de problemas complejos de optimización combinatoria. Debe su nombre al símil de su funcionamiento con el de las colonias de hormigas. Se basa en la búsqueda de caminos en un grafo a través del conocimiento que va obteniendo cada agente (representado por cada hormiga) que va reforzando la utilización de los arcos (depositando feromona) que componen las mejores soluciones, es decir, las de menor coste.

Puede obtenerse una descripción detallada del problema en el libro de (Dorigo et al., 2004) y en el artículo de (Gambardella et al., 2004).

(Dorigo et al. 1991, 1992) fueron los primeros en emplear el sistema de colonia de hormigas para resolver el TSP (Travel Salesman Problem). Entre las aplicaciones de ACO se encuentra la resolución del problema de la planificación de jopshop (Pierucci et al., 1996), el problema de máquina única con retraso ponderado (Besten et al., 2000), problema de planificación de flowshop con secuencia dependiente de tiempos de set-up (Gajpal et al., 2004), problema de planificación de flowshop con permutaciones (Rajendran et al, 2004)

Entre los estudios que han empleado ACO para el problema RCPSP se encuentran (Herbots et al., 2004), (Chen et al., 2006), (Wang et al., 2007), (Khichane et al., 2009) y (Deng et al., 2010).

La implementación del algoritmo propuesto en este artículo está basado en el estudio de (Merkle et al., 2002) y sigue el siguiente esquema:

```
Mientras(!stop)
  Inicializar()
  Para cada hormiga
    ConstruirListaTareas()
    solución = GenerarPlanificación()
    Si (Coste(solución) < Coste(mejorSolución))
      SustituirMejorSolución(solución)
    FinSi
  FinPara
  ActualizarMejorSolucion()
  ActualizarFeromona()
FinMientras
```

Este enfoque consiste principalmente en tres fases: inicialización, construcción de la lista de tareas y actualizaciones.

- **Inicialización**

Durante esta etapa se inicializan todos los parámetros y estructuras de datos que se emplearán durante la ejecución del algoritmo. Se construye la tabla de feromonas de dimensiones $N \times N$ siendo N el número de tareas e inicializando cada celda con un valor positivo pequeño e igual a **0.5**. Se construyen **10** hormigas para **500** iteraciones lo cual supone la generación de **5000** planificaciones.

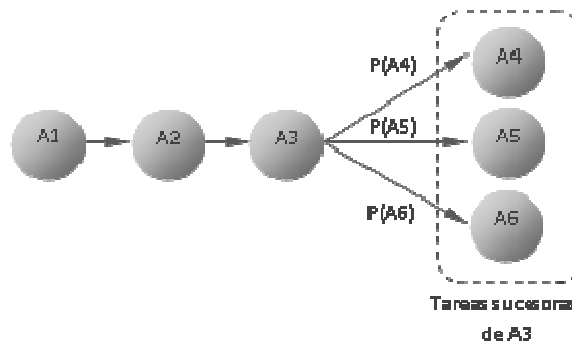
- **Construcción de la lista de tareas**

Un Esquema Generador de Planificaciones (Schedule Generation Scheme, SGS) es una técnica constructiva que planifica las tareas de acuerdo a una lista de prioridades dada, seleccionando las tareas una a una y determinando el instante de inicio de cada una de ellas. Una vez que un instante de inicio es asignado a una tarea, este instante permanece inalterable.

Existen dos tipos de SGS: el SGS en serie (S-SGS) y el SGS paralelo (P-SGS). El S-SGS está orientado a tarea e incrementa las tareas mientras construye la planificación. El P-SGS está orientado al tiempo e incrementa el tiempo mientras construye la planificación.

En este trabajo se ha escogido un S-SGS, es decir, orientado a tarea, de modo que cada hormiga construye su propia lista de tareas comenzando en la tarea ficticia **0** y terminando con la tarea, también ficticia **$n + 1$** .

Figura 3 Ejemplo de grafo de tareas



En cada paso la hormiga elige la siguiente tarea de entre las tareas que pueden ser sucesoras de ésta. Para ello para cada una de estas tareas candidatas se calcula la probabilidad de que sea seleccionada según la siguiente expresión basada en el coste que supone introducir en ese momento la tarea en la solución (planificación) y de la feromona:

$$P(n) = \frac{\tau(n-1, n)^\alpha \times \frac{1}{\text{Coste}(n)}^\beta}{\sum_{i=1}^N \tau(i-1, i)^\alpha \times \frac{1}{\text{Coste}(i)}^\beta} \quad (1)$$

Siendo:

- n : nodo (tarea) del que se quiere calcular la probabilidad.
- N : número total de nodos (tareas).
- $\tau(n-1, n)$: es el valor de la feromona entre el nodo del que se calcula la probabilidad y el último nodo que forma parte de la solución parcial actual.
- $\text{Coste}(n)$: coste de que el nodo n forme parte de la solución.
- α : peso que se da al valor de la feromona. Comienza valiendo 1.
- β : peso que se da a la componente del coste. Comienza valiendo 1.

Para asegurar la mayor exploración posible del espacio de búsqueda e intentando evitar que el algoritmo converja pronto, cada cierto número de iteraciones los valores de α y β varían para que la feromona pierda peso en el cálculo de probabilidades.

• Actualizaciones

Existen dos tipos de actualizaciones de información: la sustitución de la mejor solución y la modificación de la tabla de feromonas.

Tras el cálculo de una solución por parte de una hormiga debe comprobarse si el coste de la solución encontrada por la hormiga S_a es menor que el coste de la mejor solución

encontrada hasta el momento S_{best} . En caso afirmativo, debe procederse a la sustitución $S_{best} = S_{\alpha}$.

En este punto cabe realizar un inciso en relación al cálculo de soluciones. En la ejecución de las hormigas existen dos enfoques, la ejecución sincronizada y la ejecución no sincronizada de las hormigas. Si éstas son ejecutadas de forma síncrona cada hormiga debe esperar a que terminen de ejecutarse todas sus hermanas. Si se opta por el modo asíncrono cada hormiga continúa con la siguiente ejecución tras finalizar la actual sin realizar ningún tipo de espera. Lo habitual es emplear el funcionamiento sincronizado para evitar las dificultades derivadas de la ejecución asíncrona.

En cuanto a la modificación de la feromona, cuando todas las hormigas han calculado sus soluciones se procede a actualizar la feromona reforzando aquellas transiciones derivadas de la mejor solución encontrada en esta generación. Además si la mejor solución encontrada en esta generación no sustituyó a la mejor solución encontrada hasta el momento, entonces también se refuerzan las transiciones derivadas de ésta.

Además de realizar el refuerzo de feromona, se realiza la evaporación de la tabla de feromonas con un valor de $\sigma = 0.2$ que permite establecer un compromiso entre convergencia y exploración.

$$\tau(i, j) = \tau(i, j) \times \sigma \quad (2)$$

Si se fija un valor alto, las anteriores mejores planificaciones tienen menos peso en los futuros cálculos, con lo cual las hormigas tienen menos información del pasado para alcanzar la solución. Si se fija un valor excesivamente bajo, las soluciones nuevas no pueden competir en cuanto a feromona frente a las soluciones anteriores por tener valores de feromona muy altos, prácticamente sólo se refuerzan, no dejando opción a entrar a jugar nuevas opciones, lo que deriva en una convergencia rápida.

4. Caso de Estudio

Como consecuencia del carácter genérico del RCPSP clásico, su definición no se ajusta a la realidad de la planificación que se requiere en los proyectos del Mundo real, donde existen multitud de restricciones y características más allá de las consideradas por el RCPSP. En el apartado 2 se han mostrado algunas de las posibles extensiones o variantes de este problema y este artículo incorpora además una serie de limitaciones y características que consiguen acercar la definición del problema a las necesidades derivadas de una planificación más realista considerando aspectos importantes que no deben ser ignorados y que son descritos a continuación.

- **Relajación de restricciones**

Se considera la relajación de las restricciones que así se especifiquen, esto es, las restricciones indicadas no tienen porqué ser consideradas de forma inflexible sino que tienen un margen. Un ejemplo de restricción flexible es la fecha sugerida de finalización de proyecto.

- **Costes de penalización por retrasos en proyecto**

La penalización por retrasos en la realización de tareas es considerada frecuentemente en las funciones de costes de los problemas de tipo RCPSP. Este artículo considera además el coste por retraso a nivel global, de modo que es posible que resulte más interesante ampliar la duración del proyecto considerando los costes asociados que buscar una solución con el plazo inalterable.

- **Tiempo ocioso**

Cuando un recurso se paga por periodos y no por las horas reales de trabajo es común que aparezcan tiempos ociosos, es decir, periodos en los que no se hace uso del recurso: a pesar de estar disponible no se le asigna una tarea que deba realizar.

- **Costes variables**

La realidad de los proyectos en cuanto al coste de los recursos muestra que en muchos casos no resulta realista considerarlo constante. En este artículo se consideran dos tipos de costes variables: los debidos al instante en que se empleen los recursos y los debidos al lugar donde se usen éstos.

Un caso típico de coste variable son las horas extra realizadas por un empleado (coste variable en función del tiempo) o las dietas debidas a desplazamientos (coste variable debido a situación).

Se muestra a continuación el comportamiento del algoritmo con un ejemplo sencillo para facilitar la comprensión:

Inicio del proyecto: 01/03/2011

Se considera día laborable de lunes a viernes.

La duración del proyecto no puede superar los 15 días laborables

El proyecto debe acabar antes del 18/03/2011, penalizándose en 1.800€ cada día de retraso

Tabla 1: Relación de tareas, duraciones, precedencias y recursos del proyecto

Tarea	Duración	Predecesora	R1 = 19	R2 = 14	R3 = 6	Restricción
A1	6 d.		-	-	-	
A2	3 d.		-	-	-	
A3	4 d.		3	-	-	
A4	6 d.		10	-	6	Comenzar antes 14/03/2011
A5	4 d.	A2	-	-	-	
A6	4 d.	A5	-	10	-	
A7	8 d.		9	6	1	
A8	3 d.	A7	2	-	-	

Tabla 2: Relación de costes por recurso

Recurso	Costo estándar (€/h)	Costo extra (€/día)	Costo uso (€/día)
R1	10	14	50
R2	10	14	0
R3	10	14	30

La asignación de recursos inicial, sin tener en cuenta restricciones de capacidad y tan sólo atendiendo a las precedencias de las tareas se muestra en la Figura 4. Los 3 recursos R1, R2 y R3 se encuentran sobreasignados. Como se muestra en la Figura 5 la tarea 4 comienza antes del 14/03/2011 y la duración del proyecto es 11 días, acabando el 11/03/2011.

Figura 4 Gráfico inicial con recursos sobreasignados (R1, R2 y R3 respectivamente)

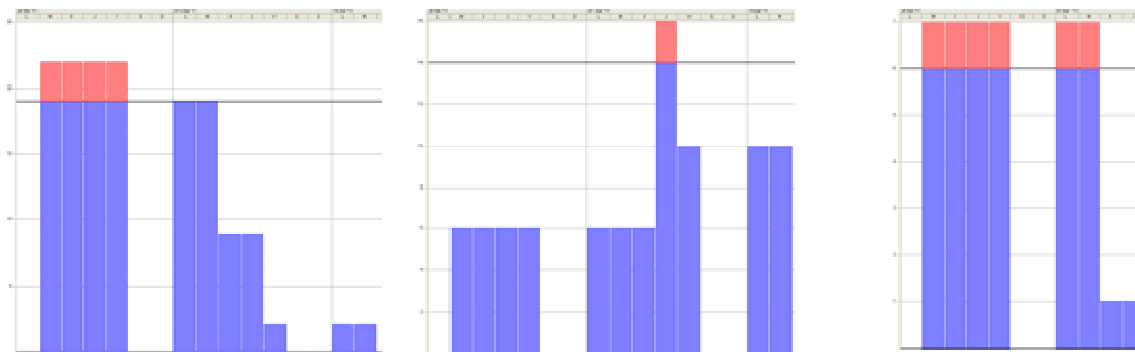
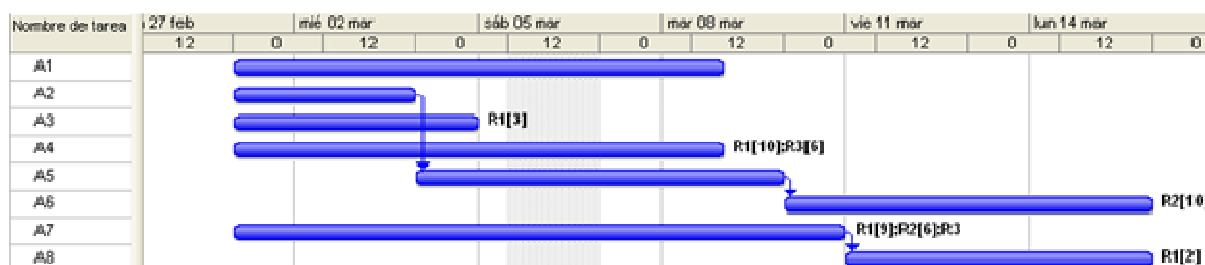
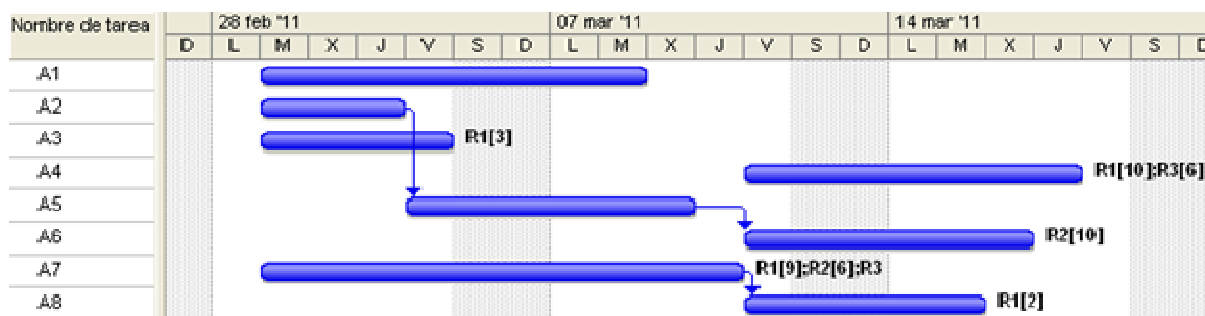


Figura 5 Gantt respetando únicamente precedencia entre tareas



La solución encontrada por el algoritmo basado en Colonias de Hormigas se representa en la Figura 6

Figura 6 Gantt solución de ACO

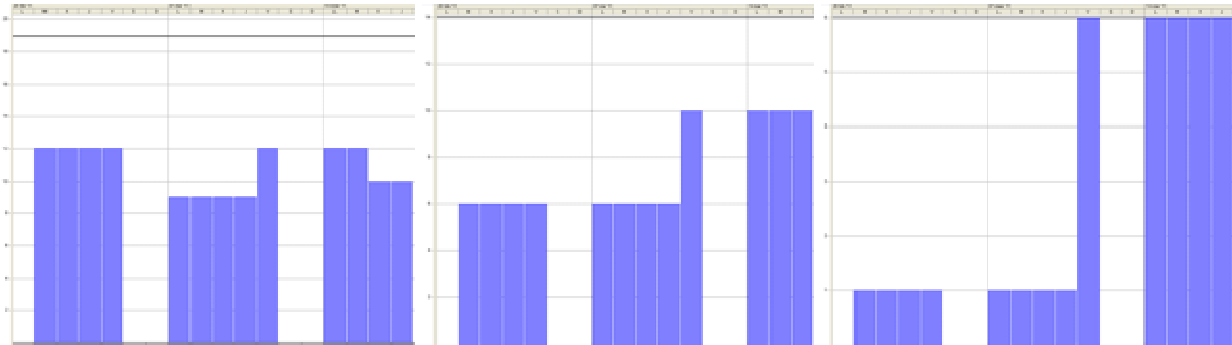


Los cambios introducidos respecto a la situación original son los siguientes:

- La tarea 4 desplaza su comienzo hasta el viernes 11/03/2011, respetando la restricción de comienzo antes del 14/03/2011
- Se asigna como día de trabajo el sábado 12/03/2011 a los recursos R1 y R3 en la tarea 4, pagándoles horas extras. El coste de hacer esto es $R1[10] \times 8h. \times 14€/h. + R3[6] \times 8h. \times 14€/h. = 1.792 €$. Se adelanta un día la fecha de finalización para respetar la restricción de acabar antes del 18/03/2011 porque pagar esas horas extra es más barato que los costes de la penalización.
- La tarea 6 comienza el 11/03/2011 retrasando un día.

La nueva situación de la asignación de recursos se muestra en la Figura 7, donde se observa que no hay ningún recurso sobreasignado.

Figura 7 Gráfico de recursos de la solución obtenida (R1, R2 y R3 respectivamente)



6. Conclusiones

- Ante un problema complejo de planificación que involucre un elevado número de tareas, recursos y restricciones se hace necesario recurrir a técnicas metaheurísticas para encontrar una solución buena y factible en un tiempo razonable.
- Se han encontrado varios estudios que abordan este problema mediante algoritmos genéticos, búsqueda tabú o recocido simulado, no ocurriendo así con el método de optimización por Colonia de Hormigas como se ha hecho en este trabajo.
- El algoritmo basado en Colonia de Hormigas se adapta muy bien a la vida real de un proyecto. En caso de que el director del proyecto quiera realizar nuevas planificaciones parciales durante la ejecución al realizar el seguimiento y control, es posible especificar las nuevas condiciones y el algoritmo convergerá más rápido debido a la tabla de feromonas guardada, sin ser necesario volver a explorar todos los caminos posibles. Esta fortaleza no está disponible en otras técnicas metaheurísticas como por ejemplo los algoritmos genéticos.
- El algoritmo aquí desarrollado podría incluirse como una de las técnicas disponibles en distintas metodologías de planificación de proyectos y nivelación de recursos.

7. Referencias

Alcaraz J., Maroto C. and Ruiz R., Improving the performance of genetic algorithms for the RCPS problem. Proceedings of the Ninth International Workshop on Project Management and Scheduling, pages 40–43, Nancy, 2004.

Artigues C., Michelon P., and Reusser S., Insertion techniques for static and dynamic resource-constrained project scheduling. European Journal of Operational Research, 149:249–267, 2003.

Ashtiani B., Leus R., Aryanezhad M-B., New competitive results for the stochastic resource-constrained project scheduling problem: exploring the benefits of pre-processing. 2008.

Barrios A., Ballestin F., and Valls V., A double genetic algorithm for the MRCPSP/max. Computers & Operations Research, 2009. Forthcoming.

- Besten, M. D., Sttzle, T. & Dorigo, M. (2000). Ant Colony Optimization for the Total Weighted Tardiness Problem. Berlin, Germany: Springer-Verlag, vol. 1917, Lecture Notes in Computer Science, pp. 611-620.
- Bianco L., Caramia M., and Dell'Olmo P.. Solving a preemptive project scheduling problem with coloring techniques. In Weglarz [193], pages 135–146. 1999.
- Brucker, P., Knust, S. Schoo, A. & Thiele, O. (1998) A branch and bound algorithm for the resource-constrained project scheduling problem. European Journal of Operational Research, 107, pp 272-288.
- Brucker P. and Knust S., Resource-constrained project scheduling and timetabling. Lecture Notes in Computer Science, 2079:277–293, 2001.
- Cavalcante C. C. B., De Souza C. C., Savelsbergh M. W. P., Wang Y., and Wolsey L. A.. Scheduling projects with labor constraints. Discrete Applied Mathematics, 112(1-3):27–52, 2001.
- Chassiakos A. P. and Sakellariopoulos S. P., Timecost optimization of construction projects with generalized activity constraints. Journal of Construction Engineering and Management, 131:1115–1124, 2005.
- Chen R-M and Lo S-T. Using an Enhanced Ant Colony System to Solve Resource-Constrained Project Scheduling Problem. IJCSNS International Journal of Computer Science and Network Security, VOL.6 No.11, November 2006
- Coelho J. and Tavares L., Comparative analysis of meta-heuristics for the resource constrained project scheduling problem. Technical report, Department of Civil Engineering, Instituto Superior Tecnico, Portugal, 2003.
- Davis, E.W. and Heidorn, G.E. (1971) An algorithm for optimal project scheduling under multiple resource constraints. Management Science 27 B803-B816.
- Debels D. and Vanhoucke M., The impact of various activity assumptions on the lead time and resource utilization of resource-constrained projects. Computers & Industrial Engineering, 54:140–154, 2008.
- Deblaere F., Demeulemeester E., Herroelen W., Reactive scheduling in the multi-mode RCPSP. Computers & Operations Research 38 (2011) 63–74. 2011.
- Demeulemeester, E., Reyck, B. D. and Herroelen, W. (1997), The discrete time/resource trade-off problem in project networks-A branch-and-bound approach, Research report 9717, Department of Applied Economics, K.U. Leuven.
- Deng L., Lin Y. and Chen M., Hybrid ant colony optimization for the resource-constrained project scheduling problem Journal of Systems Engineering and Electronics Vol. 21, No. 1, February 2010, pp.67–71.
- Dorigo M.. Optimization, Learning and Natural Algorithms (in Italian). PhD thesis, Dipartimento di Elettronica e Informazione, Politecnico di Milano, IT, 1992.
- Dorigo M., Maniezzo V. and Colomi A., Positive feedback as a search strategy. Technical Report 91-016, Dipartimento di Elettronica, Politecnico di Milano, IT, 1991.
- Dorigo M. and Stütz T., Ant Colony Optimization. June 2004.
- Drexel A., Nissen R., Patterson J.H., and Salewski F., Progen/px - an instance generator for resource-constrained project scheduling problems with partially renewable resources and further extensions. European Journal of Operational Research, 125(1):59–72, 2000.

- Drezet L.-E. and Billaut J.-C., A project scheduling problem with labour constraints and time-dependent activities requirements. *European Journal of Operational Research*, 112(1):217–225, 2008.
- Fulkerson, D.R. (1961) A network flow computation for project cost curves, *Management Science* 7, 167-178.
- Gajpal, Y., Rajendran, C., & Ziegler, H. (2004). An ant colony algorithm for scheduling in flowshops with sequence-dependent setup times of jobs. *European Journal of Operational Research*, Volume 155, Issue 2, Pages 426-438
- Gonçalves J. and Mendes J., A random key based genetic algorithm for the resource-constrained project scheduling problem. Technical report, Departamento de Engenharia, Universidade do Porto, 2003.
- Guan S., Nakamura M., Shikanai T., Okazaki T., Resource assignment and scheduling based on a two-phase metaheuristic for cropping system. *Computers and Electronics in Agriculture* 66 (2009) 181–190.
- Hartmann, S. (1997) Project scheduling with multiple modes: A genetic algorithm, *Mauskripte aus den Instituten für Betriebswirtschaftslehre der University Kiel*, No. 435.
- Hartmann S., and Briskorn D., A survey of variants and extensions of the resource-constrained project scheduling problem. *European Journal of Operational Research* Volume 207, Issue 1, 16 November 2010, Pages 1-14 2010.
- Harvey, R.T. and Patterson, J.H. (1979) An implicit enumeration algorithm for the time/cost tradeoff problem in project network analysis, *Found. Control Eng.* 4 , 107-117.
- Heilmann R.. A branch-and-bound procedure for the multi-mode resource-constrained project scheduling problem with minimum and maximum time lags. *European Journal of Operational Research*, 144:348–365, 2003.
- Herbots J., Herroelen W., Leus R.. Experimental Investigation of the Applicability of Ant Colony Optimization Algorithms for Project Scheduling. 2004.
- Hindelang, T.J. and Muth, J.F. (1997) A dynamic programming algorithm for decision CPM networks, *Operations Research* 27, 225-241
- Jarboui B., Damak N., Siarry P., and Rebai A., A combinatorial particle swarm optimization for solving multi-mode resource-constrained project scheduling problems. *Applied Mathematics and Computation*, 195:299–308, 2008.
- Kaplan, L. (1988) Resource-constrained project scheduling with preemption of jobs. Ph.D. Dissertation, University of Michigan, Ann Arbor, MI.
- Kelly, J.E. (1963) The critical path method: Resource planning and scheduling, in J.F. Muth, G.L. Thompson (Eds.), *Industrial Scheduling*, Prentice Hall, N.J., 347-365.
- Khichane M., Albert P. and Solnon C., An ACO-based Reactive Framework for Ant Colony Optimization: First Experiments on Constraint Satisfaction Problems. *Learning and Intelligent Optimization Springer-Verlag Berlin, Heidelberg* 2009
- Kolisch R., Hartmann S., Experimental Investigation of Heuristics for Resource-Constrained Project Scheduling: An Update. *European Journal of Operational Research* 174:23-37, 2006.
- Kuster J., Jannach D. and Friedrich G., Extending the RCPSP for Modeling and Solving Disruption Management Problems. *Applied Intelligence* Volume 31, Number 3, 234-253. 2008.
- Lambrechts O., Demeulemeester E. and Herroelen W., *Journal of Scheduling* Volume 11, Number 2, 121-136. 2007.

- Li H. and Womer K., Modeling the supply chain configuration problem with resource constraints. *International Journal of Project Management*, 26 (6):646–654, 2008.
- Maniezzo V., Gambardella L. M., De Luigi F., *Ant Colony Optimization. Optimization Techniques in Engineering*. Springer-Verlag. 2004.
- Merkle, D., Middendorf, M. & Schmeck, H.. Ant colony optimization for resource-constrained project scheduling. *IEEE Transactions on evolutionary computation*, 6(4), pp 333-346. 2002.
- Mika M., Waligora G., and Weglarz J., Tabu search for multi-mode resource-constrained project scheduling with schedule-dependent setup times. *European Journal of Operational Research*, 187 (3):1238–1250, 2008.
- Nonobe K. and Ibaraki T.. Formulation and tabu search algorithm for the resource constrained project scheduling problem. In C. C. Ribeiro and P. Hansen, editors, *Essays and Surveys in Metaheuristics*, pages 557–588. Kluwer Academic Publishers, 2002.
- Pierucci, P., Brandani, E. R., & Sogaro, A. (1996). An industrial application of an on-line data reconciliation and optimization problem. *Computers & Chemical Engineering*, Volume 20, Pages 1539-S1544
- Pinedo, M. (1995) *Scheduling theory, algorithms and systems*, ISBN 0-13-706757-7, Prentice Hall.
- Rajendran, C. & Ziegler, H. (2004). Ant-colony algorithms for permutation flowshop scheduling to minimize makespan/total flowtime of jobs. *European Journal of Operational Research*, Volume 155, Issue 2, Pages 426-438
- Rom W. O., Tukul O. I., and Muscatello J. R., MRP in a job shop environment using a resource constrained project scheduling model. *Omega*, 30(4): 275–286, 2002.
- Sabzehparvar M. and Seyed-Hosseini S. M., A mathematical model for the multi-mode resourceconstrained project scheduling problem with mode dependent time lags. *Journal of Supercomputing*, 44(3):257–273, 2008.
- Skutella, M. (1997) Approximation algorithms for the discrete time-cost tradeoff problem, in *Proceedings of the eighth Annual ACM-SIAM symposium on Discrete Algorithms*, New Orleans, LA, 501-508.
- Sprecher, A (2000) Scheduling resource-constrained projects competitively at modest resource requirements. *Management Science*, 46, pp 710-723.
- Sprecher, A. and Drexl, A. (1998) Solving multi-mode resource-constrained project scheduling by a simple, general and powerful sequencing algorithm, *European Journal of Operational Research* 107, 431-450.
- Tiwari V., Patterson J. H., and Mabert V. A.. Scheduling projects with heterogeneous resources to meet time and quality objectives. *European Journal of Operational Research*, 193(3):780–790, 2009.
- Valls V., Ballestin F., and Quintanilla M. S., Justification and RCPSP: A technique that pays. *European Journal of Operational Research*, 2004. Forthcoming. Proactive and reactive strategies for resource-constrained project scheduling with uncertain resource availabilities.
- Vanhoucke M., Work continuity constraints in project scheduling. *Journal of Construction Engineering and Management*, 132:14–25, 2006.
- Wang G., Gong W., De Renzi B. and Kastner R., Ant Colony Optimizations for Resource and Timing Constrained Operation Scheduling. *IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS*, VOL. 26, NO. 6, JUNE 2007

Yang B., Geunes J., O' Brien W.J.. Resource-Constrained Project Scheduling: Past Work and New Directions.. Research Report 2001-6, Department of Industrial and Systems Engineering, University of Florida. 2001.

Correspondencia (Para más información contacte con):

Área de Proyectos de Ingeniería – Universidad de Oviedo
Phone: +34 985 104 272
Fax: +34 985 104 256
E-mail: secre@api.uniovi.es
URL: <http://www.api.uniovi.es>