

Un modelo de programación de entornos multi-proyecto basado en la Metodología de la Cadena Crítica

Félix Villafañez¹, Javier Pajares¹, Adolfo López¹

¹ Gupo INSISOC. Escuela de Ingenierías Industriales. Universidad de Valladolid., villafafelix@yahoo.es, pajares@insisoc.org, adolfo@insisoc.org.

Resumen

En este trabajo, proponemos una metodología de programación de entornos multi-proyecto con restricciones en los recursos, que integra los procesos de selección y jerarquización de proyectos con su programación y asignación de recursos. La metodología combina conceptos de la Cadena Crítica de Goldratt con metodologías heurísticas utilizadas en la programación de proyectos software y procesos recursivos tipo backtracking. Las primeras simulaciones con diversos tipos de carteras de proyectos sugieren que la metodología iguala o mejora los resultados de muchos programas comerciales.

Palabras clave: Gestión de entornos multi-proyecto, dirección de carteras de proyectos, Project Management.

1. Introducción

La programación de un proyecto sujeto a restricción en los recursos (*resource constrained project scheduling problem*, RCPSP) ha sido un tema ampliamente tratado en literatura de dirección de proyectos. Sin embargo, en la práctica, las organizaciones deben gestionar varios proyectos ejecutándose simultáneamente, y que comparten los mismos recursos críticos. Consecuentemente, cobra cada vez mayor importancia el estudio del problema de la programación de carteras de proyectos sujetas a restricciones de recursos (*resource constrained multi-project scheduling problem*, RCMPSP), de mayor complejidad que RCPSP, al ser necesario tener en cuenta, no solo las relaciones existentes entre actividades de un mismo proyecto, sino estudiar las relaciones y compatibilidades que se dan entre distintos proyectos. Ejemplos de ello son, entre otros, los trabajos de Hans *et al* (2007), Kruger *et al.* (2009) o Kum-Khiong Yang y Chee-Chuong Sum (1993).

Las soluciones propuestas dependen en gran medida de la formulación del problema. Los modelos jerárquicos reparten los recursos entre los diferentes proyectos en función de su importancia para la organización, y posteriormente, cada proyecto se programa contando con los recursos asignados. En el otro extremo, se encuentran soluciones que resuelven un “macro-proyecto” formado por todas las actividades de todos los proyectos de la cartera.

En cualquiera de los casos, las soluciones propuestas tienen dificultades para ser implementadas en entornos reales, dinámicos y cambiantes (sobre-costes, retrasos, cambios de alcance), y en los que la típica dinámica empresarial hace que continuamente nuevos proyectos entren a formar parte de la cartera, siendo necesario cambiar prioridades, reorganizar las tareas y asignar recursos nuevamente. Esto provoca que, como sugieren Pajares *et al.*(2009), la decisión de aceptar un nuevo proyecto, no solo dependa de su valor intrínseco, sino de cómo éste interacciona con los recursos y con la programación de los proyectos existentes.

En este trabajo proponemos una nueva metodología que utiliza las ventajas de la Cadena Crítica de Goldratt (1997) y que integra la selección y jerarquización de proyectos con su programación y asignación de recursos. Primeramente, partimos de las programaciones individuales de cada uno de los proyectos, a los que se aplica la metodología de la Cadena Crítica (CC). Posteriormente, se estudian posibles carteras formadas por proyectos que maximicen una función de utilidad de la empresa (por ejemplo, el valor, o cualquier otra medida que integre aspectos económicos y estratégicos de la empresa). Finalmente, utilizamos un algoritmo de programación que aplica heurísticas de resolución de problemas informáticos, optando por un proceso recursivo que emplea una combinación de los métodos “Divide y vencerás” y “*Backtracking*”. Los resultados obtenidos mejoran los resultados de muchos programas comerciales de gestión multi-proyecto, sobretodo cuando las restricciones de los recursos son muy importantes.

El resto de este artículo se organiza de la siguiente manera. En el apartado dos, explicaremos el modelo propuesto. Posteriormente, ilustraremos su funcionamiento mediante un ejemplo. Finalizaremos con las conclusiones más importantes.

2. MDP/CC. Most Decisive Portfolio/Critical Chain

El cimiento principal de la metodología que aquí estamos presentando es la integración del proceso de selección de proyectos con la programación de los mismos. La razón por la que se trata de integrar estos dos aspectos del Project Portfolio Management en esta metodología, es precisamente estudiar las relaciones y compatibilidades existentes entre los proyectos que van a formar parte de una misma cartera de proyectos, para conseguir de esta manera la mejor cartera posible para la organización.

Por otro lado, la integración del método de la Cadena Crítica (CC) en la programación de una cartera de proyectos, es el segundo punto clave de esta metodología. La razón por la que la programación de la cartera de proyectos se basa en la CC, es que cabe esperar que los buenos resultados obtenidos por esta metodología en la planificación de un único proyecto se trasladen a la planificación de varios proyectos, obteniendo sinergias entre los proyectos, de tal manera que la consecución de los objetivos estratégicos de la organización resulte más sencilla. La aplicación del método de la CC a cada proyecto de manera individualizada, será el primer proceso por el que pasará todo proyecto que se esté estudiando para su incorporación como activo a la organización.

Todo esto se apoya en una filosofía de trabajo, que no es otra que el trabajo en equipo. Este es un aspecto crucial, pues si en un único proyecto se encuentran involucrados agentes tan diversos como directivos, sponsors, project managers, recursos humanos, proveedores, stake holders, etc. En una cartera de proyectos, el número de agentes que participan e interaccionan es enorme, por lo que sin una cooperación total, ligando en un único objetivo el bien global y el individual, resulta casi imposible el logro de los objetivos estratégicos de la organización.

El hecho de que MDP/CC esté basado en el trabajo en equipo implica un **cambio estructural** en la organización. Todos los recursos de los que se disponen, son gestionados de manera global, por lo que se dispondrá de un *pool* de recursos. En ningún momento y bajo ningún concepto será asignado ningún recurso de manera indefinida a ningún proyecto, por importante que sea. Por lo tanto, todos los recursos serán asignados por la organización a un proyecto durante un espacio de tiempo. Este espacio de tiempo es el que dura la actividad por la cual han sido reclamados esos recursos. Una vez que el trabajo está completado, los recursos vuelven al *pool* de la organización. De esta manera se consigue aumentar la eficiencia de los recursos.

2.1. Puntuación de Proyectos

Para realizar la puntuación de los proyectos, MDP/CC utiliza dos métodos que se pueden combinar según la importancia que se quiera dar a cada uno en cualquier proporción.

El primero es un método de ordenación, y por lo tanto, la puntuación que obtiene cada proyecto es debida a sus características y a las características del grupo de proyectos con el que se está comparando. Por lo tanto, las diferencias de puntuación entre los diferentes proyectos dependerán de lo homogéneo o no que sea la cartera considerada; a mayor homogeneidad menor diferencia entre las puntuaciones.

El segundo de los métodos que se utiliza es un método de calificación. Su modo de funcionamiento se asemeja al de un examen. La organización establece unos requisitos mínimos y máximos para cada atributo de un proyecto, en función de sus preferencias y objetivos. Cada proyecto es analizado atributo a atributo. Por cada atributo recibe una puntuación, pudiendo ser esta positiva o negativa, en función de si supera el requisito mínimo requerido para dicho atributo. La puntuación positiva que puede recibir por cada atributo está limitada, a partir de un cierto valor, requisito máximo, no se obtiene mayor puntuación.

Los atributos que se analizan para cada proyecto son seis, entre los que se encuentra el VAN, la oportunidad que representa para la organización ese proyecto, o el riesgo que supondría su realización. A cada uno de estos atributos de le puede dar mayor o menor peso en función de los objetivos de la organización.

El hecho de disponer de dos métodos de puntuación tan diferentes, hace que la organización, dependiendo de los proyectos considerados, pueda optar en mayor medida por uno u otro para cerciorarse de que prioriza los proyectos adecuadamente para lograr sus objetivos.

2.2. Formación y Programación del Portfolio

En este apartado nos vamos a centrar en la manera en que se forma y programa el portfolio. Para ello, partimos de una situación en la que se dispone de los proyectos individuales (inputs de la metodología) ya programados aplicando la metodología de la Cadena Crítica y puntuados (primer y segundo proceso de la metodología). Además se dispone de las restricciones de recursos de la organización.

El primer paso a dar es formar los portfolios candidatos. Para ello se construye una lista de los portfolios candidatos considerando todas las posibles combinaciones de proyectos que podrían formar parte de la cartera. Sobre esta primera lista se eliminarán aquellos no viables por razones de valor del portfolio, consumo de recursos, o diversificación del riesgo. Una vez realizado este paso, se ordenarán de mayor a menor valor los portfolios candidatos reales. El valor de un portfolio es considerado como la suma aritmética de los valores/puntuaciones de los proyectos que lo forman.

A partir de este momento, se tratará de encontrar una programación válida (realización exitosa de cada proyecto) para el portfolio de mayor puntuación, primer candidato a MDP. En caso de que este no sea factible, se procederá del mismo modo con el siguiente de la lista, y así sucesivamente, hasta encontrar un portfolio válido, que será el portfolio de más valor para la organización y que se puede realizar en base a su capacidad.

Como consecuencia de este proceso, ningún proyecto, por importante que sea, tiene asegurada su presencia en el portfolio final. Esto es debido a que se selecciona el portfolio que proporciona más valor a la organización, no proyectos individuales, por lo que cobra especial importancia las relaciones y compatibilidades entre los proyectos.

La programación de una cartera de proyectos es un problema NP-hard, es decir, que las posibles combinaciones de actividades tienden a infinito a medida que se introducen más proyectos, por lo que se hace inabordable la exploración de todas las posibilidades a la hora de examinar la viabilidad o no de un portfolio. Para reducir significativamente las posibilidades a explorar, se ha optado por utilizar el método de resolución de problemas informáticos “*Divide y vencerás*”.

Como consecuencia de aplicar “*Divide y vencerás*”, el algoritmo de programación de un portfolio consta de dos fases principales: en la primera, se efectúa la programación de las actividades críticas, aquellas que utilizan recursos críticos (recursos cuello de botella, con mayores restricciones); en la segunda, se realiza la programación de las actividades no críticas (actividades que no utilizan recursos críticos) que no hayan sido fijadas en el paso anterior.

Para programar las actividades críticas, se utiliza un método de priorización de actividades, basado en holguras de tiempo, sumado a “*Backtracking*”. Como resultado, se realiza la programación de las actividades de manera recursiva, generando soluciones parciales.

A medida que se avanza en la programación de las actividades, se analiza la viabilidad de la solución parcial; si se localiza una incompatibilidad irresoluble, el método vuelve sobre sus pasos, deshaciendo el último movimiento, explorando la siguiente posibilidad en busca de una solución completa, asegurando que se hace todo el trabajo que se tiene que hacer, en tiempo. De esta manera, se exploran todas las posibilidades, como si se recorriese un árbol binario (en este caso no binario), asegurándonos de encontrar una solución satisfactoria, en caso de que exista.

Mediante este proceso se estudia la compatibilidad entre proyectos, logrando el mayor rendimiento posible de los recursos de los que se dispone. Si se logra con éxito esta primera fase, o lo que es lo mismo, se ha encontrado una programación válida para las actividades críticas respetando las restricciones de capacidad de nuestros recursos cuello de botella, se pasa a la segunda parte de la programación de actividades. En caso contrario, el portfolio que está siendo estudiado quedaría descartado, y se estudiaría el siguiente candidato.

La programación de las actividades no críticas se realiza de una manera parecida. La gran diferencia, con respecto a la programación de las actividades críticas, es que en este proceso se parte de un primer scheduling, que es analizado en base a los recursos no críticos. En caso de haber incompatibilidades, se utiliza un algoritmo basado en “*Backtracking*” sumado a “*Divide y vencerás*”, gracias a lo cual podemos explorar las diversas posibilidades de programación de actividades, buscando una solución válida en un espacio más reducido de posibilidades, de manera local y focalizada en la incompatibilidad.

2.3 Pseudocódigo

Para generar una idea global más clara de la metodología MDP/CC, se va a mostrar en pseudocódigo todas las etapas a seguir. En un primer paso, vamos a ver cuál sería el cuerpo del algoritmo MDP/CC:

```

{
  aplicar CC a todos los proyectos
  puntuar proyectos
  formar candidatos a MDP
  WHILE (programacion == false AND candidatos>0)
  {
    seleccionar_candidato_a_MDP()
    IF (ejecuta_MDP())
    {
      programacion = true
    }
    ELSE
    {
      deshacer_programacion()
      eliminar_candidato()
    }
  }
}

```

Dentro de este algoritmo, nos podemos fijar en una función fundamental, que es `ejecuta_MDP()`, donde se puede observar la aplicación del método “*Divide y vencerás*” con claridad.

```

{
  fija_buffers_proyecto()
  IF (fija_actividades_criticas())
  {
    IF (fija_actividades_no_criticas())
      result=true
  }
}

```

A su vez, en este nuevo pseudocódigo, nos encontramos con una nueva función `fija_actividades_criticas()`, que es una de las partes básicas del algoritmo, en donde se puede observar la aplicación de “*Backtracking*”. El pseudocódigo de la función `fija_actividades_no_criticas()` no se va a mostrar, puesto que sería redundante, debido a su semejanza con la función mostrada, pero focalizando la programación en las incongruencias debido al consumo de recursos no críticos.

```

calcular opciones
WHILE (criticas_fijadas()==false AND opciones>0)
{
  seleccionar actividad con mayor prioridad
  IF(mover_actividad)
  {
    IF(fija_actividades_criticas()==false)
    {
      deshacer_movimiento
      eliminar opcion
    }
  }
  ELSE
  eliminar opcion
}
RETURN criticas_fijadas

```

3. Ejemplo de programación multiproyecto.

Como muestra de los resultados que se pueden obtener aplicando esta metodología, se muestra la situación inicial de la que se partiría en una organización que está estudiando siete proyectos para su incorporación (Figura 1). Cada proyecto está definido por sus actividades (duración y consumo de recursos), relaciones de precedencias entre las actividades y por una medida de su valor o importancia para la organización. Dicha organización, dispone para la realización de los proyectos, de 18 recursos genéricos y 2 recursos críticos.

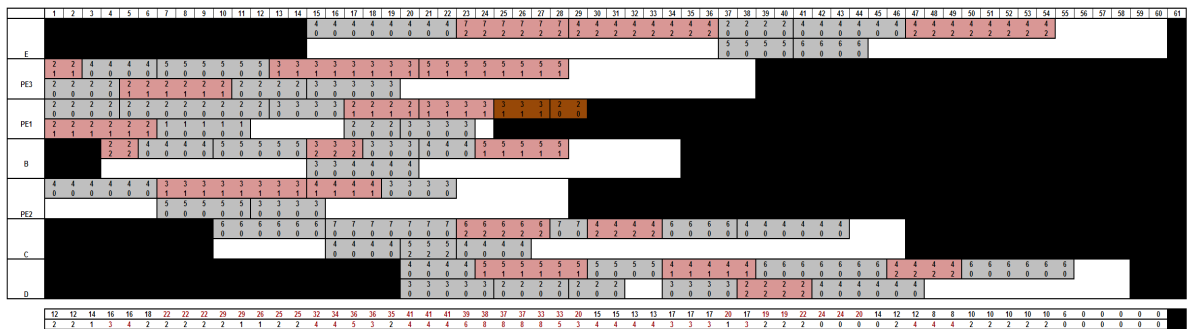


Figura 1. Situación Inicial: 7 proyectos sin aplicar CC.

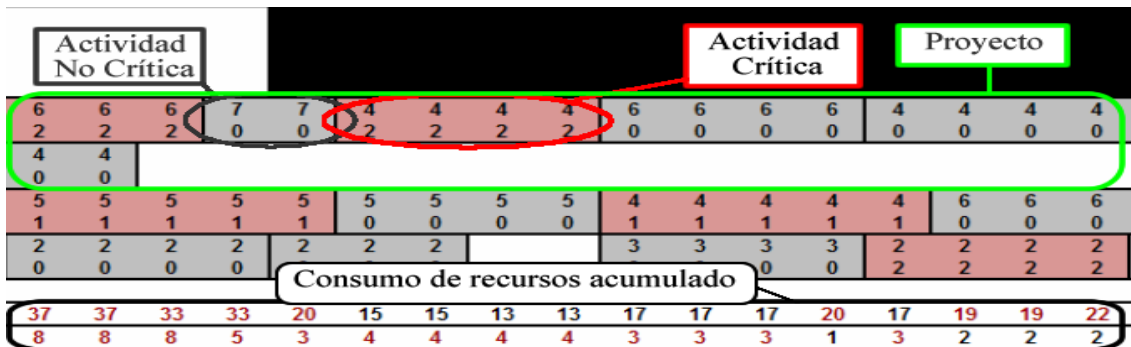


Figura 2. Zoom sobre Situación Inicial

En la figura 2, se puede observar con mayor precisión como se ha representado gráficamente las actividades críticas, las actividades no críticas y sus consumos de recursos. Además se puede ver el consumo de recursos en el horizonte temporal que se está representando, siendo demandado es muy superior al disponible.

Tras aplicar CC a todos los proyectos y ejecutar el algoritmo de programación de actividades, con los recursos disponibles, no es posible realizar los 7 proyectos (primer candidato a MDP), pero sí es posible abordar el segundo candidato. En la Figura 3 representamos el resultado de la programación de este segundo candidato a MDP, que consta de seis proyectos. Ahora como resultado de aplicar CC, tenemos nuevas actividades, que son los *buffers* y la duración de las actividades se ha visto reducida.

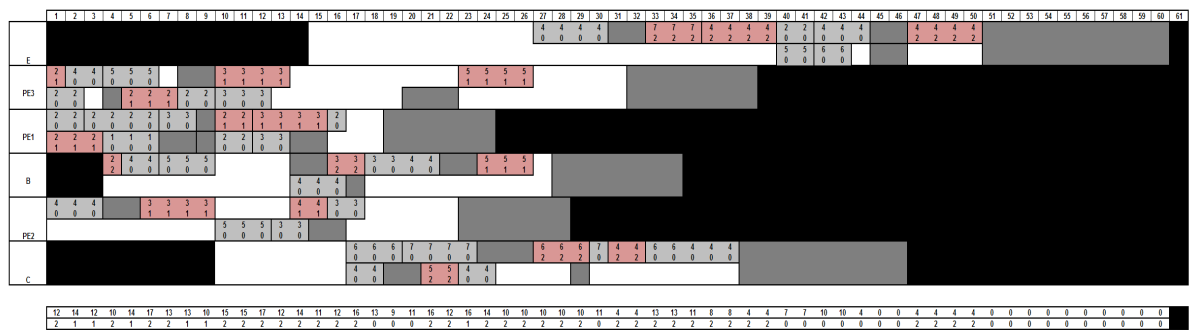


Figura 3. Situación Final: 6 proyectos con CC.

38	39	40	41	42	43	44	45	46	47	48	49	50	51	52
4	4	2	2	4	4	4	Buffer	4	4	4	4	Buffer		
2	2	0	0	0	0	0	Buffer	2	2	2	2	Buffer		
		5	5	6	6		Buffer							
		0	0	0	0									

Figura 4. Introducción de Buffers y reducción de tiempo-coste.

La Figura 4, muestra parte del resultado de la aplicación del método de la cadena crítica, con una reducción del tiempo y recursos consumidos en las actividades y la introducción de buffers, que protegen al proyecto de la incertidumbre.

En los siguientes gráficos podemos ver la evolución del consumo de recursos genéricos y críticos para el portfolio seleccionado, pasando de una situación inicial, en la que están todos los proyectos considerados sin aplicar CC (punto de partida de la metodología), con un consumo que llega a triplicar la capacidad de la organización, a una situación final en la que se respetan las limitaciones de la organización y se coopera entre los proyectos para lograr el máximo beneficio para la misma.

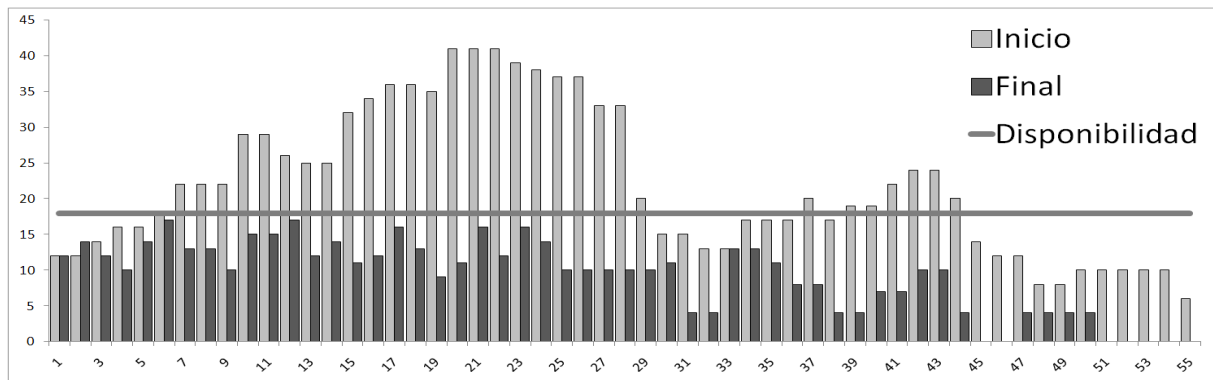


Figura 5. Comparación consumo de recursos genéricos: Inicial-Final.

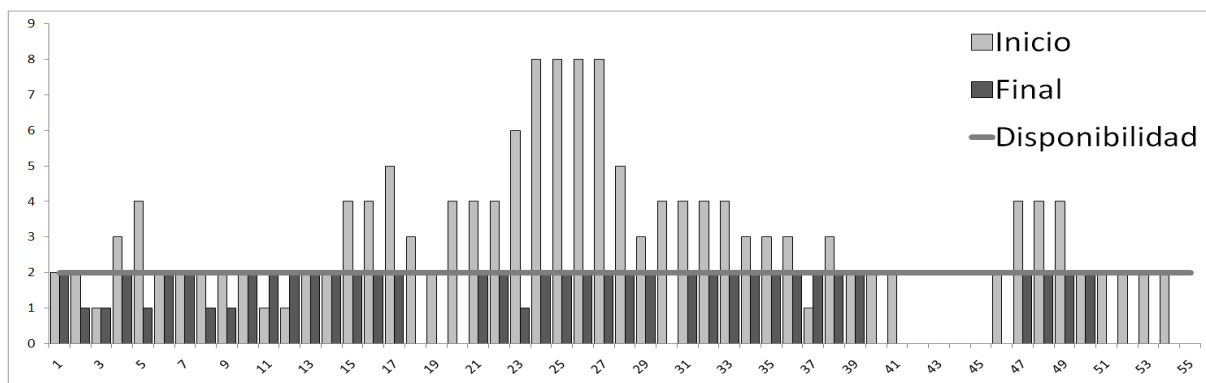


Figura 6. Comparación consumo de recursos críticos: Inicial-Final.

Hemos comparado la metodología propuesta con los resultados proporcionados, por ejemplo, con Microsoft Project. En la tabla 1, comparamos los resultados para el ejemplo anterior. Tal y como se puede observar, la mejora obtenida es muy elevada, ante la sospecha de que gran parte de las mejoras sean debidas a la programación previa de los proyectos individuales mediante CC y no a la utilización del algoritmo (*Backtracking + Divide y vencerás*), hemos tratado de separar ambas causas. En la tabla 2, comparamos nuestros resultados con los de Project, siendo el *input* en ambos casos los proyectos programados mediante CC. Observamos que también obtenemos mejoras significativas.

Tabla 1. Comparación resultados debido a metodología.

Comparación de resultados debido a metodología		
Datos Analizados	MDP/CC	MP
Proyectos potenciales	7	7
Proyectos de la cartera	6	3
Valor Cartera potencial	3,16	3,16
Valor Cartera	2,84	1,55
% Valor Conseguido	89,87%	49,05%
VAN Cartera	1660	1010
% VAN Conseguido	83,84%	51,01%

Tabla 2. Comparación resultados debido a algoritmo de ordenación.

Comparación de resultados debido a algoritmo		
Datos Analizados	MDP/CC	MP
Proyectos potenciales	7	7
Proyectos de la cartera	6	5
Valor Cartera potencial	3,16	3,16
Valor Cartera	2,84	2,45
% Valor Conseguido	89,87%	77,53%
VAN Cartera	1660	1320
% VAN Conseguido	83,84%	66,66%

4. Conclusiones.

En este artículo hemos presentado una metodología de programación multiproyecto que integra los procesos de programación y selección de proyectos de las carteras. La metodología, incorpora elementos de la Teoría de las Restricciones y la Cadena Crítica de Goldratt, así como algoritmos de búsqueda de soluciones, como Backtracking y Divide y vencerás.

Esta forma de actuar permite una utilización eficiente del “pull de recursos” de los que dispone la organización, de forma que se garantice la realización del conjunto de proyectos que proporcione mayor valor a la empresa.

La metodología propuesta mejora los resultados obtenidos por los paquetes software habituales, al conseguir programar mayor cantidad de proyectos para una determinada cantidad de recursos disponibles.

Agradecimientos

Los autores agradecen la financiación de la Junta de Castilla y León, proyectos VA006A009 y GREX251-2009, y al Ministerio de Educación y Ciencia, proyecto TIN2008-06464-C03-02;

Referencias

- Anavi-Isakow, S. Golany, B. “Managing multi-project environments through constant work-in-process”. *International Journal of Project Management* 21 (2003) 9–18
- Artto, K. and Dietrich, P. “The Wiley Guide to Project, Program & Portfolio Management Chapter one: STRATEGIC BUSINESS MANAGEMENT THROUGH MULTIPLE PROJECTS“. (2007), 1—33.
- Blichfeldt, B and Eskerod, P. “Project portfolio management – There’s more to it than what management enacts”. *International Journal of Project Management* 26 (2008) 357–365
- Carbno, C. “Optimal resource allocation for projects”. *Project Management Journal*; (1999) 22-32
- Casey, Robret J. “An Innovative Approach to Schedule Management on the F/A-22 Major Defense Acquisition Program (MDAP): Demonstration of Critical Chain Project Management”. (2005).
- Goldratt, E.M. *Critical chain*, The North River Press, Great Barrington, MA (1997)
- Hans, E.W. Herroelen, W. Leus, R. and Wullink, G. “A hierarchical approach to multi-project planning under uncertainty”. *Omega* 35 (2007) 563 – 577
- Herroelen, W. “Project Scheduling—Theory and Practice”. *Production and Operations Management Society* (2005)
- Krüger, D and Scholl, A. “A heuristic solution framework for the resource constrained (multi)project scheduling problem with sequence-dependent transfer times”. *European Journal of Operational Research* 197 (2009) 492–508
- Kum-Khiong Yang y Chee-Chuong Sum. “A comparison of resource allocation and activity scheduling rules in a dynamic multi-project environment“. *Journal of Operations Management* 11 (1993) 207-218.

Pajares, J., López, A., Arauzo, J.A. y Hernández, C. “Project Portfolio Management, selection and scheduling. Bridging the gap between strategy and operations”. *XIII Congreso de Ingeniería de Organización*. Barcelona-Terrassa (2009).