

01-029

A COMPARISON OF ALGORITHMS FOR CONSTRUCTING PERT GRAPHS FOR LARGE PROJECTS

Arauzo-Azofra, Antonio; Salas-Morera, Lorenzo; García-Hernández, Laura; Perez-Caballero, Alberto
Universidad de Córdoba

The well known PERT method for project planning uses AOA (Activity On Arc) graphs. Preparing these graphs requires the inclusion of dummy activities. Ideally, the smaller the number of dummy activities, the smaller the time of computation taken on following calculations, and the greater the understandability. However, the preparation of an optimal graph (one with the lowest number of dummy activities) is a high computational complexity problem (NP class). In order to work with large projects, there are diverse algorithms that reduce the computation time by allowing the inclusion of more dummy activities than necessary. Including recent algorithms, we make a comparison focused on performance: the number of dummy activities included in relation to the computational cost.

Keywords: *Pert; AOA graphs; Project planning; Algorithmic complexity*

COMPARATIVA DE ALGORITMOS PARA LA CONSTRUCCIÓN DE GRAFOS PERT DE PROYECTOS GRANDES

Tanto el método PERT, como otras técnicas, bien conocidos en programación de proyectos, utilizan grafos de tipo AOA (Activity On Arc). La construcción de estos grafos puede requerir la inclusión de actividades ficticias. Idealmente, cuanto más pequeño sea el número de actividades ficticias, menor será el tiempo de cálculo a emplear en fases posteriores y mayor claridad tendrá la representación del proyecto. Sin embargo, la construcción de un grafo óptimo, con el menor número de actividades ficticias, es un problema computacionalmente muy complejo (de clase NP-completo). Para trabajar con proyectos muy grandes, hay diversos algoritmos propuestos. Algunos consiguen una menor complejidad de cálculo sacrificando la reducción de actividades ficticias. Incluyendo los algoritmos más recientes, realizamos una comparativa enfocada en su rendimiento: número de actividades ficticias introducidas frente a su coste computacional.

Palabras clave: *Pert; Grafos AOA; Programación de Proyectos; Planificación; Complejidad algorítmica*

Correspondencia: Antonio Arauzo-Azofra arauzo(arroba)uco.es

Agradecimientos: Agradecer a los revisores sus comentarios y críticas constructivas que han ayudado mucho a mejorar esta contribución.

1. Introducción

Dentro de las técnicas descritas por el *Project Management Body of Knowledge* (PMBok) del *Project Management Institute* (2013) para realizar la programación de un proyecto, hay varias, como el método PERT o el método de la ruta crítica (CPM), que utilizan grafos tipo AOA (*Activity On Arc*). Estos grafos sirven para representar la programación de un proyecto interpretando los arcos como tareas y los nodos como sucesos del proyecto.

Frente a los grafos tipo AON (*Activity On Node*), los grafos AOA, pueden presentar una estructura más sencilla y ordenada del proyecto, porque organizan prelaciónes comunes a varias actividades en sucesos. Sin embargo, necesitan la inclusión de actividades ficticias para crear esta organización. Idealmente, cuanto menor sea el número de actividades ficticias, más sencillo será el grafo, menor será el tiempo de cálculo a emplear en fases posteriores y mayor claridad tendrá la representación del proyecto.

Lejos de ser trivial como la de los grafos AON, la construcción de los grafos AOA, si se quiere obtener el grafo óptimo (aquél con el mínimo número de actividades ficticias), es un problema NP-completo (Krishnamoorthy, 1979). En los problemas NP-completos, el tiempo necesario para encontrar su solución crece exponencialmente con el tamaño de la entrada. Esto no supone un problema para los proyectos pequeños pero sí afecta a los proyectos muy grandes, y a la posibilidad de modelar interacciones entre muchos proyectos.

Para evitar este alto coste computacional, hay diversos algoritmos propuestos. Son algoritmos aproximados, que reducen el tiempo necesario a costa de no reducir al mínimo el número de actividades ficticias. Estos algoritmos utilizan estrategias diferentes y es de esperar que obtengan resultados diferentes. Sin embargo, para nuestro conocimiento, no se ha publicado ninguna comparativa empírica sobre ellos.

2. Algoritmos de generación de grafos AOA

En esta contribución, se plantea la evaluación del rendimiento de los algoritmos aproximados para la construcción de grafos AOA y compararlos. Se han elegido tres algoritmos que utilizan estrategias diferentes para la construcción del grafo.

2.1 Algoritmo de Sharma

Este algoritmo (Sharma, 1998) utiliza una estrategia de dos fases. En la primera, construye sistemáticamente y de una forma sencilla, que garantiza eficiencia polinomial, un grafo con muchas actividades ficticias. En la segunda, realiza una búsqueda eliminando actividades ficticias innecesarias. La eficiencia del algoritmo completo (ambas fases) está acotada en el peor caso dentro de $O(n^3)$. Esto quiere decir que su tiempo de ejecución vendrá determinado por el tamaño del problema (n = número de prelaciónes entre actividades) con alguna función polinomial cuyo grado es 3 como máximo.

Se ha incluido que tolere varias actividades de inicio y fin y la posibilidad de que haya actividades paralelas entre ellas.

2.2 Algoritmo de Gento-Municio

El algoritmo propuesto por Gento-Municio (2004) se basa en la búsqueda de patrones de relaciones entre actividades e ir construyendo el grafo entorno a ellos.

La interpretación de la propuesta ha requerido algunas suposiciones. En el paso 8, al contrario de lo que especifica la propuesta, ya se han asignado nodos finales a todas las

actividades en el paso 4, como indica la figura 8 de (Gento-Municio, 2004). En cualquier caso, esos nodos no están conectados, por tanto, suponemos que lo que se pretende aquí es que se reemplacen si es necesario. No haber asignado los nodos en el paso 4 impediría realizar correctamente el paso 7.

Por otra parte, el paso 9 es innecesario pues, según la interpretación anterior de los pasos 4 y 8, todos los nodos finales ya han sido asignados.

La explicación completa del algoritmo requeriría un espacio excesivo. Puede consultarse la explicación en la propuesta original y el código fuente de nuestra implementación (disponible como se indica en la sección 3), que incluye comentarios indicando las suposiciones comentadas anteriormente.

2.3 Algoritmo de Cohen & Sadeh

Cohen y Sadeh (2006) proponen un algoritmo que crea un multi-grafo (estructura que permite que entre dos nodos pueda haber varios arcos) basándose en la construcción de una tabla con información sobre las prelación del proyecto.

Se ha implementado la transformación del multi-grafo en un grafo AOA para permitir trabajar con actividades paralelas en igualdad de condiciones con el resto de los algoritmos.

3. Estudio empírico

3.1 Metodología

Los tres algoritmos han sido implementados en el lenguaje de programación Python y han sido incluidos en el software PPC-project (Salas, 2013). Están disponibles en el repositorio del proyecto. Esto incluye todo su código fuente, para poder ser estudiados y garantizar la reproducibilidad de este trabajo. Todas las implementaciones han tratado de ser lo más fieles posible a la propuesta original. No obstante, siempre que una estructura de datos tipo diccionario o conjunto era más eficiente, la hemos utilizado.

La ejecución de los algoritmos se ha realizado en un servidor de cálculo con dos procesadores Intel(R) Xeon(R) E5420 a 2.50GHz con 8Gb RAM. La ejecución ha sido secuencial y el servidor estaba dedicada únicamente a la ejecución de los algoritmos para evitar fallos de caché y otras posibles interferencias. Se ha medido el tiempo de ejecución exclusivamente de los algoritmos. Sólo incluye la transformación de la tabla de prelación en la estructura de datos del grafo AOA mediante el algoritmo. Queda fuera el dibujo del grafo en formato gráfico y cualquier otro cálculo de prueba que se realice para la validación del grafo.

Los datos de los proyectos usados para realizar la evaluación han sido tomados de la librería de problemas *PSPLib* (Kolisch, 1997). Se trata de una recopilación de datos de proyectos de diversas artículos publicados. Estos datos se usan como base de medida para resolver diversos problemas de optimización sobre ellos como la programación de proyectos para asignación o nivelación de recursos con diferentes limitaciones.

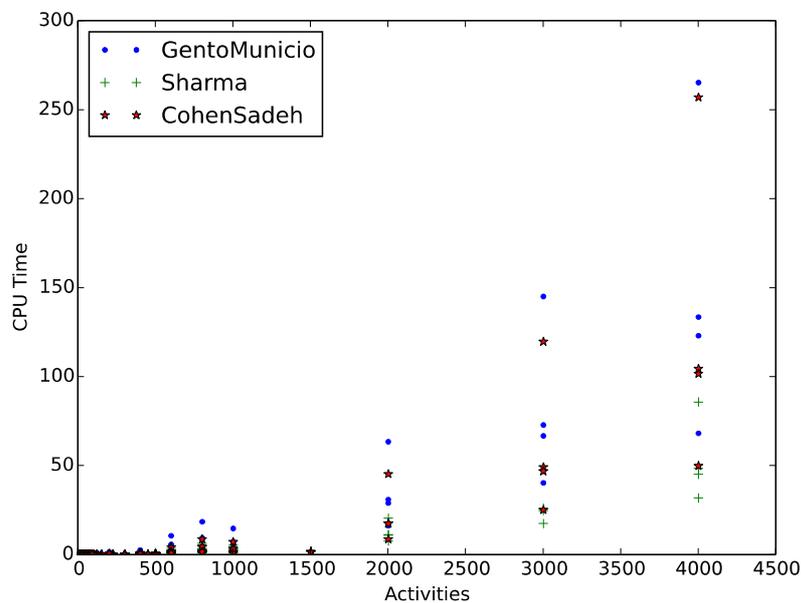
En la Tabla 1, se muestra una descripción general de los conjuntos de datos de los proyectos utilizados. Como se puede observar los conjuntos son muy diversos e incluyen proyectos con una gran cantidad de actividades y prelación.

Tabla 1: Características de los proyectos considerados por grupos

	<i>Nº de proyectos</i>	<i>Nº medio de actividades</i>	<i>Nº medio de prelaiones</i>
<i>Alvarez</i>	144	60.3	128.9
<i>Elmaghraby</i>	36	36.6	95.7
<i>Jobshop</i>	113	260.0	272.2
<i>Tavares (S)</i>	20	72.0	202.9
<i>Tavares (M)</i>	36	496.4	1372.8
<i>Tavares (L)</i>	16	2502.0	6661.4
<i>Patterson</i>	110	26.0	252.8
<i>Kolisch</i>	2040	79.6	143.8
Total	2515		

Se han utilizado los 2515 proyectos disponibles. Generando un grafo con cada uno de los tres algoritmos estudiados. Para validar que todos los grafos son correctos, se ha desarrollado un algoritmo de validación que extrae todas las prelaiones implícitas en el grafo AOA y comprueba que son idénticas a la tabla de prelaiones dada por los datos del proyecto (toda prelación presente en el grafo está presente en la tabla y viceversa). Obviamente, en el desarrollo final, todos los algoritmos son correctos y generan todos los grafos correctamente.

Figura 1: Tiempo de ejecución de cada algoritmo respecto al número de actividades del proyecto



Como variables independientes, o de entrada, para valorar el tamaño de los proyectos, hemos considerado:

- El número de actividades reales (*Activities*)

- El número de prelações establecidas entre actividades del proyecto (*Prelations*)

Como variables dependientes a estudiar hemos tomado:

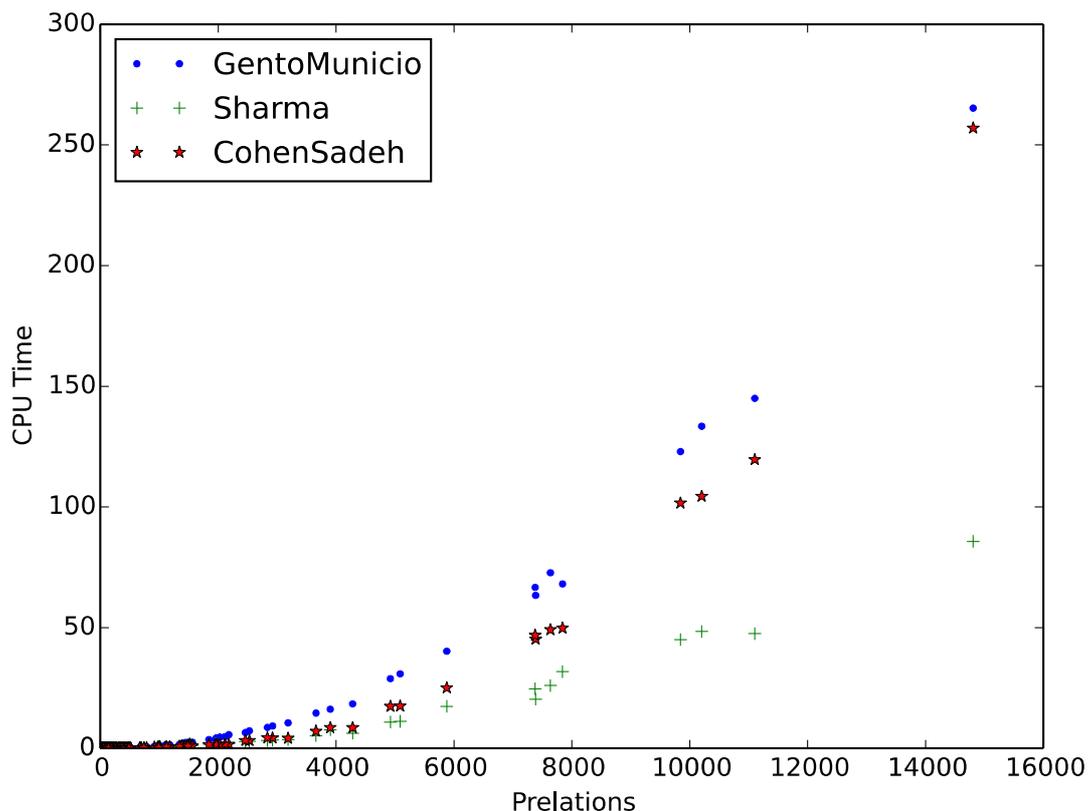
- El tiempo en segundos de ejecución real en la CPU del algoritmo (*CPU time*).
- Número de actividades ficticias generadas en el grafo (*Dummies*)

3.2 Resultados

Trabajando con el número de actividades como medida del tamaño del proyecto es más difícil apreciar la relación entre el tamaño y el tiempo de ejecución. Se percibe un gran dispersión de los resultados (ver Figura 1). Sin embargo, considerando el número de prelações en la Figura 2 podemos apreciar la relación muy claramente y ver que su tendencia de crecimiento no es lineal para ninguno de los algoritmos.

Además, se han detectado diferencias de tiempo sustanciales entre los tres algoritmos, que son especialmente notables en los proyectos grandes, donde los algoritmos Cohen&Sadeh y Gento-Municio tardan entre 2 y 3 veces más que Sharma.

Figura 2: Tiempo de ejecución de cada algoritmo respecto al número de prelações del proyecto



El número de actividades ficticias incluidas por cada algoritmo es muy parecido en la mayoría de los proyectos. Sin embargo, si se presta atención a los diferentes tipos de proyectos, hay diferencias importantes. Podemos verlo en la tabla 2, que muestra la suma

del número de ficticias de los proyectos de cada conjunto.

Tabla 2: Número de actividades ficticias en cada grupo de problemas

	<i>Cohen&Sadeh</i>		<i>Gento-Municio</i>		<i>Sharma</i>	
	Nº ficticias	%	Nº ficticias	%	Nº ficticias	%
<i>Alvarez</i>	8121	(62%)	5603	(43%)	13115	(100%)
<i>Elmaghraby</i>	0	(0%)	0	(0%)	2449	(100%)
<i>Jobshop</i>	0	(0%)	0	(0%)	0	(0%)
<i>Tavares (S)</i>	3265	(99%)	3278	(99%)	3305	(100%)
<i>Tavares (M)</i>	40041	(99%)	40318	(100%)	40298	(100%)
<i>Tavares (L)</i>	86540	(100%)	86971	(100%)	86880	(100%)
<i>Patterson</i>	1478	(79%)	1190	(64%)	1874	(100%)
<i>Kolisch</i>	144583	(100%)	144452	(100%)	145102	(100%)

En general, Sharma es el algoritmo que más ficticias genera, con diferencias importantes en los conjuntos de: Álvarez (donde Gento-Municio reduce hasta un 67%), Elmaghraby (todas las ficticias son innecesarias) y Patterson (36%). En cambio, en los demás conjuntos, los resultados son muy similares.

La figura 3 muestra un grafo generado con el algoritmo de Sharma que tiene varias actividades ficticias innecesarias. Concretamente, los nodos 5, 6, 8 y 9 podrían hacerse uno sólo y bastaría con una ficticia para resolver la actividad paralela. Este tipo de patrones no son detectados en la fase de eliminación de ficticias innecesarias de Sharma. En cambio, los algoritmos de Cohen & Sadeh y Gento-Municio crean el grafo óptimo para este proyecto, que se puede observar en la Figura 4.

Figura 3: Grafo con patrón de ficticias innecesarias no eliminadas con el algoritmo de Sharma

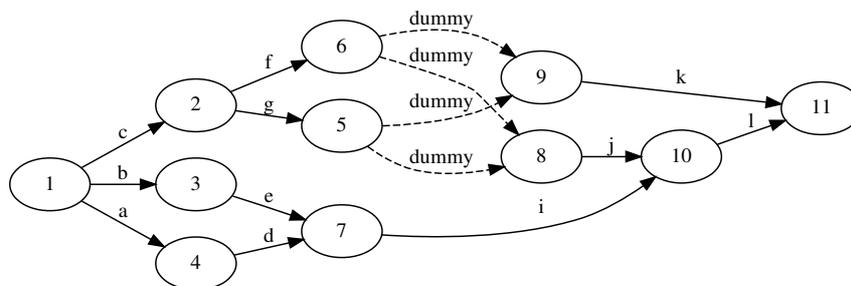
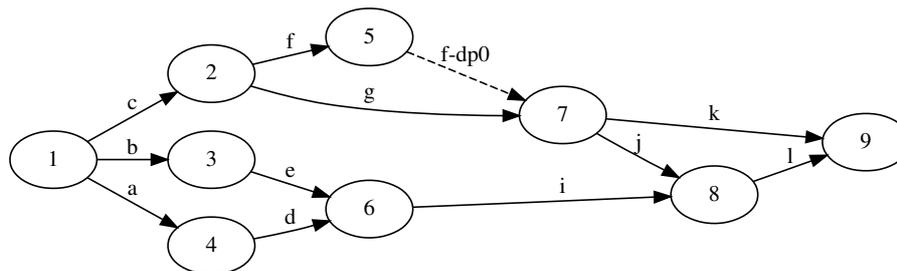


Figura 4: Grafo sin ficticias innecesarias generado con algoritmo Cohen-Sadeh



Entre los algoritmos de Cohen&Sadeh y Gento-Municio las diferencias respecto al número de ficticias generadas son menores en general. Sin embargo, hay dos conjuntos (Álvarez y Patterson) donde la diferencia es importante a favor de Gento-Municio. Para ilustrar en qué casos genera más ficticias Cohen&Sadeh, las figuras 5 y 6 muestran los grafos generados para un ejemplo pequeño (ST2748) del conjunto de Álvarez. En ellos, se puede ver cómo la reducción del número de ficticias la aporta la detección de los patrones que Gento-Municio denomina cadenas.

Figura 5: Sección del grafo generado por el algoritmo de Cohen-Sadeh para el proyecto Alvarez/ST2748

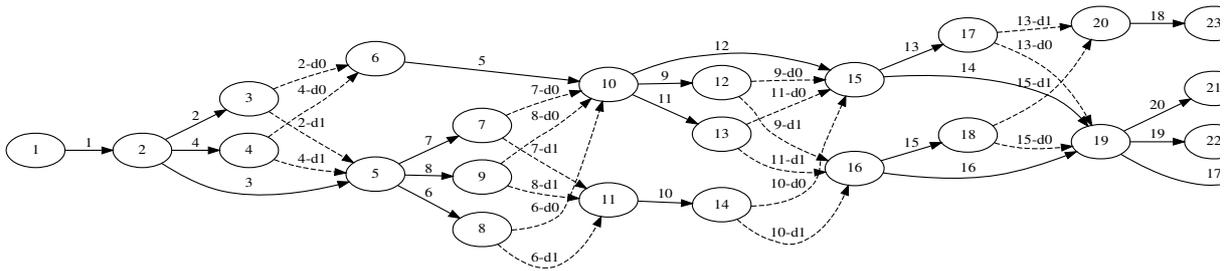
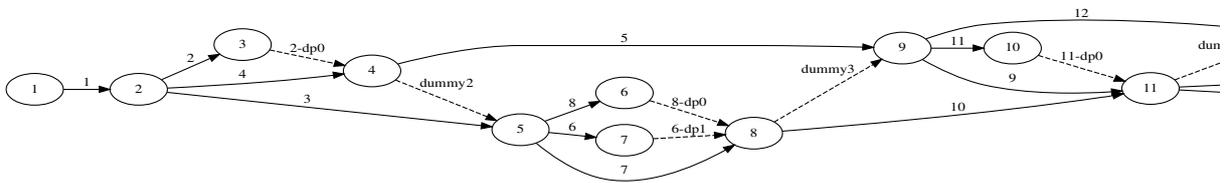


Figura 6: Sección Grafo generado por el algoritmo de Gento-Municio para el proyecto Alvarez/ST2748



Conclusiones y trabajo futuro

Se han comparado tres algoritmos aproximados para la generación de grafos AOA. En general, Sharma es el algoritmo más eficiente pero también el que genera un mayor número de actividades ficticias. Sin embargo, esto no es determinante porque, dependiendo del tipo de proyectos, el número de actividades ficticias es muy parecido al de los otros y es incluso menor en algunos casos. Entre los algoritmos que generan menos ficticias, Cohen&Sadeh y Gento-Municio, parece ser más interesante la aplicación de Gento-Municio ya que en un menor tiempo de ejecución obtiene grafos con un número similar, o incluso menor, de actividades ficticias.

Se ha visto que la base de datos de proyectos grandes utilizada es muy diversa y eso genera resultados muy diferentes dependiendo del algoritmo aplicado. Esto abre la puerta a que los proyectos puedan ser caracterizados en función de los tipos de patrones que incluyen.

Por otra parte, los datos de muchos de los proyectos utilizados son generados artificialmente, como trabajo futuro, una recopilación de datos de proyectos grandes reales sería muy interesante. Gracias a la popularización del Open Data puede que esto sea más

sencillo en breve. También, sería interesante implementar y comparar otros algoritmos publicados. Concretamente, el algoritmo propuesto por Mouhoub (2011) tiene una base matemática muy interesante y prometedora.

Bibliografía

Cohen, Y., & Sadeh, A. (2006). A new approach for constructing and generating AOA networks. *Journal of Engineering, Computing and Architecture*, 1, 1–15.

Gento-Municio, A. M. (2004). Un algoritmo para la realización de grafos con las actividades en los arcos, grafos PERT. *Cuadernos Del CIMBAGE*, (7), 103.

Kolisch, R., & Sprecher, A. (1997). PSPLIB-a project scheduling problem library: OR software-ORSEP operations research software exchange program. *European journal of operational research*, 96(1), 205-216.

Krishnamoorthy, M. S., & Deo, N. (1979). Complexity of the minimum-dummy-activities problem in a pert network. *Networks*, 9(3), 189–194. <http://doi.org/10.1002/net.3230090302>

Mouhoub, N. E., Benhocine, A., & Belouadah, H. (2011). A new method for constructing a minimal PERT network. *Applied Mathematical Modelling*, 35(9), 4575–4588. <http://doi.org/10.1016/j.apm.2011.03.031>

Project Management Institute. (2013). *A Guide to the Project Management Body of Knowledge (PMBOK® Guide)*. Project Management Institute, Incorporated.

Salas-Morera, L., Arauzo-Azofra, A., García-Hernández, L., Palomo-Romero, J. M., & Hervás-Martínez, C. (2013). PpcProject: An educational tool for software project management. *Computers & Education*, 69, 181-188.

Sharma, R. R. K. (1998). A new algorithm for preparing PERT networks. *Asia Pacific Journal of Operational Research*, 15, 37–48.