



Cuadernos del CIMBAGE

ISSN: 1666-5112

cimbage@econ.uba.ar

Facultad de Ciencias Económicas
Argentina

Gento Municio, Angel M.

Un algoritmo para la realización de grafos con las actividades en los arcos -grafos pert-
Cuadernos del CIMBAGE, núm. 7, 2005, pp. 103-126

Facultad de Ciencias Económicas
Buenos Aires, Argentina

Disponible en: <http://www.redalyc.org/articulo.oa?id=46200708>

- Cómo citar el artículo
- Número completo
- Más información del artículo
- Página de la revista en redalyc.org

redalyc.org

Sistema de Información Científica

Red de Revistas Científicas de América Latina, el Caribe, España y Portugal

Proyecto académico sin fines de lucro, desarrollado bajo la iniciativa de acceso abierto

UN ALGORITMO PARA LA REALIZACIÓN DE GRAFOS CON LAS ACTIVIDADES EN LOS ARCOS –GRAFOS PERT–

Angel M. Gento Municio
ETS Ingenieros Industriales
Universidad de Valladolid
Paseo del Cauce s/n
47011 – Valladolid - España
gento@eis.uva.es

Recibido 15 de noviembre 2004, recibido con modificaciones 16 de febrero 2005, aceptado 24 de febrero 2005

Resumen

El problema de dibujar redes con las actividades en los arcos (redes PERT) es un problema NP-completo. Diferentes autores (Syslo, 1984) han establecido límites al mismo.

En primer lugar debemos diferenciar entre redes con actividades en los nudos y redes con actividades en los arcos. Si las actividades están en los nudos, el dibujo de la red es muy fácil, pero cuando las actividades están en los arcos, generalmente es necesaria la utilización de actividades ficticias para mantener de forma correcta las relaciones entre las actividades.

En este artículo se propone un sencillo y didáctico algoritmo para el caso de un pequeño número de nodos donde es necesario un algoritmo intuitivo.

En el algoritmo se definen cuatro tipos diferentes de nodos que pueden aparecer en el grafo, permitiéndonos identificar las actividades ficticias tal y como se muestra en un ejemplo.

Palabras clave: *redes PERT, actividades ficticias.*

AN ALGORITHM TO MAKE ACTIVITY ON ARROW NETWORK

Angel M. Gento Municio
ETS Ingenieros Industriales
Universidad de Valladolid
Paseo del Cauce s/n
47011 – Valladolid - España
gento@eis.uva.es

Received 15 november 2004 , received in revised form 16 february 2005,
accepted 24 february 2005

Abstract

When there are activities in the nodes, it is easy to draw the graph of the network. But when there are activities in the arcs, we generally need dummy activities to keep the relations between activities.

In this paper we propose a simple and didactic algorithm with a small number of nodes where an intuitive algorithm is needed.

There are four different types of nodes that could appear in the graph. They are used to identify the dummy activities as it is shown in the example.

Key words: *PERT networks, dummy activity.*

1. INTRODUCCIÓN

El problema de dibujar redes con las actividades en los arcos (redes PERT) es un problema muy complejo en el que el tiempo de resolución aumenta más que polinomialmente que el número de actividades implicadas en el mismo. Este tipo de problemas se conocen con el nombre de NP-completos.

Para la representación de grafos PERT deben seguirse una serie de reglas de obligado cumplimiento:

1. En un grafo PERT sólo puede existir un suceso de inicio y uno de fin.
2. Toda actividad debe poseer, al menos, una actividad precedente y una posterior, salvo, lógicamente, las actividades que parten del suceso inicio del proyecto (que tienen actividades siguientes pero no precedentes) y las que llegan al suceso final (que tendrán actividades precedentes pero no siguientes).
3. Toda actividad ij llega a un suceso j de orden superior al suceso i del que parte. (recordamos la notación: la actividad ij parte del suceso i para llegar al suceso j).
4. No pueden existir dos (o más) actividades que, partiendo del mismo suceso inicial i , lleguen al mismo suceso final j o viceversa. Esto es, no puede haber dos (o más) actividades denominadas de la misma forma ij . Por supuesto, no hay inconveniente en que existan varias actividades ij, ik, \dots, im –caso de divergencia- o bien ij, kj, \dots, mj –caso de convergencia-.

La primera condición implica que tanto el comienzo como el fin del proyecto deben ser únicos. Entonces, si un proyecto puede comenzar con la ejecución de varias tareas de forma simultánea, todas ellas

deben partir de un mismo suceso inicio y análogamente ocurre para las actividades que concluyen el proyecto: todas ellas mueren en un mismo suceso de fin.

Las restricciones segunda y tercera suponen que toda actividad ij –es decir, todo arco ij en el grafo- forma parte de uno o varios caminos que conectan el nodo de inicio con el de fin del proyecto; además, el sentido de recorrido de estos caminos es el mencionado: de suceso de inicio a suceso de fin del proyecto. Por tanto, no existe posibilidad de hacer recorridos con retorno hacia el origen (ya que el grafo PERT es un grafo dirigido).

Y la última condición impide que surja una posible confusión entre actividades que, siendo diferentes, se denominan de la misma manera.

El cumplimiento de las normas y el respeto de las prelacións existentes conducen, en ocasiones, al empleo de ciertas conexiones lógicas denominadas actividades ficticias. Estas actividades no consumen recursos de ningún tipo ni tienen duración alguna; funcionan como puentes que permiten reflejar correctamente las relaciones entre actividades.

El objetivo de este trabajo es proponer un algoritmo intuitivo y fácilmente comprensible para la resolución de este tipo de problemas de forma didáctica.

2. DEFINICIONES

Para empezar a tomar datos se necesitan las actividades que forman parte de la red (del proyecto) y las siguientes. Las actividades precedentes son fáciles de deducir con los datos pedidos.

En la matriz de precedencia, se definen las actividades “madre” y las que las siguen tal y como se muestra en la Tabla 1.

Actividades “madre”	A	B	C	D	E	F
Siguientes	C	C, D, E	F	F	-	-

Tabla 1. Matriz de precedencia.

A lo largo de la explicación del algoritmo se denotarán las actividades “madre” mediante la letra A con un subíndice, A_i , y las actividades siguientes con la letra S más dos subíndices, S_{ij} , indicando el primero la actividad “madre” a la que pertenece. Así las actividades siguientes a la actividad A_i son $\Phi_i = \{S_{i1}, S_{i2}, \dots, S_{im}\}$.

Se van a distinguir 4 patrones en una red:

- Tipo I.
- Tipo II.
- Coincidencias.
- Cadenas.

Cada uno de estos patrones son excluyentes, o sea que si una actividad y sus siguientes pertenecen a un patrón, no pueden pertenecer a otro (excepto para los patrones de Tipo II incompletos y para las cadenas, pues tienen asociada una coincidencia).

Una de las premisas fundamentales para desarrollar este método es introducir, en cada paso que sea necesario, el menor número de elementos (nodos y actividades ficticias) posibles, de modo que las relaciones entre las actividades se respeten sin generar elementos superfluos.

2.1. Patrón de Tipo I.

En este caso, se tiene una actividad, S_{i1} , o un conjunto de actividades $\Phi_i = \{S_{i1}, S_{i2}, \dots, S_{im}\}$ que solo siguen a una actividad “madre” A_i , por lo que en la matriz de precedencia todas las actividades siguientes a A_i están exclusivamente en la fila de la actividad A_i .

Este conjunto de actividades $\{S_{i1}, S_{i2}, \dots, S_{im}\}$ sólo siguen a A_i por lo que su nodo inicial es igual al nodo final de A_i , y no es necesario introducir actividades ficticias para unirlos.

Por ejemplo, si tenemos en la matriz de preferencia la fila mostrada en la Tabla 2, y esas actividades siguientes no aparecen de nuevo en la matriz, tendremos un patrón de Tipo I, tal y como se muestra en la Figura 1.

A_i	S_{i1}, S_{i2}, S_{i3}
-------	--------------------------

Tabla 2. Ejemplo de patrón de Tipo I.

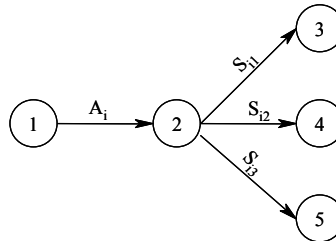


Figura 1. Mejor solución para el ejemplo de la Tabla 2.

2.1. Patrón de Tipo II.

Con este patrón, determinaremos si varias actividades “madre” pueden tener el mismo nodo final. Tendremos este patrón si dos (o más)

actividades “madre” tienen las mismas actividades siguientes sin diferencias. Se pueden distinguir dos casos:

1. Patrón de Tipo II Completo
2. Patrón de Tipo II Incompleto o Coincidencias.

En el primer caso, las actividades siguientes implicadas sólo aparecen con esas actividades “madre”. Esto permite asegurar que el nodo inicial de estas siguientes será el mismo que el final de las actividades “madre”.

Por ejemplo, si tenemos las filas mostradas en la Tabla 3 en la matriz de precedencia, formarán un patrón de Tipo II Perfecto y su representación gráfica será la Figura 2.

A_i	S_1, S_2, S_3
A_j	S_1, S_2, S_3

Tabla 3. Ejemplo de patrón de Tipo II Perfecto.

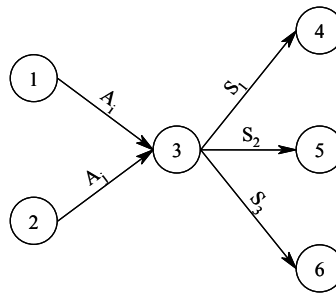


Figura 2. Mejor solución para el ejemplo de la Tabla 3.

Mientras que en el patrón de Tipo II Incompleto, al menos una de las actividades siguientes es también siguiente de otra u otras actividades

“madre” que no pertenecen a este patrón. Por ello, no se puede afirmar que el nodo de inicio de las actividades siguientes sea el nodo final de las actividades “madre”, pudiendo aparecer actividades ficticias.

2.3. Coincidencias

Esta situación aparece cuando 2 (o más) actividades tienen una (o más) actividades siguientes en común y una (o más) actividades siguientes diferentes, por lo que serán necesarias actividades ficticias.

En el ejemplo mostrado en la Tabla 4, necesitaremos una actividad ficticia para unir el nodo final de A_j con el nodo de inicio de la actividad siguiente en común S , con sentido tal que respete las relaciones en la matriz de precedencia (ver Figura 3).

A_i	S
A_j	S, S_{j2}, S_{j3}

Tabla 4. Ejemplo de patrón de Coincidencia.

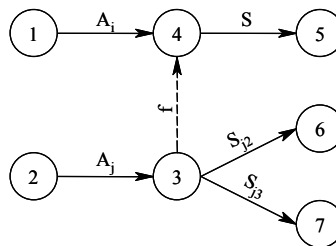


Figura 3. Mejor solución para el ejemplo de la Tabla 4.

2.4. Cadenas

En este caso, para tener un número mínimo de actividades ficticias, éstas deben estar concatenadas.

Por ejemplo, en la matriz de precedencias mostrada en la Tabla 5, la representación gráfica que necesita el menor número de actividades ficticias es la mostrada en la Figura 4.

A_1	S_1
A_2	S_1, S_2
A_3	S_1, S_2, S_3
A_4	S_1, S_2, S_3, S_4

Tabla 5. Ejemplo de patrón de Cadena.

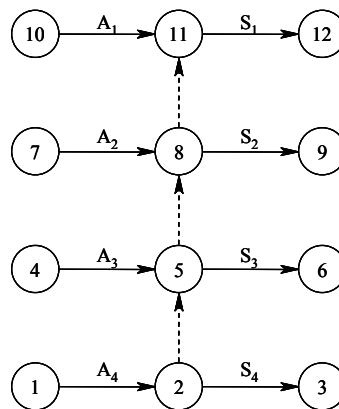


Figura 4. Mejor solución para el ejemplo de la Tabla 5.

3. ALGORITMO

Los pasos a seguir en el algoritmo propuesto son:

1. Búsqueda de actividades que comiencen en el nodo inicial.
2. Búsqueda de actividades que finalicen en el nodo final.
3. Búsqueda de patrones de Tipo I.

4. Búsqueda de patrones de Tipo II.
5. Búsqueda de Coincidencias.
6. Identificación de los nodos de inicio en las actividades de los patrones Coincidencia.
7. Búsqueda de Cadenas.
8. Identificación de los nodos finales del resto de actividades y determinación de las actividades ficticias.
9. Asignación de los nodos finales de los patrones de Tipo II Incompletos.

3.1. Búsqueda de actividades que comienzan en el nodo inicial

Las actividades que parten del nodo inicial son aquellas que no poseen tareas precedentes, o lo que es lo mismo, no aparecen en ninguna ocasión como actividad siguiente de ninguna otra, de forma que su identificación es sencilla con el vector NP porque podemos identificar las actividades A_i tales que $NP[A_i] = 0$.

Se puede disponer una matriz de nodos en la que se van apuntando los nodos de inicio y de fin para cada actividad. Para cada tarea madre en el proyecto debe responderse a la pregunta de la Figura 5 en orden a determinar su posible inicio en el nodo 0 del proyecto.

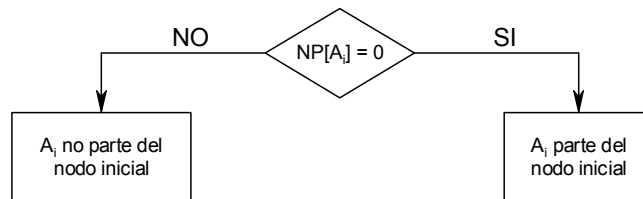


Figura 5. Búsqueda de actividades que parten del nodo inicial.

3.2. Búsqueda de actividades que finalicen en el nodo final

Estas actividades no tienen ninguna actividad posterior. Podemos determinarlas con el vector NS : basta con comprobar qué tareas A_i tienen $NS[A_i] = 0$. En la Figura 6 se presenta el esquema que debe seguirse para cada actividad, para determinar si acaba en el nodo final del proyecto.

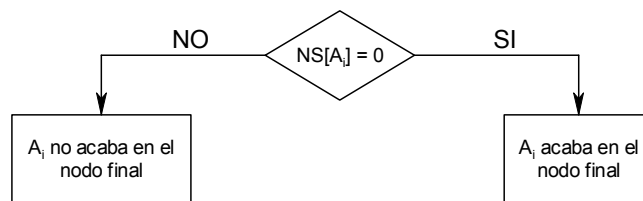


Figura 6. Determinación de actividades que llegan al nodo final.

3.3. Búsqueda de patrones de Tipo I

Debemos buscar en la matriz de precedencias aquellas actividades madre cuyas siguientes aparezcan una sola vez en la citada matriz como tareas siguientes. Si para la actividad A_i encontramos que $NP[S_{ij}] = 1, j = 1, 2, \dots, m_i$, donde $m_i = NS[A_i]$, entonces A_i es un patrón de Tipo I y el nodo final de A_i , $NF[A_i]$, y los nodos de inicio de los S_{ij} , $NI[S_{ij}]$, son iguales:

$$NF[A_i] = NI[S_{ij}], \quad j = 1, 2, \dots, m_i$$

Podemos numerar cada uno de los nodos creados en este patrón: el nodo inicial es el nodo 0 y el final el número 1. Para cada actividad, debe seguirse el esquema de la Figura 7.

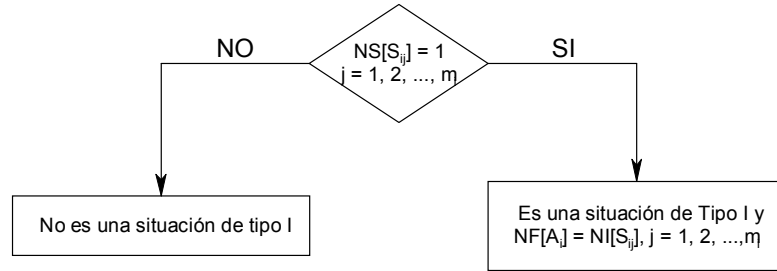


Figura 7. Búsqueda de patrones de Tipo I.

3.4. Búsqueda de patrones de Tipo II.

Buscaremos en la matriz de precedencias qué actividades “madre” tienen exactamente las mismas n actividades siguientes y asignaremos su fin al mismo nodo. Si para las actividades madre A_i, A_j, \dots, A_l se tiene que:

$$\Phi_i = \Phi_j = \dots = \Phi_l = \{S_1, S_2, \dots, S_n\}$$

entonces

$$NF[A_i] = NF[A_j] = \dots = NF[A_l]$$

Además, si este patrón es completo, el nodo final de las actividades “madre” implicadas y el nodo inicial de sus siguientes será el mismo:

$$NF[A_i] = NF[A_j] = \dots = NF[A_l] = NI[S_1] = NI[S_2] = \dots = NI[S_n]$$

y podremos numerarlo.

Pero, si una actividad S_k no cumple:

$$NP[S_k] = u \quad k = 1, 2, \dots, n$$

donde el número de actividades A_i, A_j, \dots, A_l , es u , estaremos en un patrón de Tipo II Incompleto y no podremos numerar el nodo final de las actividades “madre”.

El proceso de búsqueda está esquematizado en la .

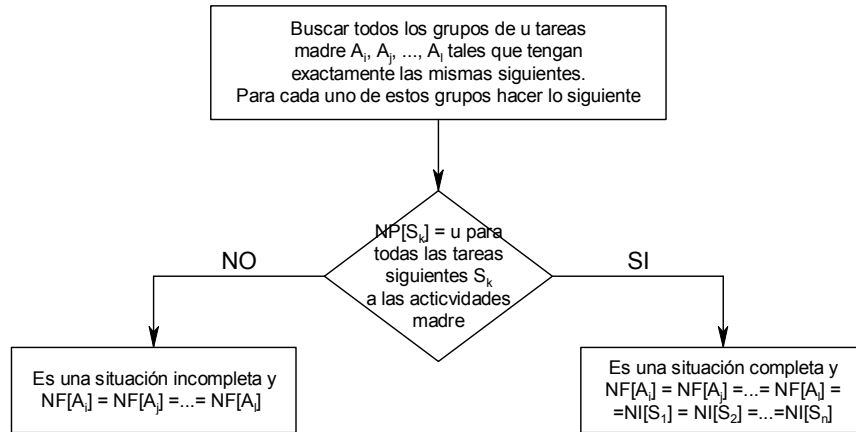


Figura 8. Búsqueda de patrones de Tipo II (Completos e Incompletos).

3.5. Búsqueda de Coincidencias.

Hasta ahora, se han asignado nodos de inicio a todas las actividades excepto para las encuadradas dentro de los patrones de Tipo II Incompletos (y las cadenas –todavía no analizadas-).

En primer lugar construiremos una nueva matriz, *MASC*, que contendrá una actividad “madre” (y sus siguientes) de cada Tipo II Incompleto. Con esta matriz, calcularemos el número de veces que en cada actividad figura como siguiente y lo almacenaremos en un vector, *NPC*. Para determinar los nodos de inicio, necesitamos ordenar las filas

de *MASC* de acuerdo con el número de actividades siguientes en coincidencia.

Para cada actividad siguiente con $NPC > 1$ buscamos en que otra u otras filas aparece; si la actividad contenida en la celda $MASC_{ij}$ se repite en la celda $MASC_{km}$, entonces existe una Coincidencia entre las actividades A_i y A_k (cada Coincidencia implica sólo a dos actividades madre). Deben buscarse sistemáticamente todas las Coincidencias presentes de esta manera.

3.6. Identificación de los nodos de inicio en las actividades de los patrones Coincidencias

Aquellas filas en *MASC* con una sola actividad siguiente identifican actividades cuyo nodo de inicio no puede estar compartido con ninguna otra tarea.

Para el resto de filas, usaremos los siguientes pasos:

1. Construiremos una matriz simétrica, *MRA* (Matriz de Relaciones entre Actividades), con filas (y columnas) igual al número de actividades sin nodo de inicio.
2. Para rellenar *MRA*, comprobaremos las actividades que están en *MASC* y sumaremos 1 por cada pareja de relaciones. Si en una fila de *MASC* aparecen S_i y S_j , sumaremos 1 en los elementos (i,j) y (j,i) de *MRA*.
3. Cuando rellenemos esta matriz, el elemento MRA_{ij} contendrá las veces en que S_i y S_j están juntos en *MASC*. Si coincide con el número total de veces en que S_i y S_j están en *MASC*, entonces S_i y S_j comenzarán en el mismo nodo de inicio.

Si tenemos el siguiente ejemplo, con la matriz de precedencia parcial mostrada en la Tabla 6.

Actividades “madre”	Actividades siguientes
A_i	A, B, C, D
A_j	B, C, D, E, F, G, H
A_k	C
A_l	C, E, F

Tabla 6. Ejemplo de matriz *MASC*.

La matriz *MRA* que se obtiene es la de la Tabla 7.

		A	B	C	D	E	F	G	H
1	A	-	1	1	1				
2	B	1	-	2	2	1	1	1	1
4	C	1	2	-	2	2	2	1	1
2	D	1	2	2	-	1	1	1	1
2	E		1	2	1	-	2	1	1
2	F		1	2	1	2	-	1	1
1	G		1	1	1	1	1	-	1
1	H		1	1	1	1	1	1	-

Tabla 7. Matriz *MRA* para la Tabla 6.

La actividad *A* comienza en un nodo de inicio que no utilizan otras actividades porque MRA_{AB} es 1 pero *B* aparece en *MASC* 2 veces, MRA_{AC} es 1 pero *C* aparece 4 veces en *MASC* y MRA_{AD} es 1 pero *D* aparece 2 veces en *MASC*, aunque *A* aparece una única vez en *MASC*. Si se analizan *B* y *D*, se puede comprobar que aparecen 2 veces en *MASC* y que el elemento MRA_{BD} es 2, por lo que *B* y *D* tienen el mismo nodo de inicio. De forma similar se pueden analizar el resto de actividades y se obtiene que *E* y *F* comienzan en el mismo nodo de inicio y que *G* y *H* también.

En la Figura 9 se muestra el esquema de este paso.

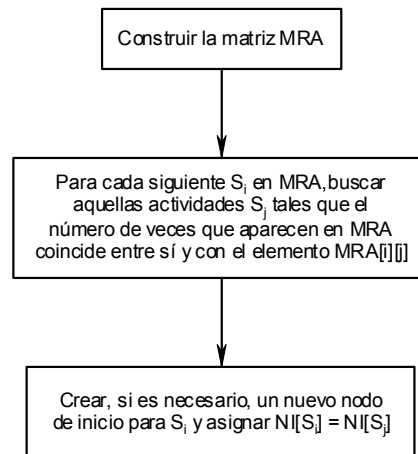


Figura 9. Identificación de los nodos de inicio en las actividades de los patrones Coincidencias.

3.7. Búsqueda de Cadenas

Una cadena es un patrón en el que una actividad ficticia termina en un nodo donde comienza otra actividad ficticia.

Para buscar las cadenas, deberemos construir una nueva matriz, *MNS* (Matriz de Nodos Sigüientes), donde se anotarán los nodos de inicio de las actividades sigüientes de la matriz *MASC* (ya calculada con anterioridad). En realidad tenemos la misma información pero con un formato más adecuado.

A continuación deberemos construir una matriz simétrica con el mismo número de filas (y columnas) que el número de nodos de la matriz *MNS*, *MRN* (Matriz de Relaciones entre Nodos). Para rellenar *MRN* comprobaremos las actividades que están en *MNS* y sumaremos 1 por cada pareja de relaciones.

El elemento MRN_{ij} muestra el número de veces en que los nodos i y j están unidos en *MNS*. Si coincide MRN_{ij} con el número total de veces en que el nodo i está en *MNS*, entonces dicho nodo está siempre unido al nodo j . Por tanto, en cada fila los elementos j que tengan el mismo número que el total en el nodo i en *MNS* forman una cadena parcial.

Para facilitar el análisis posterior, anotaremos las cadenas parciales obtenidas en una matriz, *MC* (Matriz de Cadenas):

1. Si para el nodo i , todos los nodos sigüientes j son nodos finales de las cadenas parciales, deberemos usar una actividad ficticia que comience en el nodo i y finalice en cada uno de los nodos j .
2. Para el resto de los nodos, uniremos el nodo i con el nodo j que tenga el mayor número de actividades sigüientes. Borraremos el nodo j y todos sus nodos sigüientes y se vuelve a comprobar que nodo es el que tiene el mayor número de nodos sigüientes repitiendo el proceso hasta borrar todos los nodos.

Si tenemos la matriz de precedencias parcial mostrada en la Tabla 8, podemos construir la matriz MNS (Tabla 9) y la matriz MRN (Tabla 10).

Actividades "madre"	Actividades siguientes
A_i	C
A_j	D
A_k	B, C, D
A_l	A, B, C, D

Tabla 8. Ejemplo de actividades con patrón Cadena.

Actividades "madre"	Actividades siguientes
A_i	3
A_j	4
A_k	2, 3, 4
A_l	1, 2, 3, 4

Tabla 9. Matriz MNS del ejemplo de la Tabla 8.

		1	2	3	4
1	1	-	1	1	1
2	2	1	-	2	2
3	3	1	2	-	2
3	4	1	2	2	-

Tabla 10. Matriz MRN del ejemplo de la Tabla 8.

Con estas matrices, ya se pueden identificar las cadenas parciales. El nodo 1 está una vez en MNS, por lo que (desde MRN) podemos comprobar que los nodos 2, 3 y 4 forman la cadena parcial para el nodo 1. El nodo 2 tiene a los nodos 3 y 4 en su cadena parcial y los nodos 3 y 4 son los nodos finales de la cadena. Tenemos pues la matriz MC ya ordenada tal y como aparece en la Tabla 11.

Nodo	Nodos siguientes
4	-
3	-
2	3, 4
1	2, 3, 4

Tabla 11. Matriz MC (ordenada) del ejemplo de la Tabla 8.

Ahora, buscaremos aquellos nodos que no tienen nodos siguientes: nodos 3 y 4. Esos son los últimos nodos de alguna cadena y desde ellos no puede partir ninguna actividad ficticia. Para el nodo 2, los dos

nodos siguientes (3 y 4) son nodos finales y se necesitan dos actividades ficticias para unir el nodo 2 con los nodos 3 y 4. Por último, para la fila del nodo 1, el nodo siguiente con el mayor número de nodos siguientes es el nodo 2, por lo que se deberá unir el nodo 1 y el 2 con una actividad ficticia.

Por tanto, la mejor representación es la mostrada en la Figura 10. Pueden dibujarse otros grafos, pero utilizando un mayor número de actividades ficticias lo que iría en contra de la metodología PERT.

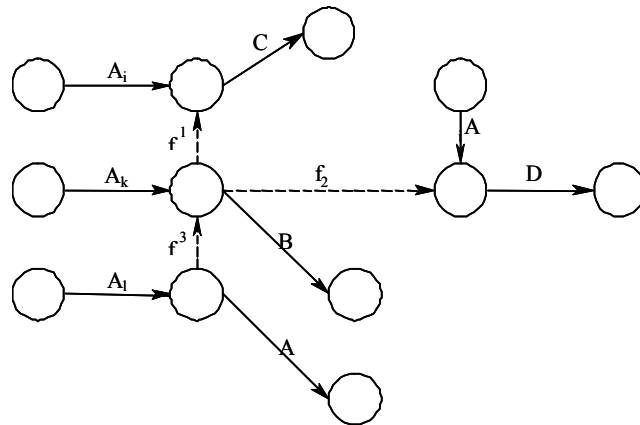


Figura 10. Ejemplo de cadena de la Tabla 8.

El diagrama de la Figura 11 sintetiza los pasos que exige la etapa de búsqueda de cadenas.

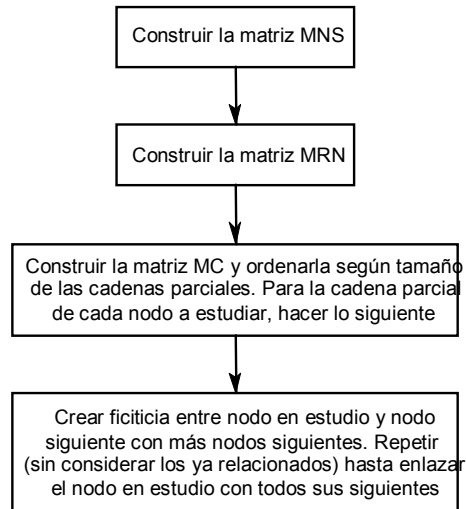


Figura 11. Búsqueda de Cadenas.

3.8. Identificación de los nodos finales del resto de actividades y determinación de las actividades ficticias

En este punto del análisis quedan por asignar los nodos finales a las actividades que forman parte de la matriz MASC y posiblemente aún deba determinarse alguna ficticia (en el caso de que se deba crear algún nuevo nodo de fin para una de estas actividades).

En primer lugar debemos comprobar si podemos asignar como nodo final de cada actividad “madre” A_i de MASC, alguno de los nodos de inicio de sus actividades siguientes:

1. Caso I. La actividad “madre” tiene una única actividad siguiente S. Si $NPC/S > 1$, el nodo final de las otras actividades “madre” de las que S es siguiente deben unirse al nodo inicial de S mediante actividades ficticias, pero si $NPC/S = 1$, estaremos en el caso II.

2. Caso II. Existen varias actividades siguientes para la actividad “madre” A_i , y al menos una de esas siguientes S , tiene $NPC[S]=1$. En este caso, el nodo de inicio de S puede ser el nodo final de la actividad “madre” A_i .
3. Caso III. Existe alguna actividad siguiente S_{ij} para la actividad “madre” A_i de la matriz *MASC* que aparece $NPC[S_{ij}]$ veces para la que podemos encontrar $NPC[S_{ij}] - 1$ otras filas en *MASC* en las que figuran todas las posteriores a la actividad A_i , además de, posiblemente, otras siguientes. Se debe empezar analizando las actividades siguientes S_{ij} con valor $NPC[S_{ij}]$ más pequeño; una vez encontrada alguna tarea S^* que cumpla la condición anterior haremos: $NF[A_i] = NI[S^*]$.

Si no se está en ninguno de los casos anteriores, forzosamente se debe crear un nuevo nodo para fin de la actividad madre analizada.

En cualquier caso, una vez que la actividad madre en cuestión tiene nodo de fin sólo resta comprobar que el resto de actividades que la siguen están enlazadas a la misma, bien porque parten del mismo nodo, o bien mediante alguna ficticia. Si el resto de siguientes parte del mismo nodo de fin de la actividad madre, no queda nada por hacer; en otro caso, se debe buscar si existe algún nodo que permita llegar a una o varias de las actividades siguientes utilizando para ello una ficticia.

La Figura 12 contiene el diagrama de flujo de la asignación de los últimos nodos de fin y actividades ficticias.

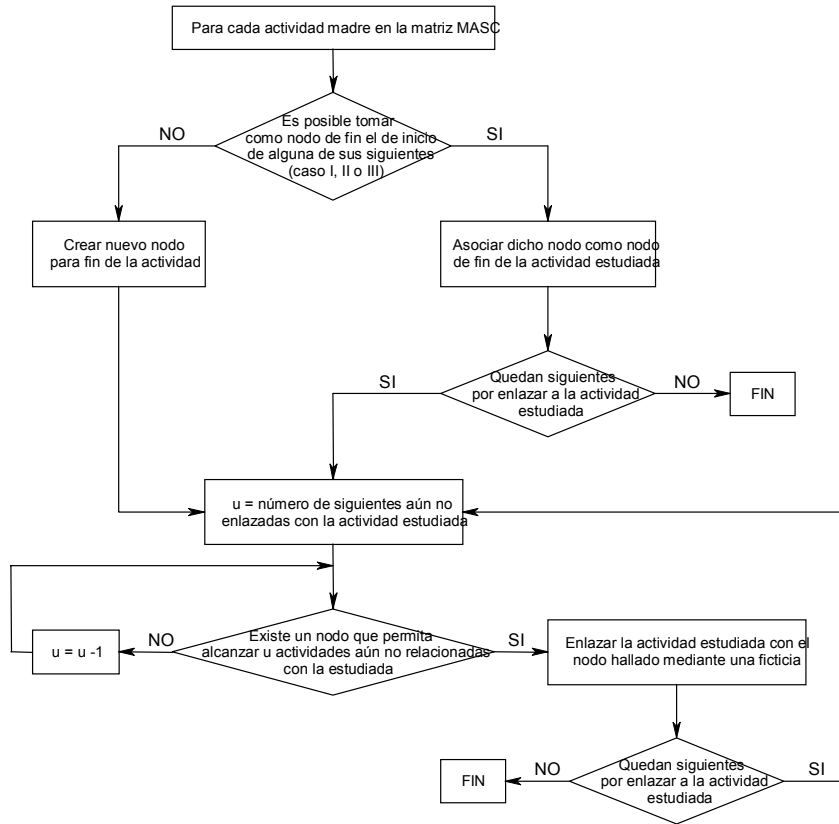


Figura 12. Asignación de últimos nodos de fin y ficticias.

3.9. Asignación de los nodos finales de los patrones de Tipo II Incompletos

Ya tenemos todos los nodos iniciales y finales asignados excepto los nodos finales de las actividades “madre” de los patrones de Tipo II Incompletos, pero una de esas actividades “madre” sí tiene el nodo final asignado pues lo ha sido en el análisis del patrón Coincidencia. Por tanto, solo restará buscar el resto de actividades “madre” y asignarlas el mismo nodo final.

4. CONCLUSIONES

El algoritmo propuesto es muy intuitivo pues va clasificando las diferentes actividades según patrones predeterminados que luego hay que ir enlazando, siguiendo un proceso similar al establecido de forma manual.

Uno de los problemas a resolver es el grado de complicación que se produce cuando el número de actividades aumenta, llegando a hacer el análisis inabordable de forma manual en un tiempo razonable.

Es un algoritmo muy sencillo de implementar en un ordenador lo que facilita los análisis complejos y permite elaborar proyectos PERT (con las actividades en los arcos) a partir de la tabla de precedencias.

REFERENCIAS

- [1] Krishnamoorthy, M.S.; Deo, N. (1979) "Complexity of minimum-dummy activities problem in a PERT network", *Networks*, vol. 9, pp. 189-194.
- [2] Mrozek, M. (1989) "Transitively reduced and transitively close event networks", *Networks*, vol. 19, pp. 47-72
- [3] Sharma, R.R.K. (1998) "A new algorithm for preparing PERT networks", *Asia-Pacific Journal of Operational Research*, vol. 15, pp. 37-48.
- [4] Syslo, M.M. (1981) "On computational complexity of minimum-dummy activities problem in a PERT network", *Networks*, vol 14, pp. 37-45.