

La programación de proyectos con recursos limitados

MANUEL ORTIGUEIRA BOUZADA (*)

I. PLANTEAMIENTO DEL PROBLEMA

Los métodos de programación reticular por "camino crítico" (PERT, CPM., etc.), que en los últimos años han evolucionado de forma vertiginosa, permiten establecer un programa que minimiza la duración de un proyecto considerando únicamente:

- las restricciones secuenciales de las actividades impuestas por la tecnología del proyecto examinado; y
- los tiempos estimados para la ejecución de cada una de las tareas.

Implícitamente, se presupone la suficiente disponibilidad de recursos en el momento de ser reclamados por las distintas actividades de un proyecto. Empero, los programas resultantes pueden no corresponder a la realidad, por existir restricciones que invalidan las bases para la determinación del camino crítico, siempre y cuando, en el programa, no sean recogidas tales limitaciones (empleando, por ejemplo, redes simuladas) (1).

Por otra parte, la dificultad de las empresas para conseguir adecuar sus disponibilidades de recursos a las fluctuaciones de los distintos proyectos cuya ejecución acometen, convierte la limitación de recursos en el problema más capital de la programación de proyectos. Se presenta, a menudo,

(*) Colaborador de la cátedra de Economía de la Empresa, que dirige el catedrático doctor Suárez y Suárez, de la Facultad de Ciencias Económicas de Málaga.

(1) JAMES M. ANTILL y RONALD W. WOODHEAD: *Método de la ruta crítica y su aplicación a la construcción*, Limusa-Wiley, S. A., México, 1967.

la imposibilidad de afrontar el comienzo de una actividad crítica debido a que los especialistas, los equipos o los medios financieros previstos como disponibles, no lo están. Esto provoca, en definitiva, el retraso en la fecha de terminación del proyecto.

Bernard Roy (2), en su sistemática, considera la limitación de recursos como una restricción "acumulativa" de los problemas de programación. Otras restricciones, llamadas "disyuntivas", aparecen cuando dos tareas se han de ejecutar en intervalos de tiempo distintos, por utilizar un mismo equipo. Señala Roy (3) que las restricciones acumulativas son de difícil formulación matemática. Ello se debe a que corresponden a fenómenos imprecisos, vagos, adaptativos, plásticos. Frente a las restricciones acumulativas, las restricciones disyuntivas corresponden a fenómenos rígidos, rigurosos, sin matices. Aunque en este trabajo no consideraremos este tipo de restricciones disyuntivas, diremos que dan lugar, cuando aparecen, a problemas combinatorios muy difíciles, cuyo tratamiento puede verse en Bernard Roy (4 y 5) y Alain Durand (6).

La búsqueda de una programación óptima en un proyecto, cuando existen recursos limitados, no resulta muy complicada en el caso de redes simples y con reducido número de tareas; pero, a medida que las redes se hacen más complejas, el problema se torna más difícil. Esta dificultad, intrínseca al problema, aumenta por la naturaleza combinatoria del mismo, al tener que coordinar restricciones y secuencias tecnológicas con requerimientos y disponibilidades. Por otra parte, una formulación matemática rigurosa choca con la relativa flexibilidad de la mayor parte de las restricciones que pesan sobre el problema. No se dispone hasta el presente, como

(2) BERNARD ROY: *Physonomie et traitement des problèmes d'ordonnancement*, Gestion, Avril, 1963.

(3) BERNARD ROY: *Algunos aspectos teóricos de los problemas de programación*, Coloquio hispano-francés sobre métodos modernos de gestión, Barcelona, 1964.

(4) BERNARD ROY - B. SUSSMANN: *Les problèmes d'ordonnancement avec contraintes disjonctives*, "Rapport de Recherche", núm. 9, Direction Scientifique SEMA, Décembre, 1964.

(5) BERNARD ROY: *Prise en compte des contraintes disjonctives dans les méthodes de chemin critique*, "Revue Française de Recherche Operationnelle", número 38, 1966.

(6) ALAIN DURAND: *Une méthode optimale de traitement des contraintes disjonctives dans les problèmes d'ordonnancement. Application a la construction d'un barrage*, "Revue Française de Informatique et de Recherche Operationnelle", número 3, 1967.

dicen J. A. Carruthers y A. Battersby (7), de algoritmos que sean a la vez generales, rigurosos y practicables: los hay rigurosos, pero no practicables; practicables, pero no rigurosos; rigurosos y practicables, pero sólo en casos especiales. Así, por ejemplo, entre los distintos instrumentos que brinda la investigación operativa, está la programación lineal. Esta permite definir un problema, como dice T. L. Pascoe (8), en términos de su lógica, requerimientos y disponibilidades de recursos, logrando alcanzar una solución óptima exacta en base a su función objetivo. Sin embargo, el número de desigualdades necesarias para modelar el proyecto es muy elevado. Así, Wiest, ha establecido que un proyecto de 55 tareas y 4 tipos de recursos, a ejecutar en 30 días, conduce a un programa lineal con 5.875 ecuaciones y 1.650 variables (sin contar las variables de holgura o las necesarias para encontrar una solución en números enteros). Esto hace que el tiempo de tratamiento del programa en un ordenador tenga un coste prohibitivo. No obstante, es de esperar que rápidos avances en la programación lineal y otros instrumentos de la investigación operativa (métodos, por ejemplo, de separación y evaluación progresiva o "Branch and Bound") (9) hagan viables estos métodos.

Se busca resolver la programación con recursos limitados mediante aproximaciones que dan resultados razonables. El proceso seguido es más bien de mejoramiento que de optimización (7). Se encuentran soluciones rápidas y muy satisfactorias recurriendo al empleo de algoritmos heurísticos, cuya operatoria sigue un proceso iterativo a base de aproximaciones sucesivas. Un ordenador electrónico hace posible, luego, su desarrollo. Existe un gran número de algoritmos heurísticos. Cada uno de ellos tiene en cuenta un mayor o menor número de factores: uno o varios recursos, uno o varios grados de intensidad, uno o varios proyectos, etc. La mayor parte de estos algoritmos realizan la asignación de recursos en serie o en paralelo. Los métodos en serie, como dice R. L. Martino (10), asignan los recursos según el orden establecido en una lista comprensiva de todas las

(7) J. A. CARRUTHERS y ALBERT BATTERSBY: *Advances in Critical Path Methods*, "Operational Research Quarterly", vol. 17, núm. 4, 1966.

(8) T. LAWRENCE PASCOE: *Allocation of Resources CPM*, "Revue Française de Recherche Operationnelle", núm. 38, 1966.

(9) RENÉ DESCAMPS y PIERRE CHEVIGNON: *Algorithmes d'optimisation pour une classe générale de problèmes d'ordonnement avec limitations de ressources*, METRA, vol. V, núm. 4, 1966.

(10) R. L. MARTINO: *Asignación y Programación de Recursos*, "Administración y Control de Proyectos", tomo III, Editora Técnica, México, 1967.

actividades. En ella, las tareas se disponen siguiendo un determinado criterio de precedencia. La asignación a las tareas se hace una a una, de forma que se asegure el examen de todas las precedentes. Dos sistemas de prioridad pueden ser empleados: uno dinámico y otro estático, según varíe o no la lista inicialmente establecida. Los algoritmos basados en la asignación en paralelo programan a la vez varias actividades. A pesar de que aparentemente presentan diferencias con los de asignación en serie, éstas no son importantes al adoptar un procedimiento dinámico en la elaboración de la lista. Se trata, en cierto sentido, de "un enfoque en paralelo a un procedimiento en serie".

En la programación de proyectos con recursos limitados existen dos problemas implicados, que nosotros trataremos en este trabajo: uno, el problema de *nivelación*; otro, el problema de *compatibilidad*.

Antes de pasar a tratar, en los próximos epígrafes, cada uno de los problemas anteriores y su instrumento adecuado, conviene establecer algunas consideraciones básicas.

II. LOS RECURSOS

La estimación de la duración, en cada una de las tareas de un proyecto, se debe hacer teniendo en cuenta la cantidad y la naturaleza del recurso o recursos necesarios para su ejecución.

Los recursos pueden ser muy variados. Así, tenemos: materias primas de distintas clases, mano de obra de diversas especialidades, equipos, dinero, espacio y cualesquiera otros medios.

Generalmente, el conjunto de actividades de un proyecto reclama distintas clases de recursos. De éstos, uno o varios desempeñan un carácter principal, lo que obliga a establecer sobre ellos un alto grado de planificación y control, en parangón con aquellos otros que se pueden conseguir con gran facilidad y reducido coste.

Para su ejecución, en un momento dado, cada tarea consume una cantidad determinada de recursos. Al número de unidades técnicas de este recurso o recursos aplicados en ese momento le llamamos *intensidad* (I). Denominamos *cantidad de trabajo* (CT) al producto de la *intensidad* por la *duración* (D) de una tarea.

Por lo general, cada tarea es susceptible de ser programada con distin-

tos grados de intensidad. Existe una intensidad llamada *normal*, para la cual el *rendimiento* (R) de los recursos aplicados es máximo. En este caso, el coste de la actividad es mínimo. Los métodos PERT y CPM emplean esta intensidad.

La cantidad de trabajo necesaria para la ejecución de una tarea, al ser considerado el rendimiento, vendrá dada por la expresión:

$$CT = I \times D \times R$$

Los distintos grados de intensidad a los que puede programarse una tarea oscilan entre dos límites, mínimo y máximo, más allá de los cuales no habrá posibilidad de ejecución. Estos límites dependen fundamentalmente de las características y naturaleza de la tarea considerada.

El empleo de distintas intensidades en las actividades de un proyecto complica extremadamente, por la gran flexibilidad que se introduce, el proceso de programación.

Por lo regular, dentro de las posibilidades existentes, cada tarea se suele programar en base a tres niveles:

- *intensidad máxima o acelerada*, que emplea el mayor número posible de unidades técnicas del recurso necesario y da como resultado la ejecución en tiempo mínimo;
- *intensidad mínima o lenta*, que corresponde a la duración máxima y emplea el menor número posible de unidades técnicas de recurso;
- *intensidad normal*, que, como ya hemos dicho, corresponde al número de unidades técnicas aplicables en condiciones normales para alcanzar un rendimiento máximo y, por tanto, coste mínimo.

La intensidad a lo largo de la duración de una tarea no siempre se mantiene constante. De hecho, suele variar, y su comportamiento se asocia al de una curva cuyas características son: intensidad reducida, uniforme o creciente en su comienzo (preparación de los trabajos); intensidad elevada y uniforme en su tramo central (ejecución de los trabajos); y, por último, intensidad reducida, uniforme o decreciente en su terminación (acabado de los trabajos).

La posibilidad de programación intermitente en una actividad hace viable la descomposición en sus diversas partes: preparación, ejecución y acabado. Estas subtareas pueden ser, así, ejecutadas separadamente y a

intensidad constante, lo que da una mayor agilidad a la distribución de recursos.

Cuando, por razones técnicas, es posible la intermitencia en la ejecución de una actividad, ésta puede descomponerse igualmente; ahora bien, en este caso, es preciso asegurar la continuidad en la ejecución de los trabajos. De cualquier forma, si de la parcialización de una tarea no derivan beneficios importantes, se debe evitar la descomposición.

Los programas PERT y CPM constituyen el punto de partida para obtener las curvas de carga de cada uno de los recursos aplicados en un proyecto. La comparación de las curvas de carga de recursos reclamados con las curvas de carga de recursos disponibles puede poner de manifiesto la existencia de áreas de sobrecarga y subcarga que obstaculicen el logro de los objetivos prefijados en el plan. Entre la serie de medidas que tal vez permitan eliminar estas anomalías, tenemos:

- El desplazamiento de las actividades no críticas dentro de los límites permitidos por sus márgenes;
- la programación intermitente de actividades;
- la variación de la intensidad en la programación de las tareas;
- la aplicación de recursos suplementarios;
- la modificación del programa en el tiempo.

A la última de las medidas anteriores se recurre cuando, tras probar con las demás, no se consiguen superar las anomalías de carga existentes. Su empleo conduce a un cambio radical en el concepto de camino crítico. La ruta crítica deja de estar formada por una sucesión continua de actividades y pasa a ser una combinación de tareas del camino de tiempo crítico y tareas con recursos limitados. Este nuevo camino se denomina "camino crítico compuesto" (11).

III. LA NIVELACION DE RECURSOS

3.1. CARACTERÍSTICAS DEL PROBLEMA.

Este tipo de problemas se plantea cuando, disponiendo de los medios necesarios, se desea que los recursos reclamados durante el tiempo de

(11) A. R. BURGESS and J. B. KILLEBREW: *Variation in Activity Level on a cyclic Arrow Diagram*, "Journal of Industrial Engineering", vol. 13, núm. 2, March-April, 1962.

ejecución del proyecto —restricción a considerar— se mantengan a un nivel de carga uniforme. Al igual que sucede con la curva de intensidad en las actividades, también aquí podemos exceptuar un período inicial de empleo progresivo de recursos y un período final de utilización decreciente.

En los problemas de nivelación, equilibrado o alisado de recursos, la carga debe aproximarse tanto como sea posible, por exceso o por defecto, al nivel fijado de disponibilidades.

3.2. RESOLUCIÓN POR EL ALGORITMO DE BURGESS-KILLEBREW.

Los norteamericanos A. R. Burgess y J. B. Killebrew dan un enfoque sistemático al problema de nivelación. En (11) presentan un método en serie cuyas bases condensa J. F. Boss (12) en el siguiente postulado:

“La eficacia en la asignación de un recurso determinado, en función de una distribución ideal, varía en sentido inverso a la suma —obtenida en cada unidad de tiempo, del principio al fin del proyecto— de los cuadrados de las diferencias entre las cargas totales que corresponden a las dos asignaciones.”

Partiendo de un proyecto con tareas programadas a una intensidad constante, el algoritmo de Burgess-Killebrew intenta establecer una curva de carga tan uniforme como sea posible. Para ello, y dado que la carga media no varía por el hecho de que una tarea se haya adelantado o retrasado, es preciso minimizar la variancia de la carga. Esto se consigue minimizando la suma total de los cuadrados de las cargas de cada período de tiempo.

3.2.1. *Desarrollo del método.*

Se comienza por elaborar el instrumento convencional de representación del proyecto: el grafo. A continuación, se establecen los calendarios de comienzo de actividades en sus fechas más tempranas o más tardías, en forma de diagramas de Gantt. Para ello, se parte de un listado

(12) J. F. Boss: *Prise en consideration des contraintes pesant sur la disponibilité des moyens dans les methodes de chemin critique*, “Revue Française de Recherche Operationnelle”, núm. 38, 1966.

en el cual se relacionan las tareas del proyecto, siguiendo un criterio determinado en el orden de precedencias.

En el caso de que el calendario de partida se haya establecido a base de programar la ejecución de las actividades en sus fechas de comienzo más temprano, el algoritmo sigue los siguientes pasos:

1. Comenzar por el final del listado de tareas y tomar la primera actividad que posea margen disponible. Esta tarea se desplaza hacia la derecha, unidad por unidad de tiempo, determinando, cada vez, el efecto obtenido en la suma de cuadrados de carga.

2. Determinar, de todas las posiciones posibles de la tarea estudiada, aquella que, sin rebasar el margen disponible, totalice el valor más bajo en la suma de los cuadrados de carga. A valores iguales, se tomará aquél que sitúe la tarea lo más a la derecha posible. Corregir las "fechas límite" de terminación de las tareas precedentes a la examinada que hayan sido afectadas por el cambio. Esto será facilitado por el orden elegido al situar las actividades en el gráfico de Gantt.

3. Habiendo así encontrado para esta primera tarea una posición óptima, remontando la lista se pasa a la tarea más inmediata que disponga de margen. Esta tarea se somete a un tratamiento semejante al descrito para su antecesora, es decir, se desplaza hacia la derecha hasta el límite permitido por su margen, de forma que su ajuste represente una carga mínima.

El proceso descrito se sigue con las demás tareas hasta llegar a la primera de la lista. De esta forma es imposible llegar a una tarea sin haber examinado previamente todas las que le siguen.

El criterio de desplazar lo más a la derecha posible las tareas pretende dejar el mayor margen posible para su ajuste a las tareas precedentes.

4. Cuando se acaba el ciclo de cálculos se vuelve a comenzar de nuevo el proceso. Para ello se puede comenzar por el principio o por el final de la lista. Se finalizará cuando, tras un ciclo, ya no sea posible disminuir el valor del último criterio y se haya alcanzado un alisado satisfactorio. Dado que el algoritmo es heurístico, resulta imposible saber si se ha logrado un alisado óptimo, a no ser que se haya alcanzado una nivelación uniforme. Por otra parte, es necesario tener en cuenta que el nivel de disponibilidades no debe ser rebasado. Si lo fuera, será preciso probar otras colocaciones, partiendo de distintas secuencias de activida-

des al comenzar el proceso iterativo. Se pueden obtener distintos resultados partiendo de un gran número de ordenaciones de tareas.

El algoritmo de Burgess-Killebrew es muy sencillo. Para redes reducidas puede ser aplicado a mano o utilizando calculadoras convencionales de oficina. Sin embargo, para los problemas complejos se impone necesariamente el empleo de un ordenador electrónico. Esto no es un problema difícil, ya que Burgess y Killebrew han elaborado el programa de ordenador para su algoritmo. (Véase 11.)

Fundamentalmente el algoritmo es aplicable para el caso de un solo recurso; pero también podría ser empleado para varios. El inconveniente estriba en que el alisado de un recurso suele destruir, frecuentemente, el alisado de otro. De cualquier forma, el principio general de aplicación del algoritmo es estudiar por orden prioritario, si se puede establecer, el nivelado de los distintos recursos. También puede que sea posible establecer algún otro tipo de relación entre el recurso principal y los demás, en cuyo caso el alisado se intenta para bloques de recursos, cual si se tratara de uno solo.

3.2.2. Desarrollo de un ejemplo.

Un ejemplo servirá para ilustrar nuestra exposición.

Partiremos de una red del tipo que Savitzky-Sussman (13) denominan "potencial-tareas". Esta representación, junto con su facilidad para integrar todo tipo de restricciones potenciales, presenta la ventaja de una rápida elaboración y flexibilidad para los cambios, sobre todo cuando se planifica utilizando un procedimiento de tarjetas semejantes al descrito por J. A. Larkin (14). (Fig. 1.)

Podemos también partir de una representación convencional "potencial-etapas" (fig. 2), elaborada directamente o por transformación de la anterior, a base de un procedimiento "universal" similar al descrito por C. Blanc (15).

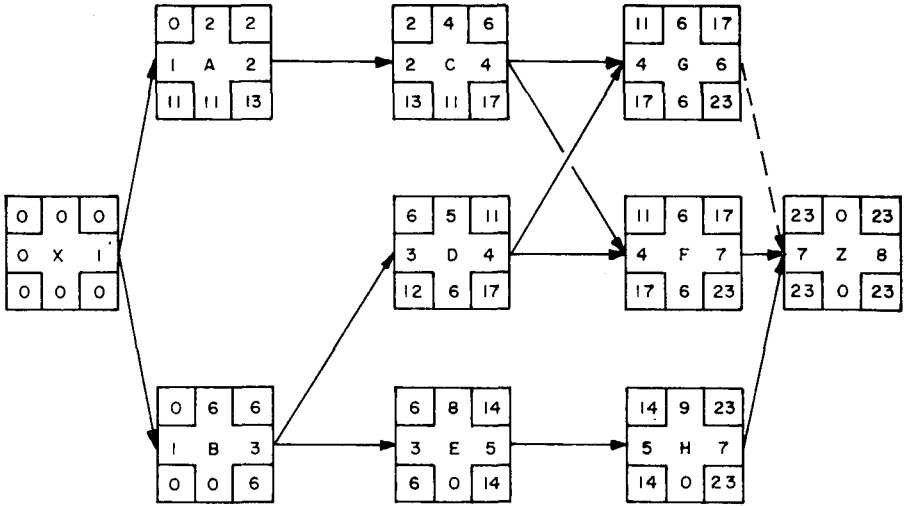
Observemos en este grafo (fig. 2) que, por razones que A. Suárez

(13) G. SAVITZKY y B. SUSSMANN: *La méthode des potentiels*, capítulo 4.º de la obra de A. BATTERSBY: *Méthodes modernes d'ordonnancement*, Dunod, París, 1967.

(14) J. A. LARKIN: *Speedier Methods of Network Planning*, "The Production Engineer", February, 1968.

(15) CHARLES BLANC: *Problèmes d'ordonnancement et méthodes de potentiel*, "Organisation Scientifique", 2/1, 1969.

Fig.-1



a	T	b
DESCRIPCION ACTIVIDAD		
c	H	d

- a = Fecha más temprana de comienzo.
- b = Fecha más temprana de terminacion.
- c = Fecha más tardia de comienzo
- d = Fecha más tardia de terminacion
- T = Duracion de la actividad
- H = Holgura total
- ij = Nudos inicial y final de una actividad

DISEÑO DE LA INFORMACION EN LA TARJETA.

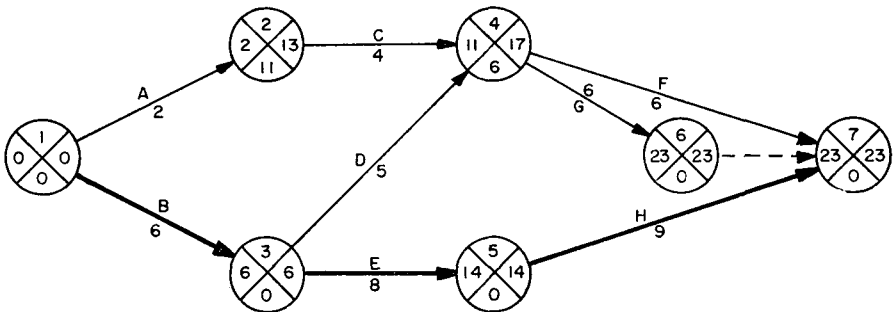


Fig.-2

advierte (16), hemos hecho coincidir los valores correspondientes a los nudos 6 y 7.

Supondremos, para simplificar, que cada una de las tareas del programa precisa para su ejecución una sola unidad técnica de un mismo recurso (un solo obrero y de una misma especialidad, por ejemplo).

Para pasar del programa propuesto (figs. 1 ó 2) al correspondiente diagrama de Gantt se utiliza la tabla I. En esta tabla o listado las tareas se ordenan por sus índices de cabeza (j) crecientes.

TABLA I

DESIGNACION TAREA	INDICES		DURACION	FECHA mas TEMP.		FECHA mas TARD.		RECURSOS		MARGEN	
	i	j		INICIO	TERMN.	INICIO	TERMN.	CLASE	CANTID.	TOTAL	LIBRE
A	1	2	2	0	2	11	13	P	1	11	0
B	1	3	6	0	6	0	6	P	1	0	0
C	2	4	4	2	6	13	17	P	1	11	5
D	3	4	5	6	11	12	17	P	1	6	0
E	3	5	8	6	14	6	14	P	1	0	0
G	4	6	6	11	17	17	23	P	1	6	6
F	4	7	6	11	17	17	23	P	1	6	6
H	5	7	9	14	23	14	23	P	1	0	0

A la vista de esta tabla se confecciona un gráfico de barras o de Gantt. Se emplea para ello una escala horizontal representativa del tiempo y una barra de longitud conveniente para cada tarea. Cada barra se numera en sus dos extremos con los índices i-j correspondientes.

En la elaboración del diagrama de Gantt comenzamos por la tarea A (1-2). A continuación, todas las actividades i-3; aquí, únicamente la actividad B (1-3), paralela a la actividad A. Pasamos a las actividades i-4: las tareas 2-4 y 3-4, que se sitúan a continuación de 1-2 y 1-3 respectivamente. Luego pasamos a las tareas i-5, i-6 e i-7. Podemos observar (fig. 3) que cada tarea, cuyo comienzo posee un índice determi-

(16) ANDRÉS SANTIAGO SUÁREZ SUÁREZ: *Consideración en torno a las holgas o tiempos sobrantes y a las actividades ficticias en el método PERT*, "Revista Técnica del Instituto de Censores Jurados de Cuentas de España", núm. 3, julio-septiembre, 1968.

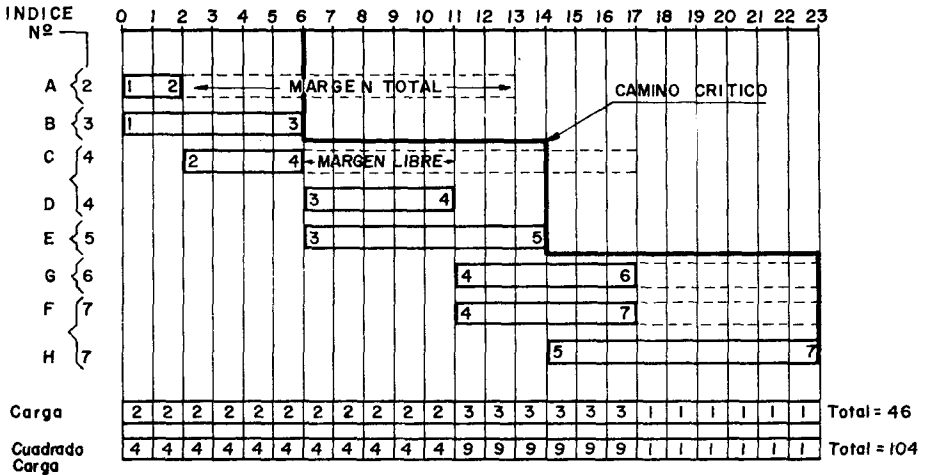


Fig. - 3

nado, se sitúa inmediatamente después de la barra señalada por este mismo índice y asentada más a la derecha.

La figura 3 pone de manifiesto cómo el hecho de haber ordenado la tabla I según los índices de cabeza de las tareas, permite determinar rápidamente la fecha más temprana de comienzo de una actividad determinada. También el plazo mínimo de terminación del trabajo, suponiendo que no existan limitaciones relativas a la carga por intervalo de tiempo.

La carga total de recursos del proyecto se eleva a 46. La carga media se calculará dividiendo este valor por el número de unidades de tiempo. Obtendremos así el valor $46 : 23 = 2$.

Vamos a tratar de reducir el valor 104, correspondiente a la suma de los cuadrados de las cargas diarias, a su valor "ideal" de carga uniforme: $2^2 \times 23 = 92$. Dado que el algoritmo de Burgess-Killebrew es heurístico, puede ocurrir que no se llegue a este valor de carga uniforme. Veamos los resultados que alcanzamos en el proceso de cálculo.

En primer lugar, comenzaremos por desplazar, unidad por unidad, la tarea 4-7. Se trata de la primera en la lista, comenzando por abajo, con margen disponible.

El primer desplazamiento convierte la carga inicial del proyecto en la siguiente:

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

LA PROGRAMACION DE PROYECTOS

La suma de los cuadrados de carga diaria se eleva a 102. Resulta, por tanto, ventajoso este desplazamiento. Siguiendo el procedimiento, podemos comprobar cómo del valor 102 pasamos al valor 100, del valor 100 al 98, de 98 a 96, de 96 a 94 y, por último, del valor 94 al valor 92, para el cual la distribución de carga es la siguiente:

$$2-2$$

Al primer intento hemos alcanzado el alisado óptimo; pero, de no haber sido así, habríamos de pasar a realizar cálculos con las actividades 4-6, tal vez 3-4 —si fueran desplazadas 4-7 y 4-6—, 2-4 y, posiblemente, 1-2, aunque intuitivamente, en este caso, claramente se ve que con el desplazamiento de 4-7, o bien 4-6, a lo largo de 6 unidades de tiempo, habríamos alcanzado un alisado perfecto.

La figura 4 presenta el gráfico de Gantt correspondiente al alisado óptimo.

Puede suceder que los planificadores no queden satisfechos del alisado conseguido partiendo de un programa con tareas dispuestas para su

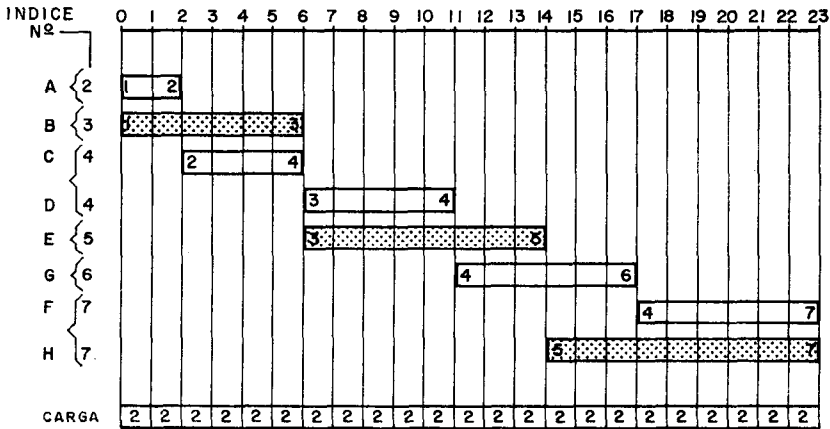


Fig.- 4

comienzo más temprano. No hay que olvidar, como indican sus creadores, que el algoritmo puede generar resultados distintos para un mismo programa. Es suficiente, para ello, partir de distintos planeamientos al comenzar la operatoria. Veamos cómo, programando inicialmente las tareas en sus fechas de comienzo más tardío (fig. 5), se alcanzan los mismos resultados.

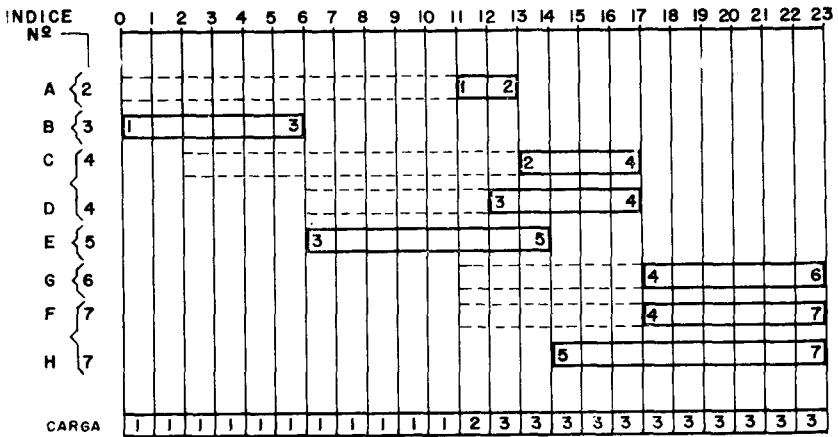


Fig. - 5

El programa de la figura 5 totaliza una suma de cuadrados de carga igual a 114. Hacemos el primer desplazamiento, comenzando, esta vez, por la parte superior del diagrama. Trasladamos el comienzo de la tarea 1-2 a la fecha 0. Reducimos así la suma de cuadrados de carga a 112. A continuación, procedemos de igual modo con las tareas 2-4 y 3-4, cuyo comienzo desplazamos a las fechas 2 y 6, respectivamente. Estos desplazamientos hacen disminuir el sumatorio de los cuadrados de carga a 104. Por último, trasladamos la actividad 4-6 a su fecha de comienzo más temprano.

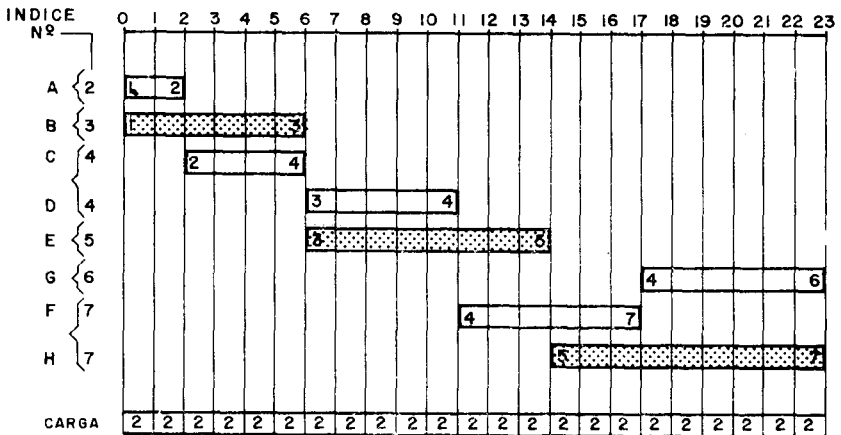


Fig. - 6

Obtenemos así una carga uniforme de valor 2 igual a la lograda anteriormente. La suma de los cuadrados de carga se eleva a 92.

Si en lugar de comenzar el proceso de cálculo por arriba (en la fig. 5), lo hubiéramos iniciado por abajo, no habríamos podido desplazar, en el primer ciclo de cálculos, las tareas 4-7 ó 4-6. En efecto, no se conseguiría con ello reducir la suma de cuadrados de carga. Únicamente serían factibles los desplazamientos de las actividades 3-4, 2-4 y 1-2. En el segundo ciclo de cálculos, y cuando la carga presentara el perfil

2-2-2-2-2-2-2-2-2-2-1-1-1-1-1-1-3-3-3-3-3-3,

comenzaríamos por trasladar la tarea 4-7 a su fecha de comienzo más temprano. Obtendríamos así la solución óptima, cuya nota más destacada está en la disposición de la tarea 4-7 con relación a 4-6, lo que vemos comparando la figura 3 con la 6.

3.2.3. La nivelación en el caso de trabajos repetitivos.

Consideremos ahora el caso de un trabajo en serie representado por un grafo y repetido indefinidamente. En este caso, los gráficos de Gantt

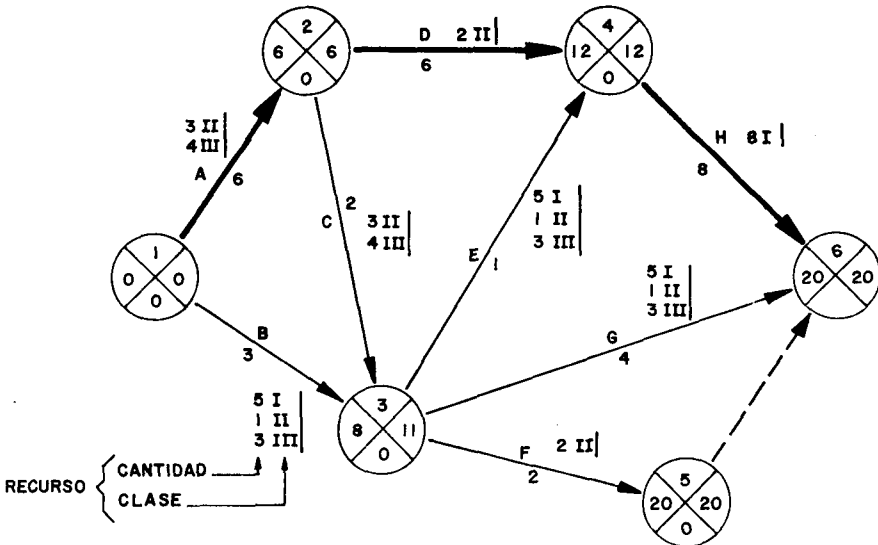


Fig. - 7

constituyen los instrumentos más idóneos para resolver el problema de determinación y nivelación de cargas.

Aparte de la nivelación de recursos, plantearemos nuestro problema suponiendo que el objetivo a alcanzar es la cadencia más rápida, con la restricción de que no se dé comienzo a ninguna actividad antes de que sea concluida la homogénea del proyecto o pieza precedente.

Sea la red de la figura 7. En ella se definen las tareas de una supuesta pieza a fabricar en serie. Como se aprecia, existen tres tipos de recursos distintos. Estos recursos son reclamados en cantidades variables por las actividades, de manera conjunta o independientemente. La prioridad establecida para la nivelación de cada tipo de recurso viene dada por el orden: I, II y III.

Al trazar el gráfico de Gantt, respetando las condiciones señaladas, está claro que el ciclo más corto posible viene dado por la tarea de mayor duración: la actividad H, con un tiempo de 8 unidades. El diagrama de Gantt que nos presenta la figura 8, elaborado a partir de la tabla II, permite representar el de la figura 9. En éste se pone de manifiesto la tipología del problema que estamos tratando, al expresar las relaciones que existen entre ciclos sucesivos.

TABLA II

DESIGNACION TAREA	INDICES		DURACION	FECHA mas TEMP.		FECHA mas TARD.		RECURSOS			MARGEN	
	i	j		INICIO	TERMIN.	INICIO	TERMIN.	I	II	III	TOTAL	LIBRE
A	1	2	6	0	6	0	6	-	3	4	0	0
B	1	3	3	0	3	8	11	5	1	3	8	5
C	2	3	2	6	8	9	11	-	3	4	3	0
D	2	4	6	6	12	6	12	-	2	-	0	0
E	3	4	1	8	9	11	12	5	1	3	3	3
F	3	5	2	8	10	18	20	-	2	-	10	10
G	3	6	4	8	12	16	20	5	1	3	8	8
H	4	6	8	12	20	12	20	8	-	-	0	0

La ventaja del gráfico de la figura 9 es que permite determinar rápidamente las variaciones de la carga de recursos a lo largo del tiempo total de fabricación.

LA PROGRAMACION DE PROYECTOS

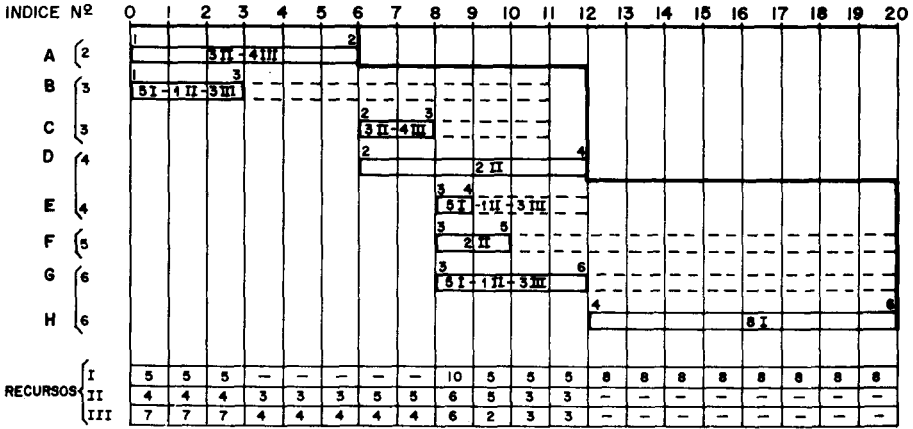


Fig.-8

Con el fin de dar claridad a nuestra exposición, supondremos que el gráfico de la figura 8 está sobre un tambor cuyo perímetro tiene una longitud igual a un ciclo de duración (en este caso 8 unidades de tiempo impuestas por la actividad 4-6), tal como muestra la figura 10.

El gráfico de la figura 10 permite determinar con rapidez la carga por cada intervalo de tiempo; sin embargo, al compararlo con la figura 9, se pone de manifiesto el peor desarrollo de las actividades. Esto obliga a tener que señalar el margen "libre" con un corchete (]) que se coloca en su límite y a indicar marginalmente a cuántas vueltas de tambor se sitúa.

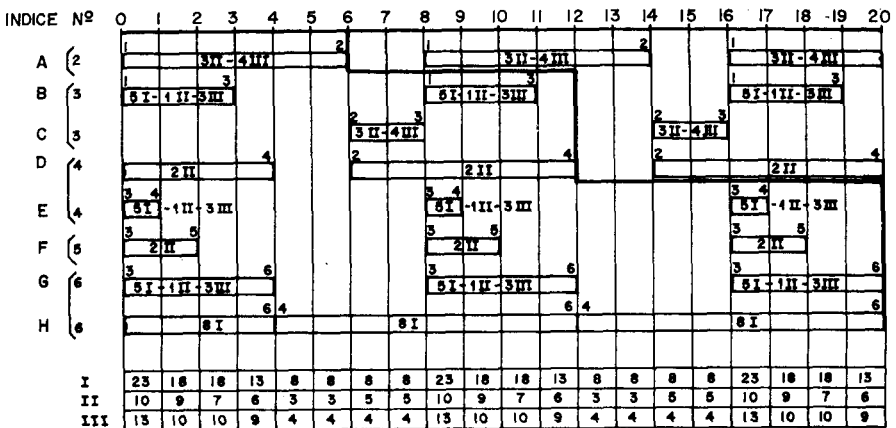


Fig.-9

En los casos complejos, para el cálculo de la carga por intervalo de tiempo y su alisado, se utiliza, como hemos dicho, un ordenador electrónico. Como describen Burgess y Killebrew (11), el principio es el siguiente:

“Para determinar si una actividad está en curso de ejecución o no, en un instante determinado, basta contar el espacio de tiempo que se para este instante del previsto para la terminación de la tarea, dividir el resultado por la duración del ciclo y examinar el resto de la división. Si este resto es inferior a la duración de la tarea, la tarea está en curso, y, no lo está, si el resto es superior a su duración.”

El ordenador opera siguiendo un proceso semejante al descrito en el párrafo anterior.

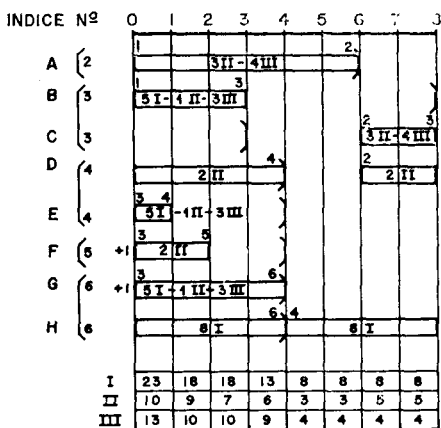


Fig.- 10

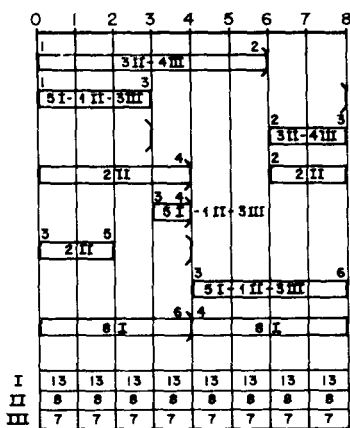


Fig.- 11

Trataremos de nivelar, primeramente, el recurso I, cuya suma de cuadrados de carga se eleva a 1.620 y su carga media a $104 : 8 = 13$. Para ello, y al no poder desplazar la actividad 4-6 (crítica), comenzaremos por la tarea siguiente, con margen disponible y que consume este recurso: la tarea 3-6.

El desplazamiento de 4 unidades de tiempo en la tarea 3-6 nos da el siguiente perfil de carga:

18-13-13-8-13-13-13-13

La suma de cuadrados de carga pasa entonces a 1.402. El desplazamiento

siguiente de la tarea 3-4 en 3 unidades de tiempo reduce a 1.352 la suma de cuadrados. El perfil resultante es ya óptimo:

13-13-13-13-13-13-13-13

La figura 11 presenta el diagrama de Gantt después de este primer proceso de nivelación. Como podemos comprobar, se ha logrado alisar simultáneamente la carga correspondiente al recurso III.

Se nivela a continuación el recurso II. El desplazamiento de 4 unidades de tiempo en la tarea 3-5 permite reducir el sumatorio de carga de 304 unidades a 288. De esta manera se ha logrado un alisado perfecto. (véanse las figs. 11 y 12).

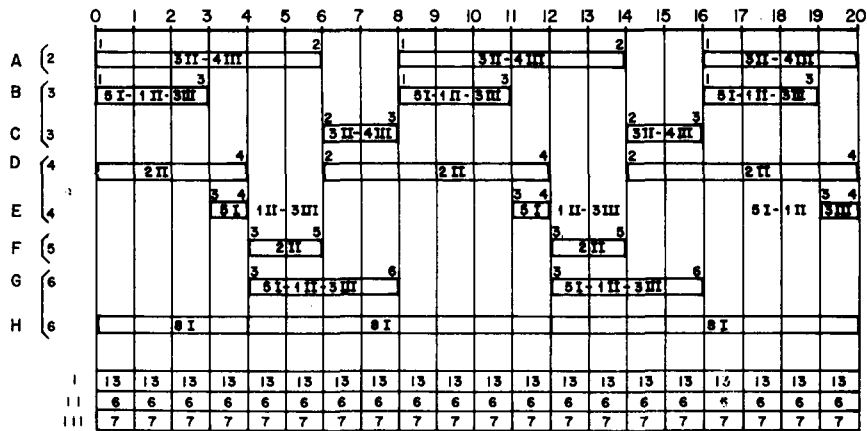


Fig.-12

El problema que acabamos de presentar responde a un tipo particular de programas repetitivos. La nivelación podría complicarse considerablemente al adoptar el problema otras formas más complejas: los programas multiciclos en serie y en paralelo que tratan C. Fisher y G. L. Nemhauser (17).

3.3. OTROS MÉTODOS DE NIVELACIÓN.

3.3.1. Algoritmo de Hughes Corporation.

Un algoritmo de naturaleza heurística ha sido puesto a punto por el Ground Systems Group, de la compañía norteamericana Hughes Corporation.

(17) CAROLYN FISHER and GEORGE L. NEMHAUSER: *Multicycle Project Planning*, "The Journal of Industrial Engineering", vol. XVIII, núm. 4, April, 1967.

El algoritmo de nivelación en paralelo HGSG, al igual que la rutina de Burgess-Killebrew, tiene como objetivo minimizar las fluctuaciones de nivel en los recursos, respetando las restricciones potenciales del problema. Generalmente, el tipo de recurso que se suele tratar es la mano de obra, y, dentro de ésta, aquellas especialidades más críticas.

En el método HGSG, el control de la minimización se efectúa sobre la magnitud absoluta de fluctuación. Sin embargo, cabe la posibilidad de nivelar el programa conforme a ciertos criterios (por ejemplo, los mínimos cuadrados de las fluctuaciones).

3.3.1.1. Parcialización del problema.

El método consiste en dividir el nivel de carga de recursos en intervalos; luego, alisar secuencialmente cada intervalo, revisando, si es preciso, las fechas de comienzo más temprano de las actividades siguientes. El problema, tratado con ordenador electrónico, genera como salida un histograma.

La división del perfil de recursos se realiza determinando aquellos intervalos críticos cuya carga mínima posible rebasa la carga media, y aquéllos en los que la carga máxima posible es inferior a esta carga media.

En los intervalos en donde la carga mínima sea superior a la carga media no debe programarse actividad alguna con margen suficiente para poder ser desplazada a otro lugar.

En los intervalos con carga máxima inferior a la carga media se intentará programar cuantas tareas puedan estar presentes, procurando elegir, entre las tareas de posible programación, aquellas que supongan una menor carga.

El proceso de parcialización en intervalos o subproblemas conduce a la reducción del margen de muchas tareas, lo que hace aparecer nuevas zonas críticas

3.3.1.2. Los límites de los intervalos.

La carga de cada uno de los períodos de tiempo a lo largo de la duración del proyecto, oscila entre dos límites: uno, el límite superior, representa el valor de la carga máxima posible que la programación de tareas puede dar en cada unidad de tiempo; otro, el límite inferior, re-

presenta el valor de la carga mínima posible. Estos dos límites son tenidos en cuenta por el método de determinación de los intervalos de carga.

Los límites superior e inferior pueden representarse gráficamente en forma de histogramas. Revelan así, al ser comparados con la gráfica de la carga media, los intervalos críticos utilizados para descomponer el problema inicial en subproblemas: límite superior bajo el nivel medio y límite inferior sobre este nivel medio.

3.3.1.3. Desplazamiento de tareas.

J. F. Boss (12) describe el algoritmo de desplazamiento de actividades de la siguiente manera:

“Para nivelar un intervalo en donde el límite inferior rebasa el nivel medio del proyecto, se calculan previamente los niveles medios de los intervalos precedente y siguiente. Las tareas son asignadas a estos intervalos siguiendo el orden de su carga y de su duración. Antes de asignar una tarea, es preciso calcular los niveles locales medios que resultan de su asignación a los intervalos precedente y siguiente. La tarea se asigna allí donde resulte un nivel medio más bajo.

La asignación de una tarea a un intervalo dado se obtiene bien sea aumentando su fecha de comienzo más temprano, bien sea disminuyendo su fecha de terminación más tardía. Los valores iniciales de estas fechas, para cada tarea, son dados por los resultados de un cálculo PERT o CPM habitual. Consecuencia lógica de las relaciones de precedencia, un cambio en la fecha de comienzo más temprano o en la de terminación más tardía, para una tarea determinada, entraña, normalmente, las correspondientes modificaciones para todas las actividades ligadas secuencialmente a la tarea en cuestión. Cada asignación, obliga, por consiguiente, a volver a poner al día la lista que resulte del cálculo PERT (existe un subprograma de puesta al día de esta lista a partir de una lista maestra).

Una vez puesta al día la lista maestra, se calculan los nuevos límites que pueden producir nuevos intervalos críticos, que serán a su vez nivelados hasta una carga mínima, asignando sus tareas a derecha e izquierda.

Se continúa el proceso hasta que ya no aparezcan más zonas críticas. En este momento son programadas las partes de la red que corresponden a los intervalos críticos encontrados anteriormente. Entre estos in-

tervalos se encuentran subproblemas o zonas de subredes en donde el límite superior está muy por encima y el límite inferior muy por debajo del nivel medio de mano de obra. Cada una de estas zonas se trata separadamente de la manera siguiente:

El nudo del camino crítico más próximo al centro del intervalo se toma como punto de separación a partir del cual la división a derecha e izquierda se efectúa junto con la puesta al día de la lista maestra y el cálculo de los nuevos límites.

La división continúa hasta que aparecen nuevas zonas críticas y el programa comienza de nuevo el tratamiento del intervalo crítico. El programa puede alternar estos procesos hasta que el subgrafo considerado esté desprovisto de márgenes. El programa trata entonces a la subred siguiente, operando de la misma manera.

Cuando todos los márgenes han sido eliminados de las tareas, el histograma de distribución de la mano de obra es idéntico al límite inferior final (y al superior)".

3.3.2. Nivelación simultánea de varios proyectos: el programa (MS)².

Los norteamericanos F. K. Levy, G. L. Thompson y J. D. Wiest (18), presentan un programa parecido en algunos aspectos a los métodos que hemos descrito. Este programa, denominado (MS)² —“multiship, multishop, workload-smoothing”—, fue diseñado para la nivelación de mano de obra en varios proyectos y varios centros (18^{bis}) de trabajo simultáneamente. Aunque, originalmente fue dirigido a problemas de nivelación de recursos en astilleros navales, se adapta perfectamente a otros problemas de programación industrial.

El (MS)² trata de alcanzar un programa satisfactorio que representa unos costes de mano de obra lo más bajos posible. El número de hombres a emplear está en función de las cargas máximas de trabajo resultantes de la programación de tareas. Por ello, el algoritmo heurístico del (MS)², intenta reducir estas cargas máximas mediante el desplazamiento de las tareas con margen que las producen a periodos con cargas más reducidas. Esta operatoria conduce a una discriminación de los costes por reducción del volumen máximo de personal necesario.

(18) F. K. LEVY, G. L. THOMPSON and J. D. WIEST: *Multiship, multishop, Workload-Smoothing Program*, "Naval Research Logistic Quarterly", March, 1962.

(18 bis) Especialidades.

3.3.2.1. Operatoria del algoritmo.

Para poner en acción el (MS)² es preciso haber elaborado las redes representativas de los distintos proyectos y tener calculadas las fechas más tempranas y más tardías de comienzo y terminación de tareas. Igualmente conocer los márgenes indicativos de la flexibilidad de las actividades susceptibles de desplazamiento, sin perturbar la fecha de terminación del proyecto.

El programa-base se establece con las tareas, correspondientes a los distintos proyectos, ajustadas en sus fechas de comienzo más temprano. A continuación, para cada centro de trabajo se determinan las correspondientes curvas de carga representativas de las necesidades de hombres hasta completar todos los proyectos. El programa fija, en cada centro, “ni-

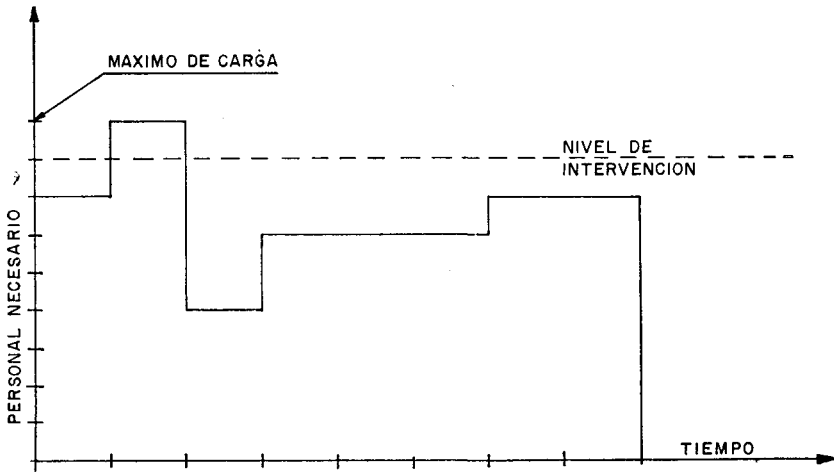


Fig.- 13

veles de intervención” (trigger levels). Estos niveles se sitúan en una unidad (hombre) por debajo del nivel correspondiente a la carga máxima requerida (figura 13).

En cada centro de trabajo se reprograman las tareas que tengan margen para lograr reducir por debajo del nivel de intervención los máximos de carga. Si por el cambio de tareas se consigue que en algún otro intervalo de tiempo de todos los centros de trabajo no sea superado el nivel de intervención, se baja este nivel de intervención otra unidad y se repite el

proceso hasta que cambiando alguna tarea resulta un máximo de carga que rebasa el nivel de intervención. De las tareas con margen que producen el máximo que supera el nivel de intervención se toma una al azar y se desplaza para que comience fuera de este intervalo máximo. Este desplazamiento puede afectar las fechas más tempranas de comienzo y los márgenes totales de las tareas siguientes a la desplazada. Recalculados estos nuevos valores se continúa el proceso de carga hasta que todas las tareas estén cargadas o hasta que el nivel de intervención se rebase y sea preciso desplazar otra tarea.

Cuando en uno o más centros de trabajo se hace imposible reducir los máximos de carga por debajo del nivel de intervención en uno o más proyectos, se retiene el nivel de intervención previo, pasando a continuación a un proceso de nivelación individual de los centros de trabajo. El orden impuesto en el proceso obliga a comenzar por aquel centro que posee el mayor valor medio de pagos por hora de trabajo.

Tras la fase de nivelación simultánea de los centros de trabajo, y ya en la fase de nivelación individual, el programa opera de una manera semejante a la descrita, trabajando con los niveles de intervención, los desplazamientos y las cargas. El punto final se alcanza al no poder reducir más el nivel de intervención de un centro de trabajo.

Es de destacar que este algoritmo, con ligeras variaciones, permite dar varios enfoques a un mismo problema.

El programa (MS)² está dispuesto para ser aplicado empleando un ordenador Bendix G-20.

3.3.3. *Los métodos de Wilson, Hu y Black.*

Más interesantes desde una perspectiva conceptual que por su utilidad práctica, señala E. W. Davis (19), son los métodos de Wilson, Hu y Black.

El método de R. C. Wilson introduce variantes en el (MS)². Trata de conseguir, para una duración determinada del proyecto, el mínimo de recursos a emplear. Un sistema de programación dinámica se incorpora en cada iteración para lograr combinaciones factibles de actividades. El método es menos flexible que el de Levy-Thompson-Wiest y conduce a un complejo proceso de cálculo cuando se abordan problemas con un gran número de tareas.

(19) EDWARD W. DAVIS: *Resource Allocation in Project Network Models — A Survey*, "The Journal of Industrial Engineering", vol. XVII, núm. 4, April, 1966.

Aparte de los cambios introducidos en el método de Levy-Thompson-Wiest, Wilson plantea el problema de nivelación de recursos en redes comparándolo con el de nivelación en líneas de montaje. Existen soluciones empleadas en los problemas de equilibrio en líneas de montaje aplicables a la nivelación en redes, como sucede con la aportación de T. C. Hu. Apoyándose en el teorema que T. C. Hu demuestra para su método de minimización de la duración de un proyecto, denominado "cortar la cola más larga" ("cutting the longest queue"), Wilson muestra cómo reducir el total de recursos necesarios para realizar un proyecto.

Adaptando un algoritmo de equilibrio en línea de montaje, de Gutjahr-Nemhauser, O. J. Black ha elaborado un método de nivelación que da todas las soluciones posibles que satisfacen las restricciones de recurso. El método, aunque particularmente adaptado para el caso de varios recursos, es inadecuado en el tratamiento de grandes redes.

IV. LA COMPATIBILIDAD DE RECURSOS

4.1. CARACTERÍSTICAS DEL PROBLEMA.

Como anteriormente hemos dicho, el otro problema fundamental implicado en la distribución de recursos es el de *compatibilidad*. En este problema se trata de lograr un programa que permita la ejecución de los trabajos bajo la siguientes condiciones:

1. Que el desarrollo del proyecto cumpla las secuencias planeadas, es decir, que se respeten las restricciones potenciales.
2. Que la demanda de recursos no rebase las disponibilidades fijas conocidas, es decir, que se *compatibilice* la carga total de recursos demandados con los disponibles.
3. Que se minimice la duración del proyecto.

Se parte aquí de la hipótesis de que los recursos que resultan ociosos poseen un coste despreciable en parangón con su coste de adquisición.

Para la resolución de este problema existe un variado número de algoritmos. Todos estos algoritmos son eminentemente heurísticos, es decir; tratan de alcanzar soluciones satisfactorias óptimas localmente. La mayor parte de los existentes están programados para ordenador y la información

publicada sobre los mismos es, a veces, meramente descriptiva (programa San Remo) (20). Esto se debe a la especulación del "software", que se está produciendo en el mundo de los ordenadores. En efecto, los explotadores de "software", sean o no fabricantes de ordenadores, guardan celosamente los algoritmos inspiradores de sus programas, cuya publicidad, por otra parte, lanzan abiertamente. No obstante, entre los métodos y programas conocidos podemos citar, por ejemplo: el método presentado por Agard y Gamot en (21), aplicado por Air France; el procedimiento MAP, descrito por R. L. Martino (10); los algoritmos de Gray-Kidd, Spivack, Brand-Meyer-Shaffer y Verhines, citados por Lawrence Pascoe (8); el algoritmo de G. H. Brooks (19); los algoritmos Milord I y Milord II, presentados por R. Descamps y P. Chevignon (9); el algoritmo Spart-I, Spart-II y Grasp (General Resource Allocation and Scheduling Program) (22), y los sistemas Puffs, Astra, Altai, Ramps, Corua, etc.

Algunos problemas de compatibilidad pueden ser tratados con los algoritmos descritos para el problema de nivelación. Tal es el caso, sobre todo, de la rutina de Burgess-Killebrew, la cual puede dar soluciones muy satisfactorias en el caso de programas con restricciones de recursos elevadas.

En adelante nos limitaremos a exponer los métodos y programas que creamos más adecuados para dar una panorámica de las distintas facetas del problema que estamos abordando bajo la hipótesis de distintas condicionantes.

4.2. PROGRAMACIÓN DE UN PROYECTO ÚNICO CON INTENSIDAD FIJA: EL PROGRAMA ROC 8001.

El Roc 8001 fue puesto a punto por la gran empresa norteamericana Richfield Oil Company. Se trata de un método de asignación en serie especialmente adaptado a problemas muy complejos. Este método está programado para ordenador en lenguaje científico Fortran II y IV.

El criterio de asignación que inspira el método es el siguiente:

(20) JEAN FRANCOTTE: *Approche d'une méthode de repartition de la main-d'oeuvre dans l'ordonnement de grands projets industriels*, "Organisation Scientifique", núm. 3, 1968.

(21) J. AGARD et G. GAMOT: *Ordonnement du grand entretien d'un avion avec egalisation des effectifs d'entretien*, "Revue Française de Recherche Operationnelle", núm. 38, 1966.

(22) JEROME D. WIEST: *A heuristic model for Scheduling large projects with limited Resources*, "Management Science", vol. 13, núm. 6, February, 1967.

1. Los recursos se asignarán prioritariamente a las tareas críticas;
2. Los recursos sobrantes de la aplicación anterior se destinan a las actividades no críticas, haciendo uso de los márgenes libres;
3. Siempre que sea necesario y posible se acude a la programación intermitente de actividades;
4. Cuando los niveles de carga no sean satisfactorios se aumenta la duración del proyecto y se repite el proceso.

4.2.1. *Desarrollo del método.*

El punto de partida en el que se apoya el Roc 8001 es un programa CPM o PERT. A continuación, se pasa al proceso de distribución de recursos en el ordenador.

Para el control del desarrollo se supone que existe un reloj imaginario que se desplaza en el tiempo período tras período.

Una vez ordenadas las tareas por sus fechas de comienzo más temprano, el algoritmo opera de la siguiente manera:

1. Poner el reloj a cero (fecha inicial del proyecto y comienzo de los cálculos).

2. Leer la lista de actividades, ordenadas según sus fechas crecientes de comienzo más temprano, hasta que estas fechas sobrepasen la indicada por el reloj. Simultáneamente con la lectura, las tareas se inscriben en una lista de espera. Las reglas de prioridad respetadas en esta lista son las siguientes:

- a) Las tareas de la lista pueden comenzar si únicamente se consideran las restricciones temporales, ya que todas sus fecha de comienzo más temprano son anteriores a la que indica el reloj.

- b) Las tareas se ordenan en la lista según sus fechas de comienzo más tardío, puesto que se convierten en críticas si no han podido comenzar antes de esa fecha.

Se debe empezar por aquellas actividades que son susceptibles de retraso. El orden más adecuado es, por tanto, el de los márgenes totales crecientes. Indica esto que entre varias actividades de igual fecha de inicio más temprano será prioritaria la que disponga de menor margen total.

- c) En el caso de que las actividades tengan igual fecha de comienzo más tardío, se dará preferencia a la de menor duración.

3. Ordenar el comienzo de las tareas críticas (pasar de la lista de

espera a una lista de trabajo) si es posible, es decir, si todas las tareas precedentes han sido terminadas y si las tareas en curso, que han comenzado antes de la fecha actual del reloj, dejan disponibles los recursos en cantidad y clase necesarios.

4. Si no hay recursos suficientes y la fase anterior no puede ser superada:

a) Retardar el comienzo de ciertas tareas en curso (pasándolas a la lista de espera), una vez examinados sus márgenes, a fin de que se liberen los recursos necesarios en cantidad suficiente para las tareas críticas del momento presente. El orden de retraso debe ser iniciado por las tareas menos críticas. En caso de indeterminación se da preferencia a la que le falte más tiempo para ser terminada.

b) Cuando la medida anterior no fuese suficiente, se procede a interrumpir actividades no críticas que se encuentren en la lista de trabajo, de modo que no se rebase la fecha de su terminación más tardía. En caso de indeterminación entre varias tareas, se interrumpirá primeramente la última en ser comenzada.

Si con el retraso y la interrupción de tareas no críticas no se logran los recursos suficientes para las tareas críticas, el proyecto habrá de ser retrasado. En este caso, se aumentarán en una unidad de tiempo todas las fechas de comienzo más tardío de la lista de tareas y se comenzará de nuevo.

5. Ordenar, si es posible, el comienzo en sus fechas más tempranas de las tareas no críticas en lista de espera.

6. Indicar el final de las tareas acabadas, borrándolas de la lista de trabajo y devolviendo los recursos que utilizaban al "stock" de recursos disponibles.

7. Avanzar la fecha del reloj en un período y volver a la fase 2.

4.2.2. *Desarrollo de un ejemplo.*

Partiremos del programa representado en la figura 14.

Elaboramos a continuación el diagrama calendario de la figura 15. En este gráfico indicamos con un número sobre cada tarea la cantidad de unidades técnicas necesarias para su ejecución. También, en números romanos, la clase de recursos reclamados. La programación de tareas se ha hecho en las fechas más tempranas de iniciación. Como podemos ob-

LA PROGRAMACION DE PROYECTOS

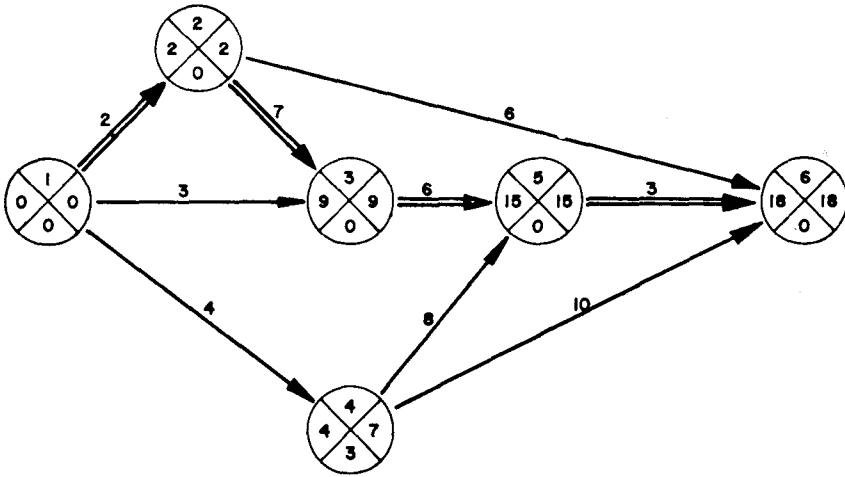


Fig-14

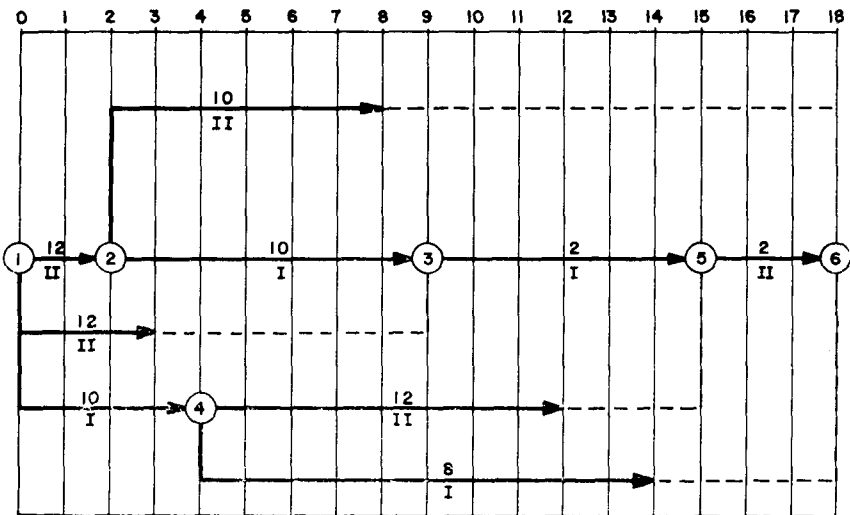


Fig-15

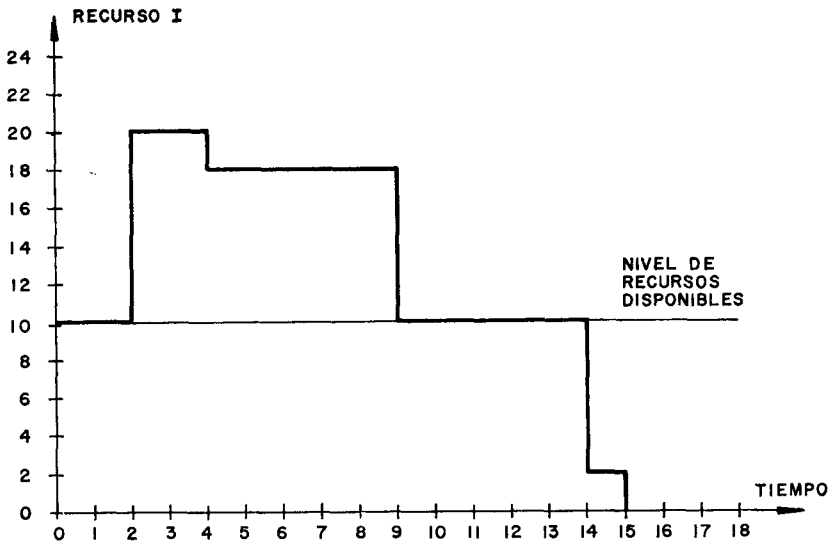


Fig-16

servar, en las gráficas de las figuras 16 y 17, los límites de recursos I y II disponibles han sido rebasados por el programa. Procede, pues, resolver este problema de compatibilidad utilizando el algoritmo que tratamos.

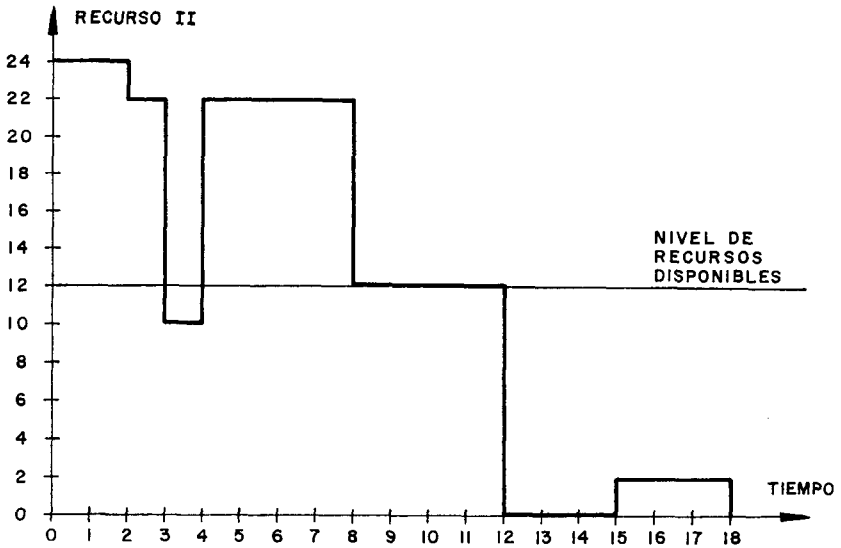


Fig-17

TABLA III

ACTIVIDAD	DURACION	FECHA MAS TEMPRANA DE COMIENZO	FECHA MAS TARDIA DE COMIENZO	RECURSOS		TAREAS PRECEDENTES
				I	II	
1 - 2	2	0	0	-	12	-
1 - 4	4	0	3	10	-	-
1 - 3	3	0	6	-	12	-
2 - 3	7	2	2	10	-	1 - 2
2 - 6	6	2	12	-	10	1 - 2
4 - 5	8	4	7	-	12	1 - 4
4 - 6	10	4	8	8	-	1 - 4
3 - 5	6	9	9	2	-	1 - 3, 2 - 3
5 - 6	3	15	15	-	2	3 - 5, 4 - 5

Para ello elaboramos la tabla III, en la cual aparecen las tareas ordenadas por sus fechas de comienzo más temprano, dentro de este orden por su fecha más tardía y, luego, por su duración.

La aplicación manual del algoritmo nos lleva a la elaboración de la tabla IV, semejante a la presentada por R. Companys en (23 y 24). Esta tabla, en el caso de un proyecto complejo, se obtendría del ordenador. La tabla recoge en cada una de sus columnas la siguiente información:

K.1.: Reloj (H).

K.2.: Lista de espera. Recoge las tareas que por no poder entrar en la lista de trabajo deben esperar. También las tareas que, estando en ejecución, son interrumpidas.

K.2.1.: Descripción de la actividad.

K.2.2.: Duración de la actividad (t_{ij})

K.2.3.: Fecha de comienzo más temprano ($T_E(i)$)

K.2.4.: Fecha de comienzo más tardío ($T_L(i)$)

K.2.5.: Fecha crítica (T_c)

(23) RAMÓN COMPANYS: *Programación y control de proyectos con recursos limitados*, "Estudios Empresariales", núm. 25, 1965.

(24) RAMÓN COMPANYS: *Planificación de proyectos*, Asociación para el Progreso de la Dirección, Madrid, 1966.

TABLA IV

4 RELOJ H	2 LISTA DE ESPERA								3 LISTA DE TRABAJO								4 Stock Recursos RESIDUALES		5 ACTIVIDADES TERMINADES		
	ACTIV.	DURAC.	T _E (i)	T _L (i)	F _c	κ	RECURSOS		ACTIV. PRECED.	ACTIV.	DURAC.	T _E (i)	T _L (i)	T _p	κ	RECURSOS		ACTIV. PRECED.	I	II	
							I	II								I	II				
0	1-3	3	0	0	0	0	-	12	-	1-2 1-4	2 4	0 0	0 4	2 4	0 0	- 10	12 -	- -	-	-	
2	1-4 2-6	4 6	0 2	0 2	0 2	2 0	10 -	- 10	- 1-2	1-3 2-3	3 7	0 2	0 2	5 9	0 0	- 10	12 -	- 1-2	-	-	1-2
4	1-4 2-6 4-5 4-6	4 6 8 10	0 2 4 4	0 2 8 8	0 2 8 8	2 0 0 0	10 - 10 12	- 10 - 1-4	- 1-2 1-4 1-4	1-3 2-3	3 7	0 2	0 2	5 9	2 2	- 10	12 -	- 1-2	-	-	
5	2-3 4-5 4-6	7 8 10	2 4 4	2 8 8	2 8 8	3 0 0	10 - 12	- 1-4 1-4	1-2 1-4 1-4	2-6 1-4	6 4	2 0	2 0	11 7	0 2	- 10	10 -	1-2 -	-	2	1-3
7	2-6 4-6	6 10	2 4	2 8	2 8	2 0	- 10	10 -	1-2 1-4	2-3 4-5	7 8	2 4	2 8	11 15	3 0	10 -	1-2 1-4	-	-	1-4	
9	2-6 3-5 4-6	6 6 10	2 9 4	2 9 8	2 9 8	2 0 0	10 - 8	10 - 2-3,1-3	1-2 1-4	2-3 4-5	7 8	2 4	2 8	11 15	5 2	10 -	1-2 1-4	-	-		
11	2-6	6	2	2	2	2	-	10	1-2	4-6 4-5 3-5	10 8 6	4 4 9	8 15 9	21 15 17	0 4 0	8 12 2	1-4 1-4 2-3,1-3	-	-	2-3	
15	5-6	3	15	15	15	0	-	2	3-5	2-6 3-5 4-6	6 6 10	2 9 4	2 9 8	19 17 21	2 4 4	- 2 8	10 2-3,1-2 1-4	-	2	4-5	
17										2-6 5-6 4-6	6 3 10	2 15 4	2 15 8	19 20 21	4 0 6	- 2 8	10 3-5,4-5 1-4	2	-	3-5	
19										5-6 4-6	3 10	15 4	15 8	20 21	2 8	- -	2 1-4	3-5,4-5	2	10	2-7
20										4-6	10	4	8	21	9	8	-	1-4	2	12	6-7
21																		10	12	4-7	

$$T_c = T_{E(i)} + \alpha$$

- K.2.6.: Unidades de tiempo ya consumidas en la ejecución de la actividad (α).
- K.2.7.: Recursos reclamados por cada tarea, expresados en unidades técnicas.
- K.2.8.: Actividades precedentes.
- K.3.: Lista de trabajo. Recoge las tareas en ejecución. Su estructura es análoga a la columna 2 (K.2), excepto:
- K.3.5.: Fecha prevista de terminación T_i
- $$T_i = H + t_{ij} - \alpha$$
- K.4.: "Stock" de recursos residuales. Recoge las unidades de recursos no consumidos por las tareas en curso de trabajo.
- K.5.: Tareas acabadas.

La tabla IV ha sido elaborada de la siguiente manera:

Reloj a cero.—La primera actividad que entra en la lista de trabajo es 1-2, tarea crítica que comienza en la fecha 0. Esta tarea no deja libre ninguna unidad del recurso II. A continuación, siguiendo la tabla III, viene la actividad 1-4, cuya fecha más temprana de comienzo es también 0. Hacemos entrar esta tarea en la lista de trabajo, agotando así las disponibilidades del recurso I. La siguiente tarea en la tabla es 1-3, pero, al carecer de disponibilidades de recurso II, pasa a la lista de espera. No queda más por hacer y avanzamos el reloj una unidad de tiempo.

Reloj en 1.—No pasa absolutamente nada. Continúa la ejecución de las tareas 1-2 y 1-4, mientras 1-3 espera entrar. El número de unidades disponibles de los recursos I y II es 0. El reloj pasa a 2.

Reloj en 2.—Suceden varias cosas. Se termina la tarea 1-2, que sale de la lista de trabajo dejando libres 12 unidades del recurso II. Este recurso es reclamado, en este momento, por las actividades 1-3 (en espera) y 2-6, listas para comenzar; pero, la insuficiente cantidad disponible del recurso II nos obliga a poner en la lista de espera la tarea 2-6 (margen 10) y en la lista de trabajo la actividad 1-3 (en este momento con margen 4).

Dos tareas se disputan el recurso I: 1-4 (en lista de trabajo y con margen 3) y 2-3 (crítica). Interrumpimos 1-4 (considerando que es tecnológicamente posible), pasándola a la lista de espera y comenzamos la ejecución de 2-3.

El reloj continúa avanzando y las tareas van entrando en la lista de trabajo tal como se puede ver en la tabla IV.

Como podemos apreciar en la figura 18 el algoritmo ha conseguido compatibilizar los recursos necesarios con los disponibles, pero eso ha supuesto el retraso de tres días en la duración inicial prevista para la realización del proyecto.

Comprobamos, en la figura 18, la programación intermitente de las actividades 1-4 y 2-6. Si no fuera tecnológicamente posible esta programación, entonces tendríamos que haber evitado, en la fecha 2, la suspen-

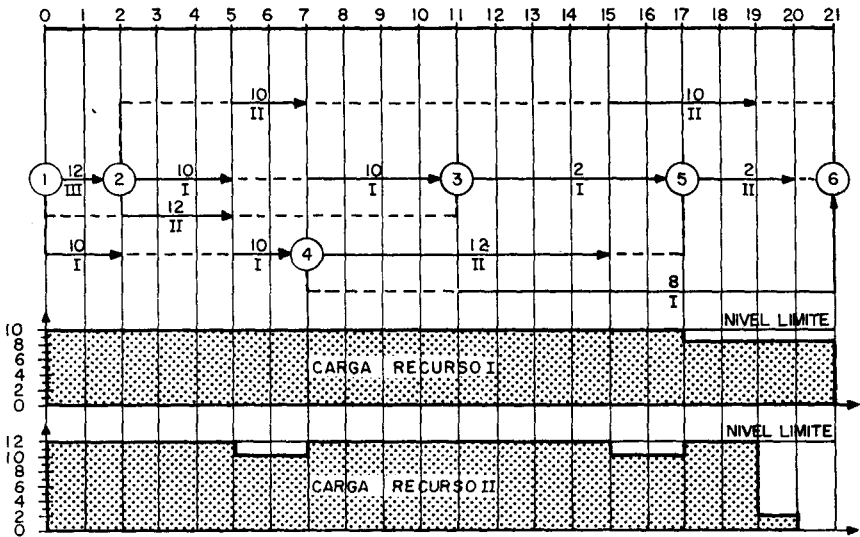


Fig.- 18

sión de los trabajos de la tarea 1-4 y hacer entrar en lista de espera la actividad 2-3. La figura 19 muestra los resultados que obtendríamos en este último caso. Como vemos, también tendríamos que retrasar 3 unidades de tiempo la duración inicial del proyecto.

4.3. PROGRAMACIÓN SIMULTÁNEA DE VARIOS PROYECTOS CON INTENSIDAD FIJA: LA TÉCNICA ALTAI.

Altai es el anagrama de "Análisi, Livellamento e Tempificazione Automatici e Integrati", que significa análisis, nivelación y coordinación automática e integrada.

LA PROGRAMACION DE PROYECTOS

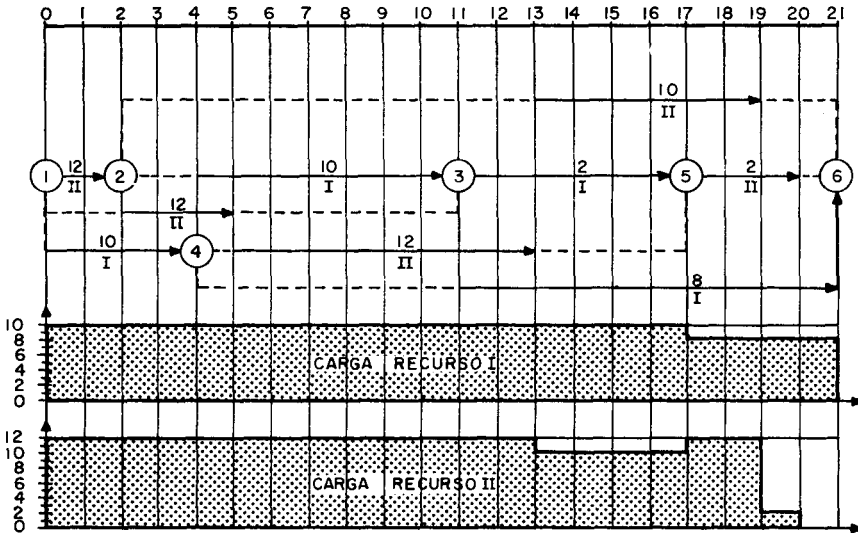


Fig.- 19

El sistema ALTAI, original de IBM Italia, acomete el problema de compatibilidad de recursos a intensidades fijas para el caso de varios proyectos a realizar simultáneamente. Para nuestra exposición nos basaremos en los trabajos de dos especialistas destacados: M. Ricciardi (25 y 26) y A. Carducci (27).

La técnica ALTAI emplea para consecución de sus objetivos el instrumental electrónico. Su objetivo fundamental es minimizar:

$$C_{(\min)} = C_r + C_s$$

en donde:

C_r = coste total de los retrasos en la terminación de los proyectos.

C_s = costes por sobrecarga y subcarga en los recursos.

En el caso que tratamos —varios proyectos iguales o distintos— se podría recurrir al empleo de una red única, comprensiva de todos los proyectos; pero esto significaría tener que resolver a continuación el gra-

(25) MARIANO RICCIARDI: *ALTAI, técnica di programmazione della produzione con più commesse contemporanee*, IBM, Italia, 1963.

(26) MARIANO RICCIARDI: *Il PERT l'ALTAI e le altre tecniche reticolari di programmazione*, ETAS KOMPASS, Milano, 1965.

(27) CARDUCCI ARTEMISIO: *PAC (PERT-ALTAI-COSTE) 1440, una procedura per la tempificazione, la valorizzazione e lo scapionamento delle fasi di lavorazione di più commesse contemporanee*, IBM, Italia, 1964.

voso problema de una compleja programación, predeterminando el orden de ejecución de numerosísimas tareas de proyectos distintos.

En el ALTAI no hay límites teóricos, ni en el número de proyectos dispuestos en el tiempo, ni en el número de tareas de cada proyecto, ni en el número de centros de trabajo de la empresa. Existe un límite práctico que viene dado por la capacidad del ordenador electrónico empleado; pero este límite se puede superar empleando memorias supletorias en discos magnéticos, o bien, ordenadores más potentes.

En ALTAI cada proyecto se representa por una red PERT. Por cada red se perforan, aparte de las tarjetas de control, tantas otras como actividades la componen.

Si el número de proyectos a programar es elevado, el número de redes a elaborar se suele reducir en la práctica. Esto se consigue reuniendo los proyectos en grupos más o menos formalmente semejantes.

La fecha de comienzo para cada proyecto se puede establecer en base a la fecha de terminación deseada. También a la duración del proyecto y a un margen de seguridad.

La intensidad de cada tarea se programa de manera que su rendimiento sea máximo, teniendo en cuenta el conjunto de tareas que por término medio estarán en ejecución en el mismo departamento.

En las redes PERT elaboradas para cada proyecto se determinan los caminos críticos. Estos caminos, junto con los cálculos de tiempos y márgenes, constituyen el punto de partida para el tratamiento con ALTAI. La figura 20 describe el proceso total realizado por ALTAI.

4.3.1. *Flexibilidad de ALTAI.*

ALTAI permite, resuelve y se adapta a:

1. Los casos de graduación de prioridades entre proyectos simultáneos. Para ello se define para cada proyecto un coeficiente de prioridad relativa. Este coeficiente hace que en el tratamiento con ALTAI sean comenzadas, en primer lugar, aquellas actividades que posean un coeficiente de prioridad relativa más alto.

2. Los casos de pedidos no retrasables más allá de sus fechas de entrega. Para ello está prevista la posibilidad de establecer códigos de pedidos intocables, en los que sólo pueden ser retrasadas las tareas dentro de los límites permitidos por sus márgenes. Si fuera preciso para

LA PROGRAMACION DE PROYECTOS

DIAGRAMA DEL PROCESO "ALTA I"

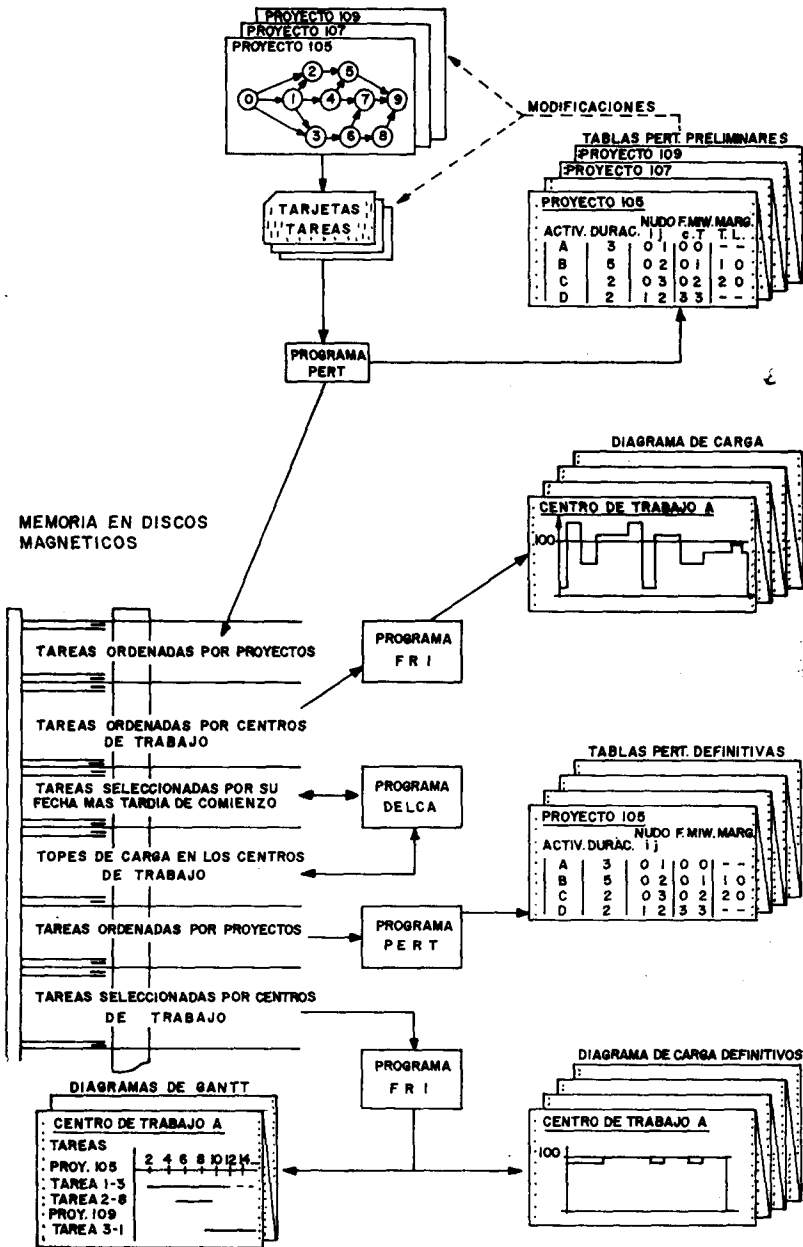


Fig.- 20

mantener inalteradas las fechas de terminación de estos pedidos, se puede recurrir a modificar los coeficientes de prioridad relativa o incluso a contratas o ayudas exteriores.

3. La puesta al día periódica, merced a la conservación en cintas magnéticas de todas las redes de los proyectos.

4. Las variaciones de las duraciones previstas para las tareas futuras.

5. Los casos extraordinarios, tales como roturas de máquinas, accidentes, etc.

6. Introducción de nuevos proyectos.

7. Eliminación de proyectos en un programa.

8. Eliminación de tareas acabadas.

9. Variaciones en la fuerza nominal.

10. Variaciones en el porcentaje medio de ausencias.

11. Variaciones en los límites de carga.

12. Control de las redes.

Tras la exposición que acabamos de realizar, vamos a tratar de dar una versión práctica aproximada de las distintas fases operativas de ALTAI. Para ello nos apoyaremos en una serie de gráficos y tablas que, aunque no son exactas a las obtenidas de un ordenador, se acercan bastante en su estructura. No obstante, conviene exponer previamente una serie de conceptos.

4.3.2. *Conceptos fundamentales.*

— *Pedidos*: Este término es utilizado por ALTAI como sinónimo de proyectos, prestaciones, construcciones, etc.

— *Centro de trabajo*: Una empresa industrial está compuesta por cierto número de secciones. Cada una de estas secciones puede comprender varios puestos de trabajo. Para referirse indistintamente a unos y otros se emplea la expresión que consideramos.

— *Fuerza nominal*: Es el número nominal de unidades de trabajo (hombres) conocidas, ya existentes o hipotéticas.

— *Fuerza presente*: Es el resultado de disminuir la fuerza nominal en un porcentaje medio de ausencias.

— *Fuerza extraordinaria*: Es el número de unidades laborales aplicadas a las actividades para dilatar las restricciones impuestas por las disponibilidades normales. Esto se consigue:

LA PROGRAMACION DE PROYECTOS

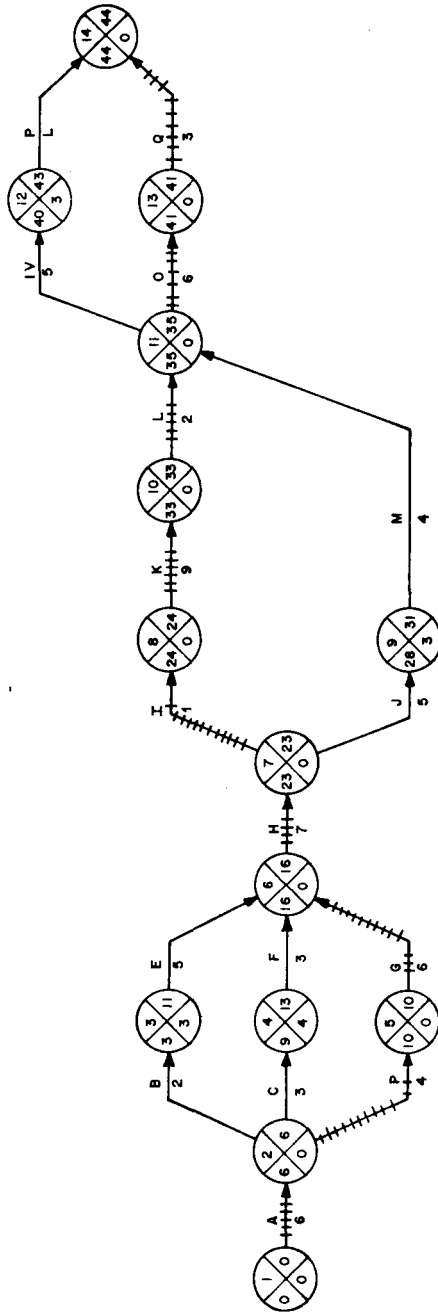


Fig.-21

- Aumentando la fuerza presente y manteniendo constante el número de cada unidad (horas).
- Aumentando el número de horas semanales de trabajo en cada unidad y manteniendo constante la fuerza presente.
- Aumentando la fuerza presente y el número de horas semanales de cada unidad.

4.3.3. *Operatoria de ALTAI. Desarrollo de un ejemplo.*

Partiendo de una serie de proyectos más o menos semejantes al de la figura 21, comenzaremos las siguientes fases:

FASE I:

En la primera fase de tratamiento electrónico se consigue, empleando un programa PERT, coordinar todas las actividades de todos los proyectos, alimentar con información la memoria de discos magnéticos y obtener la primera serie de tablas.

Las tablas, generadas como resultado por el ordenador, presentan la siguiente estructura:

a) *Encabezamiento:*

- Fecha final calculada (para el proyecto que se considere).
- Fecha final prefijada.
- Lista con descripción del proyecto.

b) *Cuerpo:*

- Marginalmente, tantas filas como tareas.
- Por columnas:
 - Descripción de las tareas.
 - Siglas del centro de trabajo donde será ejecutada.
 - Código del puesto de trabajo.
 - Duración.
 - Etapa inicial y etapa final.
 - Resultados del cálculo:
 - Fecha de comienzo.

LA PROGRAMACION DE PROYECTOS

- Fecha de terminación.
- Holguras (libre y de interferencia).
- Señalizaciones de criticidad de etapas finales (.) y tareas (.-.).

TABLA V

" PROYECTO NUMERO 6 "					Fecha final calculada 129 Fecha final prefijada					
VAR.	ACTIVIDAD				ETAPA		FECH. MINIM.		HOLGURA	
DURAC.	DESCRIP.	REPT.	PUESTO	DURAC.	INIC.	TERM.	INIC.	TERM.	LIBRE	INTERF.
	Fecha Inicio			85		1.		85		
	A	TEC.	12	6.-	1	2.	85	91		
	B	EST.		2	2	3	91	94		3
	C	PRO.	7	3	2	4	91	94		4
	D	FRE.	4	4.-	2	5.	91	95		
	E	CER.	3	5	3	6.	93	98	3	
	F	PRO.	7	3	4	6.	94	97	4	
	G	TEC.	12	6.-	5	6.	95	101		
	H	PRO.	7	7.-	6	7.	101	108		
	I	FRE.	4	1.-	7	8.	108	109		
	J	TOR.	2	5	7	9	108	113		3
	K	EST.		9.-	8	10.	109	118		
	M	CER.	3	4	9	11.	113	117	3	
	L	MON.	5	2.-	10	11.	118	120		
	N	TOR.	2	5	11	12	120	125		3
	O	TEC.	12	6.-	11	13.	120	126		
	P	PRO.	7	1	12	14.	125	126	3	
	Q	MON.	5	3.-	13	14.	126	129		

.-. ACTIVIDAD CRITICA
 . MARGEN NULO

FASE II:

Las actividades de cada proyecto se seleccionan por centros de trabajo. Hecho esto se verifica un control gráfico de las diversas cargas de trabajo mediante el empleo del programa FRI. Las tablas generadas por el empleo de programa presentan la siguiente estructura:

- a) *Encabezamiento*: Recoge las características del centro de trabajo.
- Código.
 - Número de unidades laborales nominales.
 - Fuerza presente (considerando el porcentaje medio de ausencias).
 - Número de máquinas (eventualmente).
- b) *Fila 1*: Esta fila alcanza los 25 períodos que componen cada exploración.
- c) *Fila 2*: Correspondiendo con cada período, se consignan las unidades laborales necesarias.
- d) *Fila 3*: Cada uno de los valores de cada período, recogidos en la fila 2, se resta de la fuerza presente. Estas diferencias constituyen los valores de esta fila.
- e) *Fila 4*: Recoge el porcentaje de la fuerza necesaria con respecto a la fuerza presente.
- f) *Diagrama*: En el cuerpo principal de la tabla se da una versión cartesiana de las cuatro filas anteriores.

El eje de abscisas recoge los 25 períodos de tiempo, y el de ordenadas es común a tres escalas:

- Escala 1.^a (de la derecha): Compromisos porcentuales de la fuerza presente para valores crecientes, de 5 en 5.
- Escala 2.^a: Horas semanales de trabajo.
- Escala 3.^a: Horas semanales totales del centro de trabajo correspondiente a los varios niveles de compromiso.

En el caso de querer explorar los 25 períodos siguientes, bastaría con repetir la exploración.

FASE III:

Esta fase constituye la parte más importante del ALTAI.

El programa DELCA entra en acción, ejecutando las siguientes operaciones:

LA PROGRAMACION DE PROYECTOS

PUESTO DE TRABAJO

CODIGO - FUERZA NOMINAL - FUERZA PRESENTE - N° MAQ.
 5 22 20

105 106 108 110 112 114 116 118 120 122 124 126 128 130

ESPECIALIS. NECESARIOS..... 20 20 20 20 24 24 24 26 30 24 18 18 18 30 22 40 12 12 22 6 35 8 31 12 30 20
 DIF. CON FZA. PRESENTE - - - - 4 -4 -4 -6 -10 -4 2 2 2 -10 -2 -20 8 8 -2 14 -15 12 11 8 -10 -
 PORCENTAJE FZA. PRESENTE..... 100 71 00 100 100 120 120 130 150 120 90 90 90 50 110 200 80 60 110 30 75 40 155 60 150 00

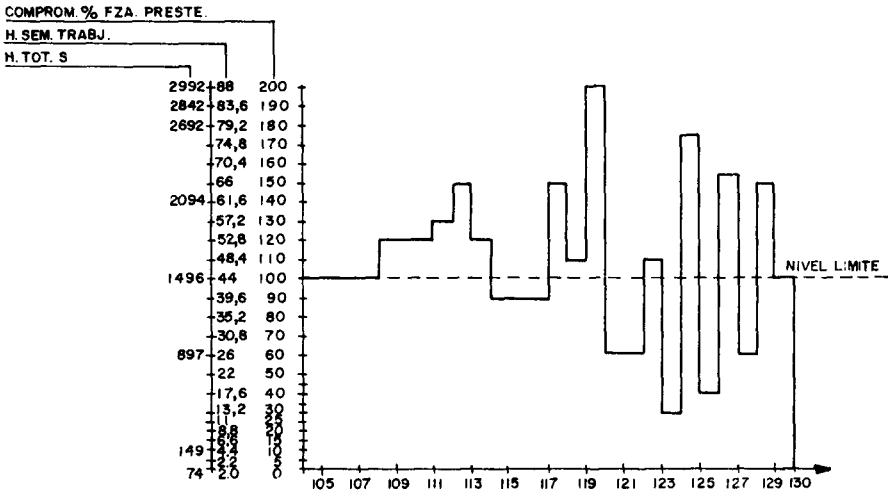


Fig.- 22

- 1.º Verifica la posibilidad de asignar a los distintos centros de trabajo las tareas que sucesivamente son consideradas. Esta operación de exploración secuencial se realiza siguiendo una tabla en donde las tareas han sido ordenadas por sus fechas crecientes de comienzo tardío.
- 2.º Controla, y si es posible sin variación, la carga de los distintos centros de trabajo, a partir de la fecha de comienzo más temprano. En caso de necesidad, retrasa las tareas hasta encontrar una nueva fecha de comienzo más temprano, que permite compatibilizar las cargas de trabajo preexistentes, retocando eventualmente las fechas de comienzo de todas las tareas que parten de la etapa final examinada.

FASE IV:

Una vez que las tareas respetan todas las limitaciones a que están sometidas, se reagrupan por proyectos; a continuación, y para cada uno de éstos, se efectúa un cálculo definitivo con programa PERT. Las tablas que se obtienen son semejantes en su estructura a las descritas en la fase I; la única diferencia estriba en que aparece una nueva columna (la primera), que recoge la variación en las duraciones totales de las tareas.

La variación en las duraciones de algunas tareas se debe al retraso a que se han tenido que someter para poder respetar la compatibilidad con los precedentes compromisos de trabajo. Este retraso, sumado a su duración efectiva de ejecución (que no es variada), da la nueva duración total de la tarea (tiempo de espera + tiempo de ejecución).

Comparando, para cada proyecto, las tablas PERT resultantes en la fase I con las obtenidas en esta fase IV, se podrán observar las diferencias producidas en las fechas de terminación.

FASE V:

En esta fase las tareas son ordenadas por centros de trabajo a tenor de los códigos preestablecidos para éstos. Merced a la aplicación del programa FRI se pueden obtener las tablas de cargas de trabajo por cada centro, semejantes a las descritas en la fase II, si bien en éstas aparecen respetados los niveles máximos de disponibilidades.

El programa FRI genera también los correspondientes diagramas de Gantt de las tareas de cada centro de trabajo, a ejecutar en los 25 períodos que se examinan en cada secuencia de programación. La estructura de estos diagramas es la siguiente:

a) *Encabezamiento*:

1. Descripción del centro de trabajo.
2. Escala de tiempos.

b) *Cuerpo*:

1. a) Descripción de la tarea.
- b) Fecha de comienzo.
- c) Holgura de interferencia.

LA PROGRAMACION DE PROYECTOS

TABLA VI

" PROYECTO NUMERO 6 "					Fecha final calculada 133 Fecha final prefijada					
VAR.	ACTIVIDAD				ETAPA		FECH. MINIM.		HOLGURA	
DURAC.	DESCRIP.	REPT.	PUESTO	DURAC.	INIC.	TERM.	INIC.	TER.	LIBRE	INTERF.
	Fecha Inicio			85		1.		85		
	A	TEC.	12	6	1	2.	85	91		
	B	EST.		2	2	3	91	94		2
	C	PRO.	7	3	2	4	91	94		4
	D	FRE.	4	4	2	5.	91	95		
6	E	CER.	3	5	3	6.	94	101	2	
	F	PRO.	7	3	4	6.	94	101	4	
	G	TEC.	12	6	5	6.	95	101		
	H	PRO.	7	7	6	7.	101	108		
	I	FRE.	4	1	7	8.	108	109		
	J	TOR.	2	5	7	9	108	116		3
	K	EST.		9	8	10.	109	118		
7	M	CER.	3	4	9	11.	116	124	3	
	L	MON.	5	2	10	11.	118	124		
9	N	TOR.	2	5	11	12	124	131		1
10	O	TEC.	12	6	11	13.	124	130		
7	P	PRO.	7	1	12	14.	131	133	1	
7	Q	MON.	5	3	13	14.	130	133		

- ACTIVIDAD CRITICA
- MARGEN NULO

2. a) Código del proyecto.
- b) Código del reparto.
- c) Número de unidades de trabajo comprometidas.
- d) Fecha de terminación.
- e) Holgura libre.

c) A la derecha de los datos b) se realiza una indicación gráfica para cada tarea.

4.4. PROGRAMACIÓN SIMULTÁNEA DE VARIOS PROYECTOS CON INTENSIDADES VARIABLES.

La consideración de varias intensidades en la realización de las actividades, convierte la programación simultánea de proyectos en un enor-

me problema combinatorio. Las tareas pasan a ser definidas por la cantidad de trabajo. La noción de camino crítico desaparece, y, en tanto no se fija la duración de las tareas dentro de su intervalo de variación, la duración total de los distintos proyectos es indeterminada. Una serie de métodos de naturaleza heurística se han desarrollado. Todos ellos renuncian a la obtención del óptimo y se limitan a la búsqueda de una solución aceptable desde una perspectiva de orden práctico.

4.4.1. *El método de McGee-Markarian.*

A. A. McGee y M. D. Markarian, investigadores de IBM, han elaborado un método para resolver el problema que consideramos en el caso

PUESTO DE TRABAJO

CODIGO - FUERZA NOMINAL -- FUERZA PRESENTE - Nº MAQ.
5 22 20
105 106 108 110 112 114 116 118 120 122 124 126 128 130 +-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
ESPECIALIS. NECESARIOS..... 20 20 20 20 18 18 18 18 18 20 20 20 18 20 12 12 18 18 18 18 20 20 18 18
DIF. CON FZA. PRESENTE..... - - - - 2 2 2 2 2 2 - - - - 2 - 6 6 2 2 2 2 2 - - 2 2
PORCENTAJE FZA. PRESENTE..... 100 100 100 100 90 90 90 90 90 100 100 100 90 90 60 60 90 90 90 90 100 100 90 90

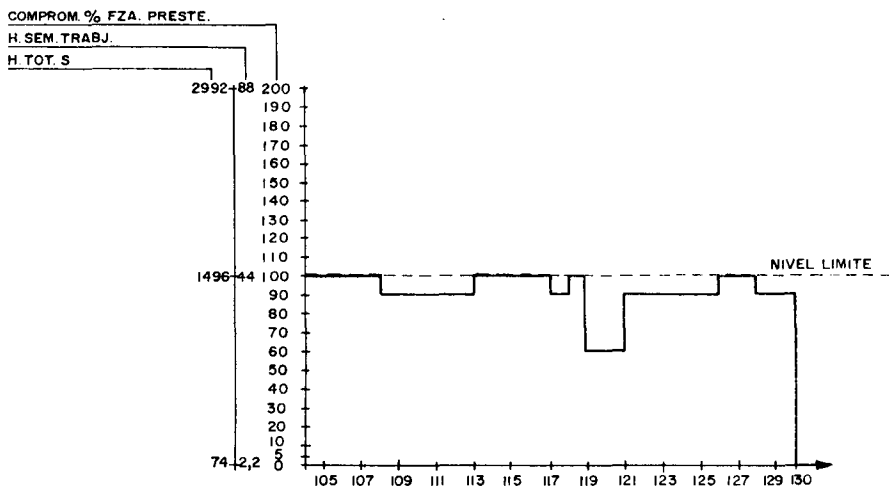


Fig.- 23

DIAGRAMA DE GANTT Y CONTROL DE LAS CARGAS DE TRABAJO

PUESTO DE TRABAJO

CODIGO - FUERZA NOMINAL - FUERZA PRESENTE - N° MAQ.
 7 44 40

DESCRIP. ACT.- INIC - H.INTERF.
 COD.PROY- COD.REP.- N°-TER-ML

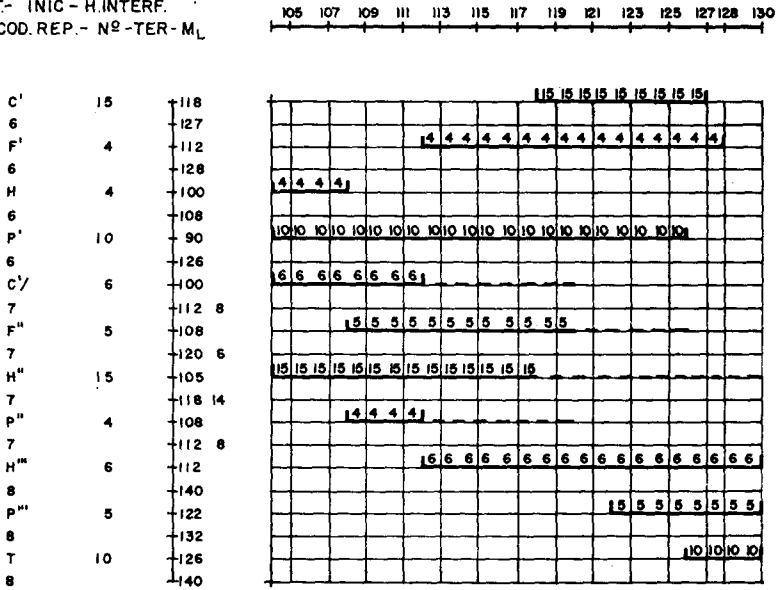


Fig.- 24

de un único recurso: la mano de obra. Su estudio tomó como base el trabajo de redacción de programas para ordenador. Originalmente, el recurso analizado era el programador indiferenciado. La idea motriz fue resolver el problema de limitación de recursos alargando ciertas actividades por reducción de los recursos asignados.

Para cada tarea establece una relación lineal recurso-duración. Esto implica que McGee-Markarian no razonan en "cantidad de trabajo" constante, ya que en este caso, la relación hubiera sido hiperbólica. El punto mínimo de recursos (minimum essential effort), corresponde a una sola persona o grupo; el máximo (crash effort), es aquel en el que la gente se entorpecería al trabajar.

El algoritmo descrito por McGee-Markarian opera de la siguiente manera:

Conociendo las limitaciones para cada intervalo de tiempo, se comienza por elaborar un programa asignando en cada tarea el nivel mínimo "esencial" de recursos. Se calcula el camino crítico y las fechas más tempranas y más tardías de comienzo y terminación de tareas. Se establecen las curvas de carga y se comparan con las de disponibilidades, determinando el exceso de defecto de personal. Si con el programa inicial se rebasan los límites de disponibilidades, se procede a desplazar las tareas con margen de aquellos intervalos en que se producen sobrecargas. Si no se consigue respetar las limitaciones habrá que buscar recursos suple-

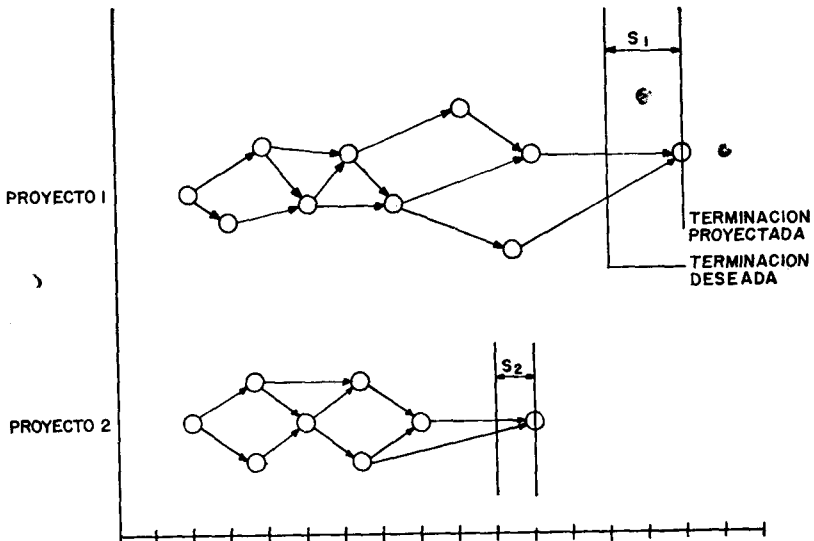


Fig-25

mentarios (al no poder descender más abajo el nivel mínimo de asignación), o bien el proyecto habrá de ser retrasado. Si el nivel de disponibilidades se respeta, se comprueba si la duración del proyecto se encuentra dentro del plazo de terminación planeado. Si no se encuentra dentro de este plazo, se asignan incrementos sucesivos del recurso sobrante hasta reducir el plazo de terminación del proyecto a la fecha deseada. Esta asignación se hace a las tareas del camino crítico en las que la reducción en tiempo por unidad (hombre) añadida es mayor.

En el caso de varios proyectos simultáneos se comienza por establecer, para cada proyecto, su correspondiente programa de asignación mínima. A continuación, y para cada proyecto, se calcula el intervalo:

$$S = T_p - T_d$$

en donde,

T_p = Fecha proyectada de terminación del proyecto.

T_d = Fecha deseada de terminación del proyecto.

De los distintos valores de S_i calculados se toma el mayor. Al proyecto correspondiente a este valor máximo se le hace la primera aplicación de recursos. Se calcula luego su nueva fecha de terminación proyectada y su correspondiente valor S . Comparados de nuevo los valores S_i , se toma el mayor y se repite el proceso hasta que se alcancen las fechas deseadas de terminación o bien el recurso se haya aplicado en su totalidad.

Se extraña E. W. Davis (19) de que McGee y Markarian (o incluso otros) no hayan publicado todavía el programa de cálculo que en su publicación original decían tener "en desarrollo", pese a los avances que el método parece representar en el campo de los modelos tiempo-coste. También, McGee-Markarian terminaban su informe diciendo que desarrollarían su algoritmo para el caso de n recursos. Que sepamos, hasta la fecha no lo han hecho, pero ya J. F. Boss (28) lo apuntaba diciendo que el problema es teóricamente muy difícil como se comprueba consultando las razones que da Bernard Roy (3).

4.4.2. *El programa CORUA*

M. Doligez (29) presenta un programa de cálculo electrónico escrito en lenguaje Fortrán IV, denominado CORUA. Se trata de un programa muy rápido (20 proyectos, 1.150 tareas y 20 categorías de recursos = 120/1.000 horas en ordenador), y, en su nueva versión, de una gran capacidad (7.000 tareas). Mediante un algoritmo heurístico el programa busca una solución tan próxima al óptimo como sea posible. Esta solución debe garantizar la compatibilidad entre los recursos necesarios y los disponibles, reduciendo

(28) J. F. BOSS: *Los métodos del camino crítico y las ligaduras de tipo acumulativo*, Coloquio Hispano-Francés sobre Métodos Modernos de Gestión, Barcelona, 1964.

(29) M. DOLIGEZ: *La resolution des problèmes de programmation simultanée de plusieurs projets avec limitation de capacité: Le programme de calcul électronique CORUA*, Gestion, Mai, 1966.

do el plazo total de ejecución de los distintos proyectos. En el proceso se tiene muy en cuenta el grado de urgencia de cada proyecto.

Con el fin de asignar los distintos recursos disponibles de la mejor manera posible, el algoritmo trabaja efectuando una simulación en el tiempo. En cada período realiza una selección de las tareas que deberán ser comenzadas, continuadas, interrumpidas o reducidas, en base a un criterio que tiene en cuenta: la duración de las tareas, el grado de urgencia, el margen del proyecto al que pertenecen y el margen dentro de su proyecto. La situación en que se encuentra cada tarea se recoge en una serie de listas a través de las cuales se desarrolla la operatoria.

El programa es muy flexible. Aparte de considerar varios proyectos y varias categorías de recursos, introduce para cada actividad la posibilidad de variar su intensidad. También considerar sus grados de prioridad e interrupción de acuerdo con los códigos establecidos al efecto.

Para cada una de las clases de recursos se define un techo de disponibilidades. Este techo corresponde al punto máximo de la curva de disponibilidades y pretende controlar las curvas reales de recursos disponibles.

Los recursos comprendidos entre el techo y la curva de disponibilidades se supone que son consumidos por un proyecto ficticio P_0 , con un código de superprioridad 0 (figura 26).

Los proyectos son ligados a un nudo común inicial y otro final por medio de tareas ficticias (figura 27). Esta operación permite dar flexibilidad a los márgenes de los proyectos reales cuando se asignan valores a las tareas ficticias. El proyecto P_0 cumple aquí la misión de reglar el conjunto.

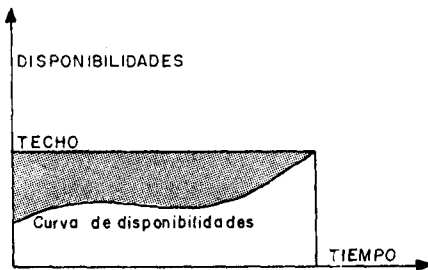


Fig-26

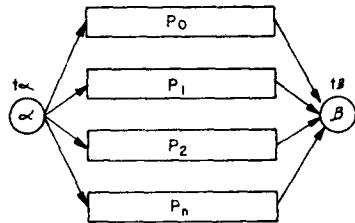


Fig-27

LA PROGRAMACION DE PROYECTOS

DIAGRAMA DEL PROCESO CORUA

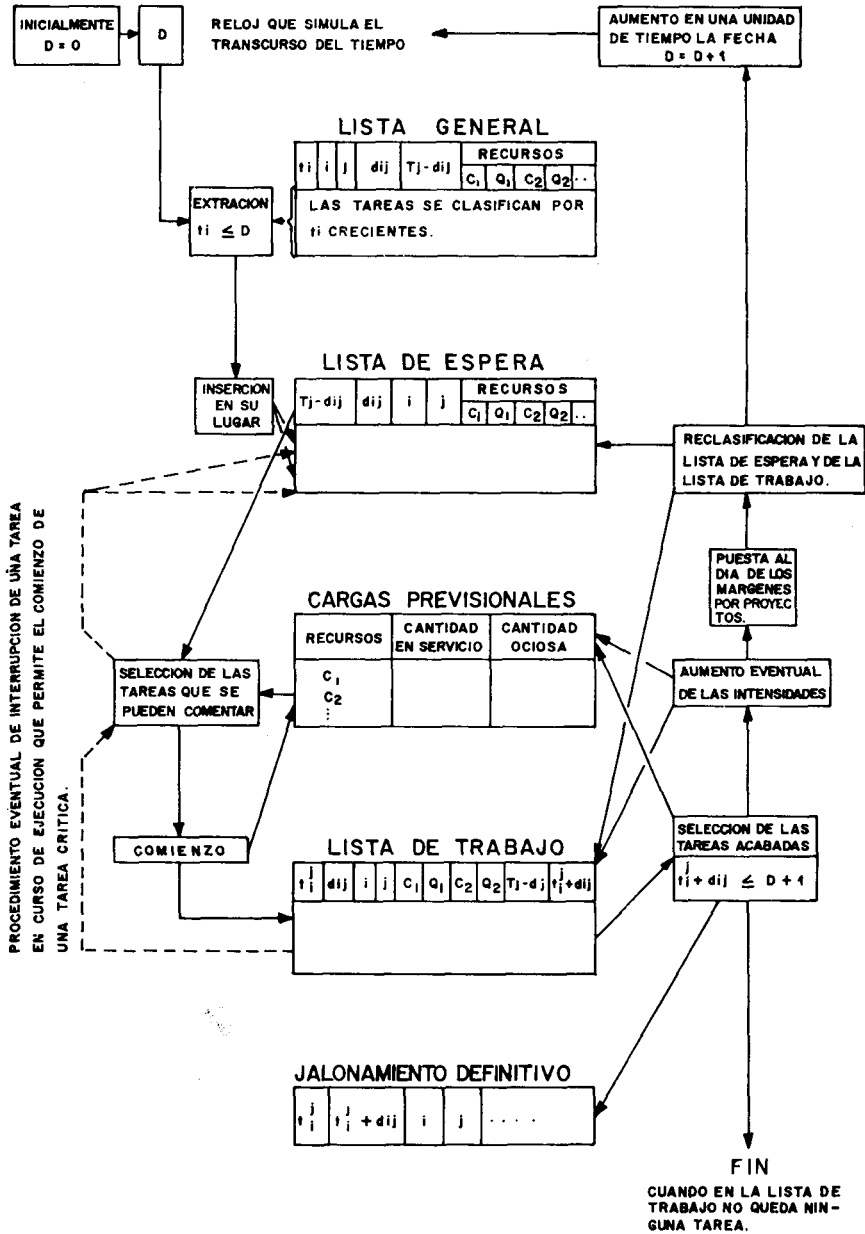


Fig.- 28

4.4.2.1. Operatoria de CORUA.

La figura 28 presenta un diagrama simplificado que describe el funcionamiento de CORUA.

Las actividades en la lista de espera se clasifican por su código de prioridad (de 0 a 9) y su margen disponible. Esta clasificación, resultado de una ley muy compleja, determina el orden de introducción en la lista de trabajo.

El orden de clasificación en la lista de trabajo es el mismo de la lista de espera. Este orden sirve de base para la aplicación de una ley de interrupciones. En efecto, cuando por falta de recursos no se puede comenzar alguna tarea, se consulta la lista de trabajo de abajo arriba, siguiendo el código de interrupciones, hasta liberar los necesarios interrumpiendo una tarea. Si fuera imposible liberar estos recursos de la lista de trabajo, el programa comprueba la posibilidad de comenzar la tarea reduciendo su intensidad. Si para la nueva intensidad reducida son suficientes los recursos disponibles, la tarea será comenzada. En caso contrario, seguirá en la lista de espera.

El ordenador facilita como resultados del proceso diversos listados que facilitan un informe detallado y completo de los distintos programas.

4.4.3. *El método RAMPS*

RAMPS (Resource Allocation and Multi-Proyect Scheduling) es un sistema elaborado para la distribución de recursos y la programación simultánea de proyectos que emplea el cálculo electrónico (30, 31, 32 y 33). Originalmente, fue desarrollada por su actual compañía explotadora, C-E-I-R Inc. (filial de Control Data Corporation), para la empresa norteamericana Dupont de Nemours. RAMPS fue inicialmente programado para ordenador IBM 7090, con una capacidad de 700 tareas, 60 recursos y seis proyectos simultáneamente. A pesar de que no conocemos los detalles sobre el algoritmo que utiliza el sistema (ya hemos mencionado el problema existente en torno a la explotación del software), basándonos en

(30) JACK MOSHMAN, JACOB JOHNSON and MADALYN LARSEN: *RAMPS, A Technique for Resource Allocation and Multi-Proyect Scheduling*, Proceedings 1963, Spring Joint Computer Conference.

(31) *RAMPS, An introduction*, C-E-I-R Inc., Washington.

(32) *RAMPS, User Guide*, C-E-I-R Inc.

(33) F. ROSS: *RAMPS, Resource Allocation and Multiproyect Scheduling*, Gestion, Mai, 1964.

la documentación reseñada, daremos una somera descripción del mismo.

Ya hemos hecho referencia a la naturaleza combinatoria del problema que tratamos; pues bien, la base de apoyo de RAMPS es emplear un modelo combinatorio, pero buscando una optimización local. Se intenta lograr únicamente una solución satisfactoria. Para ello el problema se divide en segmentos, se optimizan localmente los segmentos y se agrupa el conjunto de los óptimos (28).

4.4.3.1. Fases en la aplicación del sistema.

Como se ve en la figura 29, el sistema RAMPS se aplica en tres fases:

FASE I:

En esta fase se comienza por elaborar redes detalladas de cada uno de los proyectos a realizar. Se calculan las fechas de comienzo y terminación de los distintos proyectos y se especifican las penalidades corres-

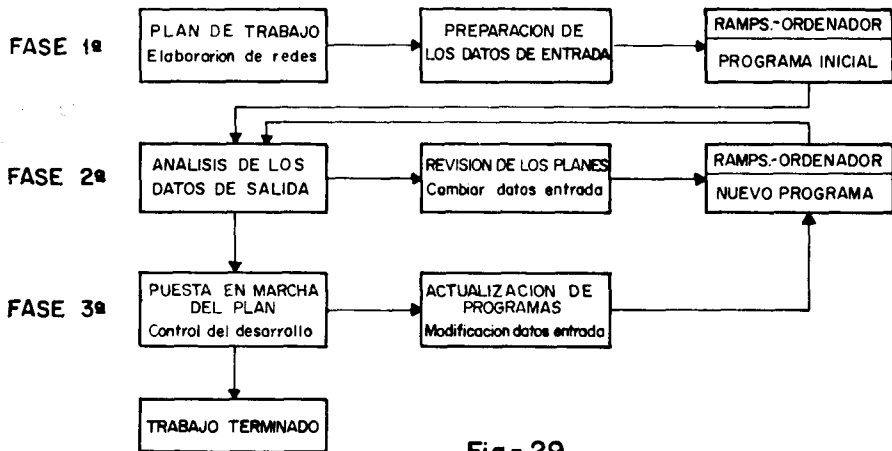


Fig- 29

pondientes a las posibles demoras. En cada proyecto se estudia minuciosamente cada actividad. Con ello se pretende determinar los recursos necesarios para la ejecución de las actividades, sus intensidades, las fechas estimadas de terminación para cada intensidad e incluso, si se desea, la penalidad de interrupción para cada tarea. A continuación, para todos los proyectos, se determina el volumen total de recursos necesarios. Una relación para cada tipo de recursos es elaborada. En esta relación se espe-

cifican: el volumen de disponibilidades, los períodos en que se encuentran disponibles los recursos, su coste unitario, etc. El último eslabón de esta fase hace acopio de la información anterior, la procesa en el ordenador y genera como resultado un plan inicial de desarrollo de los proyectos.

FASE II:

En esta fase se realiza un análisis y valoración de los programas generados en la fase anterior. Si a juicio de la dirección los programas obtenidos son satisfactorios, se ordena su puesta en marcha. En caso contrario, se buscan los puntos flacos que posteriormente se han de someter a un proceso de corrección. Esta corrección obliga a modificar los datos de entrada en el ordenador y a generar nuevos programas. Estos programas contienen ya las modificaciones efectuadas en los planes iniciales. Las etapas de esta fase se repiten cuantas veces sea necesario hasta alcanzar un plan factible. Este plan será el centro de atención en la fase siguiente.

FASE III:

La primera etapa de esta fase utiliza los programas obtenidos para poner en marcha el plan, dar comienzo a los trabajos, reclamar los recursos necesarios y establecer el control del desarrollo. Se pasa luego a una etapa de actualización. En ella se recogen todas cuantas desviaciones se han ido produciendo en la ejecución de los trabajos. La información de entrada original es modificada y se incluyen las variaciones oportunas: cambios de intensidades, comienzo de nuevos proyectos, eliminación de tareas y proyectos, aumento y disminución de recursos, modificación de fechas y penalidades, etc. Luego se realiza una nueva programación. Esta programación vuelve a ser analizada a fin de comprobar los efectos que los desarrollos realizados hasta ese momento pueden producir sobre los trabajos pendientes. Las etapas de las fases II y III se repiten hasta la terminación de los proyectos.

4.4.3.2. La información en las redes.

El punto de apoyo del sistema RAMPS son las redes. Estas constituyen el soporte sobre el cual RAMPS acumula una cuantiosa información que luego ha de ser tratada en el ordenador. La red es, por consiguiente,

LA PROGRAMACION DE PROYECTOS

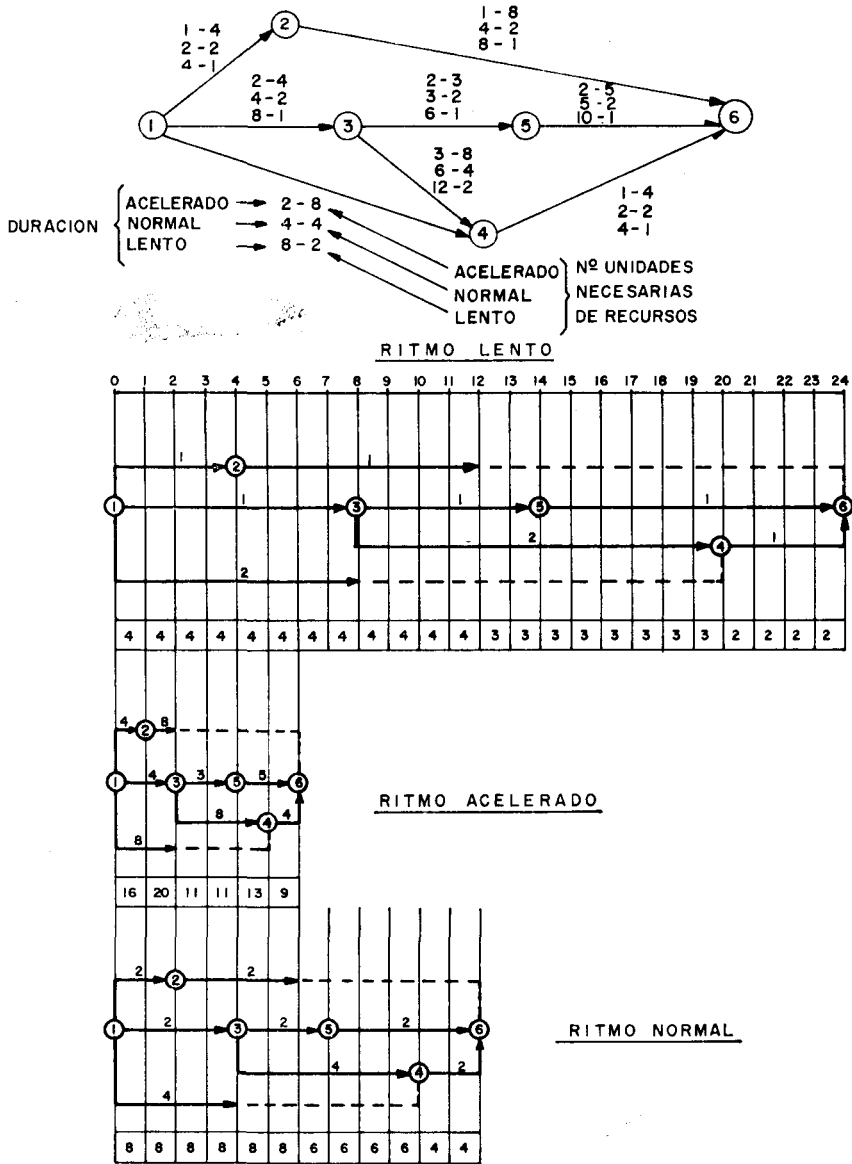


Fig- 30

la imagen de las secuencias tecnológicas de cada proyecto y su centro de información.

Objetivo importante de RAMPS es la estimación exacta o aproximada del tiempo y de los recursos necesarios para el cumplimiento de cada tarea. Una mala estimación de estos valores conduciría los programas a un seguro fracaso.

Elemento fundamental en RAMPS es la *cantidad de trabajo* necesaria para ejecutar cada tarea. Sobre las redes elaboradas se especifican, una vez calculados, los valores determinantes de esta magnitud: la duración, la intensidad y el rendimiento (este último cuando difiere del valor 1).

Empleando la noción de cantidad de trabajo, RAMPS define para cada tarea tres intensidades: lenta, normal y acelerada. Al desarrollar el proceso de programación empleando estos tres ritmos de trabajo se consigue una flexibilidad excepcional. Esta flexibilidad de cada tarea se hace extensiva a cada proyecto. En efecto, los proyectos, según los plazos que le son impuestos, pueden hacer uso de tres ritmos de trabajo que posiblemente le permitirán respetar los compromisos contraídos. La figura 30 ilustra el caso de un proyecto planeado a tres ritmos de trabajo. Cada ritmo, como se comprueba, supone duraciones y cargas distintas. La combinación de las intensidades con los márgenes disponibles pone al sistema en condiciones de generar programas eficientes. Estos programas consiguen los recursos necesarios para la ejecución de las tareas, minimizan la duración de cada proyecto y respetan el nivel de los recursos disponibles.

Cuando son varios los tipos de recursos reclamados por las tareas, RAMPS define el "equipo de recursos". Con ello se pretende mantener el concepto de cantidad de trabajo. El "equipo de recursos" es una combinación de los recursos que intervienen en la ejecución de una tarea. Está formado por un recurso "principal" (el de mayor demanda) y uno o varios recursos secundarios. El recurso principal es el que interviene prácticamente en la definición de la cantidad de trabajo.

Cuando es imposible respetar los niveles de recursos disponibles, RAMPS recurre al empleo de recursos adicionales. Esta medida sólo en casos excepcionales es adoptada.

4.4.3.3. Establecimiento de objetivos y controles de dirección.

A pesar de las muchas restricciones que pesan sobre los proyectos, RAMPS tiene que valorar muchos factores para poder conseguir una pro-

gramación satisfactoria de los mismos. La dirección necesita establecer, a veces, objetivos determinados que deberán ser alcanzados. RAMPS hace entrar estos objetivos en su proceso operativo a base de una serie de parámetros que son incluidos con la información de entrada en el ordenador.

Al afrontar la programación de varios proyectos simultáneos, es posible que se puedan seleccionar ciertos proyectos cuya terminación producirá mayores beneficios a la empresa. Esto induce a que se les conceda un carácter prioritario sobre los demás, en cuanto a los recursos limitados que reclaman. Las fechas de comienzo del proyecto, las fechas de terminación y las penalidades por retrasos son datos que determinan, al ser introducidos en el proceso calculatorio, la importancia relativa de los proyectos. Igualmente los costes originados por demoras pueden ser considerados para establecer proyectos prioritarios.

Aparte de los datos anteriores, utilizados para expresar la prioridad entre proyectos, también pueden ser considerados otros. RAMPS procura constantemente satisfacer los siguientes objetivos:

1. Empezar y terminar cada tarea lo más pronto posible (minimización de la duración total).
2. Asegurar un ritmo uniforme en la ejecución de los trabajos y en el empleo de los recursos (previsión de estrangulamientos).
3. Minimizar el coste total.
4. Minimizar los recursos ociosos.
5. Maximizar el número de las tareas en ejecución simultánea.
6. Dar prioridad a las tareas críticas.
7. Evitar las interrupciones de tareas en ejecución.

Muchos de estos objetivos pueden ser incompatibles y pueden crear problemas conflictivos. Para facilitar la elección, RAMPS define los *parámetros de control* (Control Factors). Estos parámetros son indicativos del orden de prioridad entre los distintos objetivos. Los parámetros cuyos pesos relativos influyen en las decisiones de programación que se adoptan en RAMPS, son los siguientes: margen total, margen libre, previsión, continuidad del trabajo, número de tareas programadas y recursos ociosos.

4.4.3.4. El empleo de los parámetros de control.

Los parámetros de control se emplean en la programación de cada período de tiempo. Su peso es de una gran importancia en las decisiones de programación y en el control directo de los programas generados. Los cuatro primeros parámetros son utilizados en la valoración de las tareas individuales que reclaman un mismo recurso en el período de tiempo que se programa. Los otros dos parámetros intervienen en la evaluación de las distintas asignaciones de recursos del período. Veamos, a continuación, los efectos que producen sobre la programación y los objetivos de estos parámetros.

Margen total

Este parámetro tiene en cuenta, particularmente, aquellas tareas que no disponen de margen total o bien cuentan con uno muy reducido. Cuanto menor es el margen total de una tarea, mayor será el peso asignado al parámetro considerado. Con ello se pretende conseguir un efecto prioritario en la saturación de los recursos reclamados por las tareas críticas. El objetivo perseguido con la intervención del parámetro *margen total* es minimizar la duración total del proyecto.

Margen libre.

Se trata de conceder aquí la prioridad a aquellas tareas que carecen de margen libre o disponen de él en cantidad muy reducida. El parámetro margen libre cumple una doble finalidad: reduce la duración del proyecto al dar prioridad a las tareas críticas y acelera las tareas con reducido margen de interferencia evitando posibles atascos. Al conceder una gran importancia a las tareas con elevada propensión a la criticidad, este parámetro constituye un medio eficaz para combatir las demoras en los casos en que se precisa una pronta terminación de los proyectos.

Previsión

El objetivo de este parámetro es evitar que se produzcan atascos por amontonamiento en el trabajo e interrupciones en la ejecución de las tareas. Se trata de conceder carácter prioritario a aquellas tareas de las

cuales dependen muchas otras. Este parámetro es muy poderoso para generar programas que precisan de una duración mínima. El efecto global es el empleo máximo de los recursos.

Continuidad en el trabajo

Con el fin de evitar la interrupción de tareas, cuyas consecuencias se traducirían en una elevación importante de los costes, el parámetro *continuidad en el trabajo* concede prioridad a aquellas tareas cuya interrupción no es deseable.

Número de tareas

Intenta este parámetro maximizar el número de tareas programadas durante cada período de tiempo. El empleo en la programación de la intensidad lenta, distribuyendo los recursos entre muchas actividades, cumple aquella finalidad. El empleo del parámetro que consideramos pretende lograr un desarrollo continuado y extensivo de los trabajos, aunque para ello haya que sacrificar la velocidad de ejecución de los proyectos.

Recursos ociosos

Se destaca aquí la importancia de mantener ocupados al máximo los recursos disponibles. El objetivo perseguido es minimizar el volumen de los recursos ociosos.

4.4.3.5. Resultados.

Como se aprecia en el diagrama de la figura 31, RAMPS genera una serie de informes como resultado de un proceso operativo con un programa específico en un ordenador. Los informes principales son:

1. *Programas de trabajo.*

Para cada proyecto la impresora del ordenador facilita un programa de trabajo. Cada uno de estos programas contiene exclusivamente la información que se relaciona con su proyecto. Marginalmente, identificadas por sus nudos i-j, se relacionan las tareas que componen el proyecto.

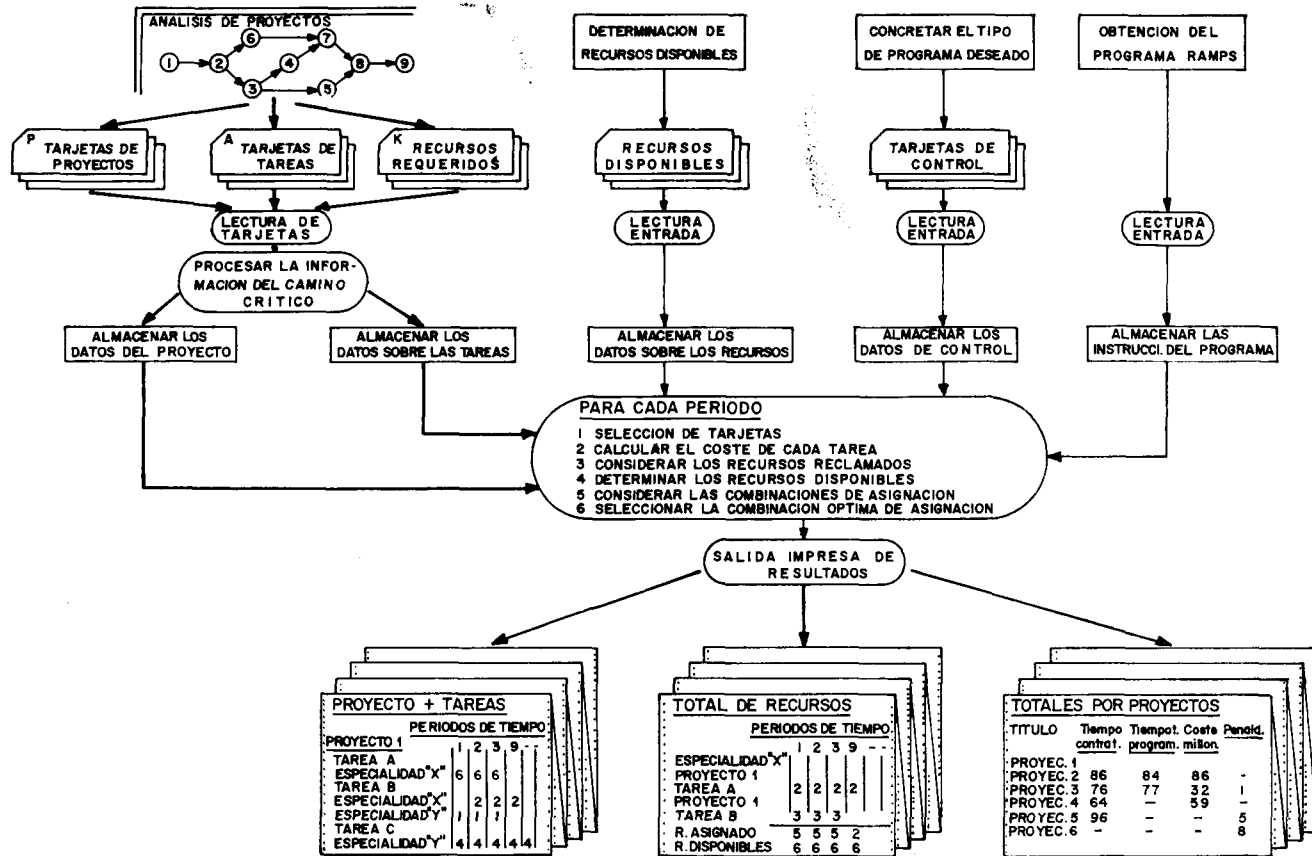


Fig.- 31

DIAGRAMA GENERAL DE INFORMACION EN EL SISTEMA "RAMPS"

Para cada tarea se especifican los distintos tipos de recursos empleados, las intensidades de utilización y la cantidad de trabajo. A lo largo de una escala de tiempo, y frente a cada tipo de recursos de cada actividad, se especifica la cantidad de recursos necesaria para la ejecución en cada unidad de tiempo. El programa de trabajo indica también, en su encabezamiento, las fechas deseada y programada de terminación y los costes por retrasos si existen.

Este programa es clave para un análisis de la idoneidad del programa de trabajo examinado.

2. Informe de aplicación de recursos.

Por cada uno de los tipos de recursos necesarios en los proyectos, el ordenador imprime un informe sobre su aplicación. La estructura de este informe es la siguiente:

Marginalmente se detallan las tareas agrupadas por proyectos. Las columnas coinciden con los valores de una escala de tiempo. En cada período y frente a cada tarea se consignan las cantidades reclamadas de recurso. Las columnas se totalizan y comparan con las cantidades de recurso disponible en cada período. Esta comparación indica las cantidades de recursos ociosos o escasos de cada período.

A la vista de este informe la dirección puede adoptar las medidas oportunas de rectificación en los programas tras detectar los fallos de los planes examinados. Esta es, pues, la gran ventaja de RAMPS: detectar, antes de que se puedan producir, las lagunas de la programación.

Aparte de los dos informes reseñados, RAMPS facilita también otros con información complementaria.

4.4.4. Los métodos SPAR y RPSM.

Existe una variada gama de métodos que se apoyan en el ordenador. J. D. Wiest ha desarrollado dos sistemas de una gran aplicabilidad denominados SPAR-1 y SPAR-2 (Scheduling Program for Allocation of Resources). Ambos utilizan, como veremos, reglas de decisión muy complejas. SPAR-1 dispone de un programa para ordenador de 32 K. Este programa, escrito en lenguaje Fortrán, está dispuesto para tratar simultáneamente, en el ordenador, 1.200 tareas, 500 nudos y 12 especialidades durante 300 períodos de tiempo. SPAR-2, se trata de una versión mejorada de SPAR-1,

en el cual se han introducido los conceptos de "secuencia crítica" y "margen condicional".

La complejidad del modelo SPAR-1 la podemos ver si analizamos, a grandes rasgos, la descripción de Wiest (22).

SPAR-1 realiza la distribución de recursos siguiendo un procedimiento en serie. Para ello se comienza por elaborar una lista con todas las tareas ordenadas para que comiencen en sus fechas más tempranas. El proceso se inicia ordenando las tareas que pueden comenzar en el primer intervalo de tiempo por sus márgenes totales. Las tareas críticas son las que tienen mayor probabilidad de ser comenzadas. Se programan tantas tareas como lo permiten las disponibilidades de recursos. Las tareas que no pueden ser comenzadas por falta de recursos se dejan para el período siguiente. Estas tareas se hacen más críticas y se mueven hacia la cabeza de la lista de tareas prioritarias.

La mecánica anterior se modifica mediante la intervención de una serie de sub-rutinas, concebidas para mejorar la utilización de los recursos disponibles y/o reducir la duración del programa.

A cada tarea se le asocian tres intensidades: normal, mínima y máxima. La selección de estas intensidades sigue las siguientes reglas:

Si una tarea a programar es crítica, se coloca en la lista de prioridad de acuerdo con su grado de criticidad (parámetro de entrada) y se le concede un tratamiento especial. Si hay suficientes hombres disponibles la tarea se programa con una intensidad o "equipo" de tamaño máximo. Si no hay suficientes hombres disponibles se programa la tarea a una intensidad normal. Si esto tampoco fuera posible, se intentan conseguir los hombres necesarios mediante el empleo de rutinas de "Préstamo y Reprogramado" (Borrow and Rescheduling) que luego consideraremos. Si a pesar de esta medida todos los esfuerzos fallan y la tarea no puede ser programada ni a intensidad mínima, se procede a retrasar su fecha más temprana de comienzo una unidad de tiempo, y se deja en espera para el período siguiente. Las tareas no críticas se programan a intensidad normal siempre y cuando estén disponibles los suficientes hombres necesarios; pero, si no lo están, ni para realizar la programación a intensidad mínima, entonces las tareas se retrasan para ser consideradas el próximo día.

Para su ejecución, las tareas pueden precisar un número diferente de recursos (por ejemplo, hombres de diferentes especialidades) que pueden ser limitados. Estas tareas multi-recursos se descomponen en tareas independientes, una para cada recurso, imponiéndoseles la restricción de que

se comiencen el mismo día y con el mismo nivel de recurso asignado: normal, mínimo o máximo (o algún nivel intermedio).

Si los recursos disponibles son insuficientes para la programación de alguna tarea crítica j, entonces el modelo utiliza un procedimiento para examinar si las tareas en ejecución tienen suficientes hombres que les permita prestarlos para la ejecución, en ese día, de la tarea j. Los hombres se prestan únicamente cuando la tarea que se desprende de ellos no provoca un retraso en la fecha de terminación del proyecto.

A veces, una tarea crítica j se programa si otras tareas que emplean el mismo recurso y están ya programadas desplazan su comienzo a un período posterior. El modelo examina la lista de tareas en ejecución y señala aquellas que se pueden demorar sin perjudicar la fecha final del proyecto. Si de esta forma se pueden obtener suficientes hombres y/o del procedimiento de préstamo descrito anteriormente, entonces se programa la tarea j y se realizan los ajustes precisos en la asignaciones previas.

Como podemos apreciar, la operatoria de SPAR-1 es complicada. Esta parece ser la característica común de todos los sistemas de programación múltiple de proyectos. Pasemos ahora al RPSM.

EL RPSM (Resource Planning and Scheduling Method) es explotado por Manchly Associates y se destina fundamentalmente a la mano de obra. El RPSM está programado para ordenador IBM 1620, con un límite de 1.600 actividades. El sistema tiene tres fases:

- Un ciclo de tiempo normal del CPM.
- La compilación de recursos (denominado ciclo no limitado del RPSM).
- Los recursos disponibles limitados.

Entre las ventajas de RPSM se encuentra la de poder reducir el volumen de mano de obra extraordinaria. J. J. O'Brien (34) cita la comparación a que la "Automotive Safety Foundation", de Washington, sometió a los sistemas RAMPS y RPSM. Para un mismo problema los resultados obtenidos por ambos fueron los mismos. No nos extraña esto si tenemos en cuenta la sistemática que Hooper (véase 35) hace de la gama RPSM,

(34) JAMES J. O'BRIEN: *CPM in construction management — Scheduling by the Critical Path Method*, cap. 15, "Equipment and Manpower Planning", McGraw-Hill, New York, 1965.

(35) ALBERT BATTERSBY: *Planificación y programación de proyectos complejos*, Ariel, Barcelona, 1970.

cuyo programa original se debe a Imbrie. Hooper clasifica la gama RPSM conforme a tres modelos: RPSM 4, RPSM 90 y RAMPS.

V. OTROS METODOS

A lo largo de nuestra exposición hemos desarrollado exclusivamente métodos heurísticos, es decir, métodos que emplean reglas de decisión empíricas susceptibles de ser establecidas en términos formales. Estos métodos parten de una solución inicial y mediante un proceso iterativo tratan de mejorarla. Puede que se llegue a una solución mejor que la inicial, pero sin la seguridad de que sea la solución óptima. También hemos hecho alusión a los métodos analíticos, pero por su laborioso cálculo resultan impracticables.

Aparte de los métodos citados existen otros tipos: métodos analógicos y métodos del perfil funcional. Los métodos analógicos se basan en analogías (físicas o abstractas) que permiten agrupar los datos de un problema de forma que la habilidad del cerebro humano pueda alcanzar una solución por juicio o por intuición. Entre las analogías más utilizadas prácticamente están las analogías físicas que utilizan gráficos de barras. Battersby (35), cita los métodos ideados por Fine y Whattingham, a base de tareas representadas por cubos de longitud unitaria desplazables en carriles horizontales con una escala de tiempo. También el panel progresivo de Andrew, a base de barras móviles; el modelo de Reaney y otros. Nosotros hemos podido comprobar la gran utilidad práctica de estos sistemas de planning a través de los modelos elaborados por la firma francesa Arianex. Otras analogías de posible utilización son las eléctricas, si bien, por el momento, no podemos hablar de una aplicación ya desarrollada.

Los métodos del "perfil funcional" estudian la forma de los perfiles de recursos y tratan de describirlos como funciones matemáticas. La "función de Norden" parece tener interés para el alisado de proyectos múltiples

VI. CONCLUSION

El tiempo, el coste y los recursos, son los tres elementos que configuran los problemas de programación de proyectos. Hemos orientado nuestro trabajo en dirección al tercero de estos elementos. De nuestra expo-

sición podemos sacar una conclusión: la necesidad de emplear el cálculo electrónico en el proceso de resolución, por los distintos métodos, de los problemas planteados. El cálculo electrónico es hoy, como indica Jacques Wanty (36), junto con la aplicación masiva y generalizada de métodos científicos en los problemas de gestión y dirección de empresas públicas y privadas, un hecho experimental, logrado e irreversible. Ahora bien, la utilidad de este poderoso instrumental depende de la disciplina y del control que se tiene sobre el conjunto de los hombres que han de facilitar la información a procesar. Una información deficiente conducirá, de cierto, a unos resultados erróneos. Por esta razón los métodos presentados en nuestro trabajo reclaman, para el logro de una fructífera aplicación, unas estructuras en donde exista autoridad, disciplina y responsabilidad. De esta manera los métodos considerados favorecerán en su aplicación práctica, como dicen Kaufman y Desbazeille (37), la "exactitud", la "clarividencia" y la "audacia" en una excepcional demostración de lo que pueden hacer la investigación operativa y la informática en la organización.

(36) JACQUES WANTY: *Contenu et portée de la science du management*, "Organisation Scientifique", núm. 4, 1969.

(37) A. KAUFMANN y G. DESBAZEILLE: *Método del camino crítico*, Sagitario, Barcelona, 1965.

