ESCUELA TÉCNICA SUPERIOR DE ARQUITECTURA DE SEVILLA

GENERACIÓN Y OPTIMIZACIÓN AUTOMÁTICA DE ESQUEMAS DE DISTRIBUCIÓN DE EDIFICIOS EN PLANTA

Tesis Doctoral
presentada por:

Juan José Sendra Salas

Dirigida por:
D. Alberto Donaire Rodríguez
Dr. Arquitecto
Catedrático de Elementos de Composición, E.T.S.A. de Sevilla

Sevilla, 1984

ÍNDICE

Pág.

I. INTRODUCCIÓN

Los teóricos de la arquitectura que, en los años sesenta y primeros setenta, auguraban una transformación radical de los modos de proyectar gracias a la utilización de los ordenadores, no han visto cumplidas sus previsiones. En contraste con ello, asistimos a una rápida adopción de técnicas informáticas como instrumento de trabajo en las ciencias físicas y también en las ciencias humanas, que han visto así eficazmente aumentadas sus posibilidades de desarrollo.

La diferencia que en este sentido existe entre el campo de las ciencias y el de la arquitectura estriba quizá en que, en el primer caso, los ordenadores han podido aportar directamente su capacidad instrumental, mientras que, en lo que al diseño arquitectónico se refiere, era necesario previamente revisar y establecer las formulaciones teóricas de partida; era imprescindible elucidar primero qué operaciones y decisiones se podrían encomendar al ordenador, y cuales otras sería inadecuado o imposible hacerlo. O incluso, llegando más lejos, averiguar si es factible, estudiando a fondo la naturaleza del proceso de diseño, sustituir al hombre por la máquina, en qué medida y con qué modelos operativos.

Negroponte (1970, p.11) afirmaba hace una década que el ordenador puede emplearse en el proyecto de tres modos posibles:

> 1. Automatizzare i procedimenti attuali rendendoli più rapidi e meno costosi.
> 2. Modificare i metodi attuali per adattarli alle esigenze e alla struttura della macchina, il che implica di - considerare soltanto i problemi che si ritengono compatibili con la macchina.
> 3. Presentare il processo della progettazione considerato come evolutivo, a una macchina pure considerata come - evolutiva, in modo che si sviluppino un apprendimento, un adattamento e una crescita reciproci.

Pero hoy, que la evolución tecnológica y los métodos de producción de componentes nos ofrecen ordenadores de gran fiabilidad a bajo precio, vemos sin embargo que las aplicaciones - informáticas más extendidas en el campo de la arquitectura se limitan al primer grupo de los señalados por Negroponte: la - automatización de los procedimientos conocidos, como cálculos estáticos o estimaciones de costo, en los que sólo se aprovecha la rapidez operativa o la capacidad de almacenamiento de información, aspectos en los que el ordenador supera indudablemente a la mente humana, quedando la toma de decisiones reservada enteramente al diseñador.

Además de las posibilidades señaladas, existe otro campo de - empleo del ordenador en el proyecto, no tan inmediato pero de gran valor. Scarano (1979, p. 38) lo describe así:

> L'uso di un calcolatore programmato per produrre immagini può aumentare in modo considerevole la posibilità di controllo del prodotto finale, sia da parte del proggettista sia da parte dell'utenza; permette, infatti, di effettuare in laboratorio prove sperimentali che altrimenti sarebbe - impossibile compiere, mostrando su video il prodotto architettonico in diversi ambienti e sotto differenti angolazioni prima ancora della sua reale trasformazione in prodotto fisico compiuto.

No tan extendidas como las anteriores, se ofrecen hoy aplicaciones en esta dirección. Se trata de complejos programas, mediante los cuales el diseñador, en diálogo con la máquina, va tomando decisiones en el terreno formal, ensayando y cambiando los espacios y formas que ha hecho aparecer en pantalla. El ordenador actúa aquí como un rápido y sofisticado instrumento de dibujo, pero no toma decisiones por sí mismo. En este sentido y aunque más espectaculares, los programas de dibujo ayudados por ordenador (CAD) no dejan de pertenecer al primer grupo de Negroponte.

Entender el diseño de arquitectura como un proceso automatizable en su totalidad es, hoy en día, una mera elucubración. Y así, autores que propugnan el empleo del ordenador en arquitectura, reconocen sus límites al llegar a este punto. Alexander (1966 pp. 76-77) cuyo pensamiento posterior se ha alejado de estas cuestiones, decía:

> Todos aquellos problemas de creación de formas que tradicionalmente se designan como "problemas de diseño" requieren inventiva |...| , no es posible, por lo tanto, reemplazar las acciones de un diseñador diestro, por decisiones computadas mecánicamente.

O Broadbent (1974, pp. 14-15) con parecido criterio escribía:

> El diseño en arquitectura no puede ser nunca un hecho de decisión automática o automatizable |...| hay cosas en el diseño arquitectónico que no se pueden cuantificar. Cosas que son cuestión de la imaginación, valores, identidad, sentido del lugar, etc.

Otros autores, en cambio, confían en que las investigaciones sobre la naturaleza del proceso de diseño pudieran hacernos avanzar hacia su automatización. Así, Sevilla (1981, p. 8), afirma:

> Los esfuerzos por informatizar el proyecto de arquitectura han estado vinculados a investigaciones teóricas sobre el diseño. Estas se orientan a la racionalización de procesos y búsqueda de modelos matemáticos que permitan organizaciones complejas.

Seguí (1) está también en esta línea, tratando de distinguir categorías en el proceso de diseño,su separación en partes, que permitan elegir modelos matemáticos adecuados.

Con motivo del Congreso de la Unión Internacional de Arquitectos que tuvo lugar en Madrid en 1975, se realizaron encuestas para recoger opiniones de diversos países sobre esta cuestión. Las respuestas fueron diversas; desde los que, como la República Democrática Alemana, afirmaban la unidad del proceso de diseño y la imposibilidad de su descomposición para un tratamiento sistemático, hasta los que, como Estados Unidos o la Unión Soviética, confiaban en la posibilidad de emplear modelos analíticos matemáticos o gráficos como instrumentos innovadores en el proceso de diseño. En las conclusiones del Congreso (2) se lee:

> De todo lo anteriormente expuesto, puede deducirse que el empleo de computadoras es idóneo para el diseño de algunos aspectos técnicos de la edificación, pero no se encuentra desarrollado en forma suficiente para resolver el proceso de diseño en el campo de la ideación arquitectónica; a este respecto, nos encontramos sin duda en el comienzo de toda una búsqueda investigadora que quizás en el futuro arroje resultados más complejos.

Nuestro posición se encuentra en un punto intermedio en este debate. Nos declaramos escépticos ante una posible automatización global del diseño. Pero ello no obsta para que no pueda -

(1) Citado por Sevilla (1981, p. 17)

(2) Congreso de la U.I.A., Actas, varios autores (1975)

aceptarse la posibilidad de crear instrumentos que ayuden a la toma de decisiones que el diseño comporta.

La reflexión sobre muchos de los problemas no cuantitativos - que el diseño encierra nos lleva a observar que su estructura interna es traducible a modelos lógico-matemáticos; arquitectos como Alexander, Xenakis o Friedman han ofrecido métodos - que se basan en esta idea.

La matemática moderna ofrece, efectivamente, instrumentos adecuados para afrontar problemas también cualitativos, no sólo - cuantitativos. El desarrollo de las nociones de teoría de conjuntos, grupos y grafos permite modelar situaciones contempladas no ya por la geometría métrica, esencialmente cuantitativa, sino también por la geometría proyectiva o la topológica.

Estos modelos matemáticos son susceptibles de programación y - tratamiento en un ordenador, utilizando ampliamente su capacidad lógica.

Entre los problemas que pueden ser abordados de este modo se - encuentran los de optimización. Y nuestro trabajo pretende hacer una contribución a este campo, centrándonos en la generación automática de esquemas de distribución del espacio en las dos dimensiones de la planta de un edificio, a partir de condiciones de diversa índole impuestas por el diseñador. Es un problema inicialmente topológico y finalmente métrico. Y de su resolución depende, en cierta medida, la posibilidad de contar - con un instrumento válido para una primera aproximación a la - organización del espacio al proyectar.

II. GENERACIÓN AUTOMÁTICA DE ESQUEMAS DISTRIBUTIVOS. DESCRIPCIÓN DEL PROBLEMA.

II.1. ESTADO ACTUAL DE LA CUESTIÓN

El modo de abordar el problema de la generación automática de esquemas adimensionales de organización espacial ha sido diferente según los objetivos propuestos y las condiciones fijadas. Sin embargo, los investigadores que perseguían como resultado final esquemas dimensionados, han seguido en general un proceso dividido en dos fases sucesivas:

- Generación de esquemas adimensionales.
- Dimensionamiento.

Así han tratado el problema, entre otros, Krejcirik (1969), Cousin (1970), Shaviv y Gali (1974), Mitchell, Steadman y Ligett (1976), Mitchell (1977), Korf (1977), March y Earl (1977 y 1979) y Scarano (1979). Será, pues, en función de estas dos etapas como analizaremos los distintos métodos de solución propuestos a partir de los años sesenta.

II.1.1. Generación de esquemas adimensionales

Existen dos planteamientos iniciales de este problema, que luego se traducen en una amplia gama de métodos de resolución: el primero lo enfoca como un problema de optimización, buscando la mejor disposición posible de los espacios que van a componer la planta; el segundo planteamiento entiende el problema como combinatorial, y busca todas las disposiciones posibles que cumplan unas condiciones dadas.

Son relativamente muchos los investigadores que se han inclinado por resolver la generación de esquemas adimensionales como un problema de optimización (1). Ya en los primeros trabajos en este campo -Levin (1964), Eastman (1970), Shaviv y Gali (1974)- el criterio de "óptimo" iba fundamentalmente dirigido a la disposición y distancias entre los distintos ambientes, y se definía por medio de una "función objetivo" en la que se hacían intervenir factores tales como la minimización de recorridos entre dos actividades -Levin (1964)- o la condición de que dos espacios estuviesen uno junto a otro -Ritzman (1972)- o una combinación de ambos -Shaviv y Gali (1974)-. Las distintas unidades espaciales que componen el conjunto están además, sujetas a unas limitaciones que, fundamentalmente, son de adyacencia, orientación, acceso, visibilidad y distancia -Eastman (1972)-.

El problema es planteado como combinatorial en los años setenta por el LUBFS de la Escuela de Arquitectura de Cambridge (Inglaterra) y por la Escuela de Arquitectura y Planeamiento Urbano de la Universidad de California (USA). El número de soluciones posibles viene restringido por una serie de requisitos

(1) "Il problema di localizzare in modo ottimale ambienti è nato nella cultura anglosassone all'interno di un discorso settoriale di programmazione e pianificazione della progettazione, al fine di produrre un sistema di organizzazione spaziale che, in termini di costo, minimizzi la distanza, intesta questa come tempo di spostamento e quindi come lavoro, anch'esso da minimizzare". Scarano (1979, prefacio).

que han de cumplir los elementos espaciales, limitaciones que cuando existen en elevado número pueden llegar a alterar incluso la índole combinatorial del problema.

Han sido numerosos los intentos y variados los algoritmos propuestos en la última década para generar, las más de las veces, esquemas rectangulares formados por espacios también rectangulares. Casi todos ellos se han basado en la teoría de grafos: en efecto, desde los primeros trabajos -Cousin (1970), March y Steadman (1971)- se parte de la idea de que las condiciones de adyacencia de los espacios a ordenar pueden expresarse mediante un grafo, cuyo grafo dual será precisamente un esquema que cumpla las condiciones fijadas.

Dados los requisitos exigidos a los espacios componentes, es preciso comprobar si el grafo resultante es conexo y planar, es decir si los distintos elementos forman un solo conjunto de espacios interconectados, y si las condiciones impuestas tienen solución en dos dimensiones. Existen procedimientos que comprueban si el grafo tiene esas características; en este sentido cabe destacar los trabajos de Kuratowski (1930), Tutte (1962), Fisher y Wing (1966); Bruno, Steiglitz y Weinberg (1969); Hopcroft y Tarjan (1973,a y 1974), Read (1979), etc.

Si, además, se impone la limitación de rectangularidad del conjunto, a los ya citados "tests" de conectividad y planaridad habrá que añadir un test de rectangularidad -Mitchell, Steadman y Ligett (1976), Scarano (1979)-. Las operaciones de reflexión y rotación no alteran la naturaleza de las soluciones, salvo si las condiciones de orientación de los espacios se toman como requisitos, como se hace con frecuencia.

Ha habido algunos intentos de superar la limitación de la rectangularidad de los espacios: Shaviv y Gali (1974) introduciendo rectángulos ficticios, o Korf (1977) con mayores pretensiones teóricas.

En cuanto a los métodos de solución, Nugent (2) distingue entre técnicas de "construcción" y técnicas de "mejora". Las primeras parten del estado nulo y, mediante introducción de razones de conveniencia de localización, van llegando a una solución. En las técnicas de mejora se parte de una solución inicial que va optimizándose en sucesivas evaluaciones.

Mitchell (1977) clasifica los métodos de solución, de más "débiles" o generales a más "fuertes" o específicos, del siguiente modo:

- Procedimientos de generación y comprobación: considerados como poco eficientes, consisten en una generación secuencial de elementos, hasta que aparece un elemento del conjunto "objetivo", generación que puede ser aleatoria o exhaustiva.

- Procedimientos de mejora: recomendables cuando hay una buena solución de partida, coinciden con la clasificación de Nugent. Establecida una "función objetivo" a optimizar, se introducen en la solución inicial sucesivos cambios, que son adoptados si aumenta el valor de la función. La estrategia de cambio puede ser aleatoria o de búsqueda de dirección de mayor mejora.

- Procedimientos heurísticos: se trabaja sobre un cierto conocimiento previo de la estructura del problema, a los que se incorporan los conocimientos obtenidos tras cada operación de cambio, para seleccionar secuencias de operaciones a realizar. Existen diferentes técnicas de evaluación de las operaciones efectuadas y determinación de las operaciones a realizar o planificación de los procesos de búsqueda y "feed - back".

(2) Citado en Portlock y Whitehead (1974).

II.1.2. Dimensionamiento

La mayoría de los investigadores tratan la fase de dimensionamiento de esquemas como un problema de optimización: Krejcirik (1969), Cousin (1970), Mitchell, Steadman y Ligget (1976), Mitchell (1977), Gero (1977), Earl y March (1979), o Scarano (1979) están entre ellos.

Para Gero (1977) el diseño es un problema de optimización, que necesita determinar los valores actuales de las variables de una función objetivo en función de los valores a alcanzar en el futuro, a diferencia de los problemas de simulación, que infieren los valores futuros de los actuales.

La formulación general de un problema de optimización viene dada por el establecimiento de una función objetivo y unas restricciones a cumplir por las variables que en ella intervienen. En los casos que nos ocupan, estas variables son esencialmente la longitud y anchura de cada uno de los espacios, con límites mínimos o máximos, y el contorno del conjunto. Casi todos los investigadores trabajan con esquemas rectangulares, y contorno asímismo rectangular, a lo sumo, introducen el concepto de "rectángulo ficticio" para poder emplear también espacios con planta en T, en L, en U, etc., y contorno general de similares características.

Para el dimensionamiento de esquemas se han establecido diferentes tipos de funciones-objetivo: los más usuales maximizan o minimizan el contorno general, su longitud o su anchura, su proporción o su superficie. Unas funciones objetivo serán, pues, lineales, otras no-lineales.

En cuanto a las restricciones de las variables, un primer grupo es el ya citado, referido a la longitud o anchura de los espacios, su superficie, etc., que se materializan en acotación de estas variables. Un segundo tipo engloba los requisitos de adyacencia o de comunicación entre espacios, y existe un tercer grupo, por último, que incluye restricciones derivadas de la propia naturaleza del esquema adimensional, y de las relaciones entre sus espacios. Todas estas limitaciones pueden formularse, unas en forma lineal, otras en forma no-lineal.

Para expresar estas restricciones, muchos investigadores han reducido el modelo adimensional a dos grafos orientados, uno "vertical" y otro "horizontal", a los que se aplican las leyes de Kirchhoff para redes eléctricas.

La naturaleza, lineal o no-lineal, de las funciones-objetivo y los valores, reales o enteros, que pueden adoptar las variables, determinan el método de resolución a emplear en el dimensionamiento. A continuación se enumeran y describen los más comunes:

- Programación lineal.

  Es una técnica muy poderosa, que garantiza el alcanzar una solución óptima con gran eficiencia, utilizando con gran frecuencia el algoritmo "simplex". Lógicamente, tanto las restricciones de las variables como la función-objetivo han de ser traducibles a expresiones lineales. Cuando se emplea una trama modular, las variables solo asumen valores enteros.

- Programación cuadrática.

  Técnica usada con menor frecuencia, permite que, siendo lineales las restricciones de las variables, la función-objetivo pueda adoptar forma cuadrática.

- Programación no-lineal.

  Se emplea esta técnica cuando tanto las restricciones de las variables como la función-objetivo pueden adoptar forma no-lineal. Se han desarrollado varios métodos con esta técnica: algunos son rudimentarios y aplicables solo en ciertos casos -por ejemplo, los métodos de linealización-; otros utilizan algoritmos más sofisticados y de mayor aplicabilidad.

- Programación dinámica.

  Se llega a la solución final a través de una secuencia de decisiones. Con ella pueden resolverse problemas tanto lineales como no-lineales.

II.2. ALTERNATIVAS E HIPÓTESIS DE TRABAJO

El punto de arranque de nuestra investigación para desarrollar un procedimiento automático de generación de esquemas distributivos bidimensionales fue la propuesta de Mitchell (1977), que utiliza la "representación mediante el grafo dual". Este autor subdivide el proceso en las siguientes etapas:

1.- Definición, en forma matricial, de las condiciones de adyacencia entre los distintos espacios componentes.

2.- Comprobación de la planaridad del grafo correspondiente a tales condiciones de adyacencia.

3.- Trazado del grafo planar, sin cruce de aristas.

4.- Construcción del correspondiente grafo dual, una representación en planta que cumple los requisitos iniciales.

5.- Introducción de nuevas limitaciones de formas, dimensiones y localización de los espacios que, junto con las de adyacencia, determinarán el esquema final.

Mitchell considera sin embargo su propuesta de difícil implementación en un ordenador. Siguiendo la indicación del mismo y otros autores, hemos estimado conveniente en primer lugar subdividir el procedimiento en las dos fases señaladas en II.1: en la primera se obtienen esquemas adimensionales estableciendo las condiciones de adyacencia y orientación de los espacios componentes, y limitando además la forma de los esquemas de modo que tanto los componentes como el conjunto sean rectangulares; como vimos en II.1.1, este criterio simplificador se sigue en la mayor parte de los trabajos realizados hasta la fecha. En la segunda fase se hacen intervenir restricciones métrico-geométricas de los espacios para conseguir un esquema dimensionado.

La totalidad del procedimiento se ha programado en lenguaje Fortran IV para un ordenador Hewlett Packard serie 1000; los programas operan de forma interactiva con arreglo a los algoritmos que se desarrollan en el capítulo siguiente.

II.2.1. Generación de los esquemas adimensionales

Las cuatro primeras etapas del procedimiento de Mitchell obedecen a una formulación de tipo combinatorial: dada una serie de espacios, se van a generar todas las localizaciones "posibles" de los mismos -test de planaridad- que cumplan con los requisitos de adyacencia inicialmente impuestos. El número de soluciones dependerá lógicamente de la cantidad de restricciones introducidas.

La orientación de nuestro trabajo es distinta. Pretendemos llegar a definir las condiciones necesarias para que la solución sea única; por ello, el problema no podrá tener un planteamiento combinatorial.

Pero imponer indiscriminadamente condiciones para llegar a una solución única haría cerrado e inflexible el proceso, porque es difícil, sobre todo con un número elevado de espacios componentes, saber "a priori" que las condiciones establecidas conducen a una única solución posible. Aumentando las condiciones disminuyen las posibilidades de existencia de solución. De modo que se convertiría en un hecho casual -de probabilidad mínima- el llegar a una solución a partir de condiciones establecidas en bloque.

Por ello hemos optado, aprovechando la interactividad del programa, por una introducción gradual de requisitos, comenzando con los que se estiman imprescindibles o más importantes, y continuando con otros que, al final, conduzcan a la solución.

Nuestro procedimiento de generación de esquemas adimensionales se divide en las siguientes etapas:

1. Condiciones iniciales: definición de las adyacencias y orientaciones de los espacios componentes. Construcción del grafo inicial.

2. Test de conectividad: si el grafo inicial no es conexo, introducción de nuevas condiciones de adyacencia para lograr que lo sea.

3. Primer test de planaridad: si el grafo no es planar, sustitución o eliminación de las condiciones que lo impiden.

4. Test de triconectividad: si el grafo no es triconexo, trazado de sus componentes triconexas e introducción de condiciones que las enlacen suficientemente.

5. Segundo test de planaridad, para comprobar si las condiciones introducidas siguen permitiendo una solución planar.

6. Test de rectangularidad.

7. Triangulación del grafo.

8. Transformación de coordenadas de los vértices del grafo como paso previo a la construcción del grafo dual.

9. Construcción del grafo dual, que es el esquema de distribución buscado, cumpliendo todas las condiciones impuestas.

Este proceso que se detalla ampliamente en el capítulo siguiente, no es lineal: permite un "feed back" si el resultado obtenido no es satisfactorio. Cuando las modificaciones que ahora introdujéramos, que pueden incluso afectar a las condiciones iniciales, no fuesen tales que variara la estruc tura del grafo de modo esencial, bastaría repetir las etapas 7, 8 y 9. En caso contrario se trataría de un nuevo problema, cuya solución comenzaría desde el principio.

Como puede verse, el proceso es altamente interactivo: el usuario decide en todas las etapas, añadiendo o alterando las condiciones que juzga oportunas, y determina cuándo el proceso ha terminado, al aceptar la solución generada.

II.2.2. Dimensionamiento

La quinta etapa en el proceso de Mitchell supone la introducción de requisitos dimensionales y formales en el esquema adimensional, sin atender a la formulación del mismo. Pero la mayoría de los investigadores han entendido el dimensionamiento de esquemas como un problema de "optimización" y, por consiguiente, han aplicado como métodos de resolución los derivados de la programación. Nosotros consideramos igualmente la "optimización" como técnica idónea para resolver el problema de dimensionamiento, y emplearemos la programación lineal y no lineal como métodos de resolución adecuados a esa formulación.

No aplicaremos, sin embargo, la programación dinámica, porque su utilidad depende en gran medida de la habilidad del usuario para descomponer la estructura del problema.

El procedimiento que se propone para el dimensionamiento de esquemas puede subdividirse en las siguientes etapas:

1. Sustitución del esquema adimensional por dos grafos dirigidos, uno "horizontal" y otro "vertival"

2. Definición de las condiciones de accesibilidad entre locales.

3. Condiciones métrico-geométricas que deben cumplir los espacios componentes y el contorno.

4. Expresiones, lineales o no lineales, correspondientes a las condiciones anteriores.

5. Determinación de la función objetivo.

6. Resolución del problema por programación lineal o no lineal, según su naturaleza.

El proceso termina con el trazado del esquema dimensionado. El contorno y todos los espacios componentes tendrán forma rectangular.

III. ALGORITMOS PARA LA GENERACIÓN DE ESQUEMAS DISTRIBUTIVOS

En el capítulo anterior quedan someramente descritas las etapas a seguir en la generación de esquemas adimensionales y su posterior dimensionamiento. Como ya se dijo, estos dos procesos, independientes pero complementarios, resuelven el problema del trazado de distribuciones en planta. En el presente capítulo entraremos a detallar las operaciones y algoritmos que desarrolla cada etapa.

III.1. GENERACIÓN DE ESQUEMAS ADIMENSIONALES

III.1.1. Fijación de condiciones iniciales

Las condiciones iniciales del problema son el conjunto de espacios o locales intervinientes en el esquema y sus relaciones de adyacencia y orientación. Como se irá viendo, es conveniente en este punto fijar sólo las relaciones más importantes, dejando las demás para decisiones posteriores.

Estos datos constituyen un grafo simple (sin bucles ni aristas múltiples): los locales serán sus vértices y las relaciones entre ellos sus aristas.

Este grafo abstracto, G_1, queda definido por una matriz de incidencia, B (b_{ij}), cuyas dimensiones son el número de vértices $|V|$ y el de aristas $|A|$. Sus términos serán: $b_{ij} = 0$ cuando la arista i no incide en el vértice j, y $b_{ij} = 1$ en caso contrario.

Los cuatro primeros vértices representan siempre a los puntos cardinales (vértices exteriores), en sentido antihorario: 1, norte; 2, oeste; 3, sur, y 4, este. Los diferentes locales se enumeran del 5 en adelante.

Igualmente, las cuatro primeras aristas representan las relaciones naturales de adyacencia entre puntos cardinales: 1, norte-oeste; 2, oeste-sur; 3, sur-este y, 4, este-norte. Las relaciones entre locales se enumeran del 5 en adelante.

De la matriz de incidencia se obtiene inmediatamente la de adyacencia, A (a_{ij}), de dimensión $(|V|, |V|)$, cuyos términos son: $a_{ij} = 0$ cuando los vértices i y j no están ligados por arista, y $a_{ij} = 1$ en caso contrario.

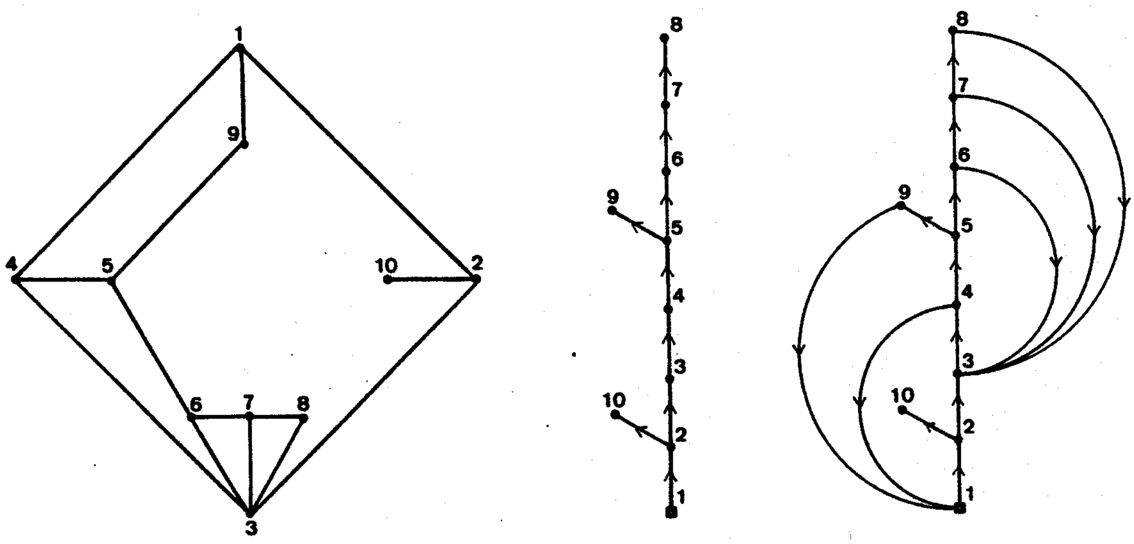


Figura 1

III.1.2. Comprobación de conectividad

Condición previa a la ulterior obtención del dual de un grafo es la comprobación de que éste es conexo y planar.

El algoritmo elaborado como "test de conectividad" se basa - en los estudios de Hopcroft y Tarjan (1973,a). Se utiliza la técnica denominada "depth first search" que, partiendo de la matriz de incidencia, transforma el grafo en una "palmera" - (véase apéndice).

La técnica consiste, en primer lugar, en construir un árbol dirigido a partir de un vértice inicial, cuyos arcos se corresponden con aristas del grafo de partida. Las aristas restantes, "ramas", se añaden al árbol dirigidas en sentido contrario a los arcos, es decir, su origen será el vértice más alejado del vértice inicial. Figura 1.

Cuando la palmera así obtenida no contiene a todos los vértices del grafo, éste será inconexo, y tendrá por lo tanto más de una componente conexa.

El algoritmo procede a "marcar" los vértices accesibles desde el que tomó como inicial. Si el grafo es inconexo, irá construyendo nuevas palmeras tomando como vértice inicial alguno de los no marcados, y determinando así todas las componentes conexas que existan.

Llegado a este punto, el programa informa del estado de conectividad del grafo, y describe las componentes en caso de no conectividad. El operador debe decidir qué nuevas relaciones introduce para conectar entre sí las componentes.

Con esta intervención, el grafo de partida G_1, es ahora un grafo conexo, G_2. Si G_1 era conexo, G_2 es idéntico a él; si no lo era, G_2 tiene el mismo número de vértices, pero más aristas. Las matrices de incidencia y adyacencia han quedado modificadas según esta nueva estructura conexa.

III.1.3. Primera comprobación de planaridad

La comprobación de planaridad se realiza mediante algoritmos desarrollados a partir de los trabajos de Hopcroft y Tarjan (1974) (1), que son una versión de un método originalmente propuesto por Auslander y Parter y formulados posteriormente por Goldstein.

---

(1) El análisis de una serie de propiedades del grafo: conectividad, biconectividad, triconectividad y planaridad puede abordarse desde una misma óptica. Así lo han tratado estos autores. Su eficiencia frente a otros métodos con idénticos fines y, sobre todo, su adaptabilidad al método que desarrollamos, ha resuelto que sea recogido por este trabajo.

Como en el test de conectividad, se construye ahora una palmera a partir del grafo G_2. A continuación se aplica el procedimiento de Auslander, Parter y Goldstein, consistente en identificar un circuito que comience y termine en el vértice inicial. Al suprimir este circuito, el grafo queda reducido a unos "segmentos" conexos que inciden en ese circuito. El test de planaridad consistirá en comprobar si es posible una representación plana del circuito base y de los segmentos sin cruce de aristas.

No obstante, una aplicación eficiente de la serie de algoritmos que se necesitan desarrollar para comprobar la planaridad exige un procedimiento más completo. Comprendería: una primera comprobación de la planaridad por la aplicación de la fórmula de Euler, una descomposición del grafo en sus componentes biconexas y, por último, la comprobación de la planaridad de cada una de las componentes biconexas.

Describamos a continuación cada una de estas operaciones:

1.- Aplicación de la fórmula de Euler.

Para que un grafo sea planar, es condición necesaria que se cumpla la siguiente expresión (2): $|A| \leq 3 \cdot |V| - 6$

2.- Descomposición del grafo en sus componentes biconexas.

'Un grafo es planar si y sólo si sus componentes biconexas son planares', Berge (1970).

Las componentes biconexas se identifican con comodidad en la palmera obtenida de G_2. Los vértices de separación de componentes serán comunes a dos subpalmeras no unidas entre sí por ramas. Figura 2.

---

(2) Para $|V| \geq 3$

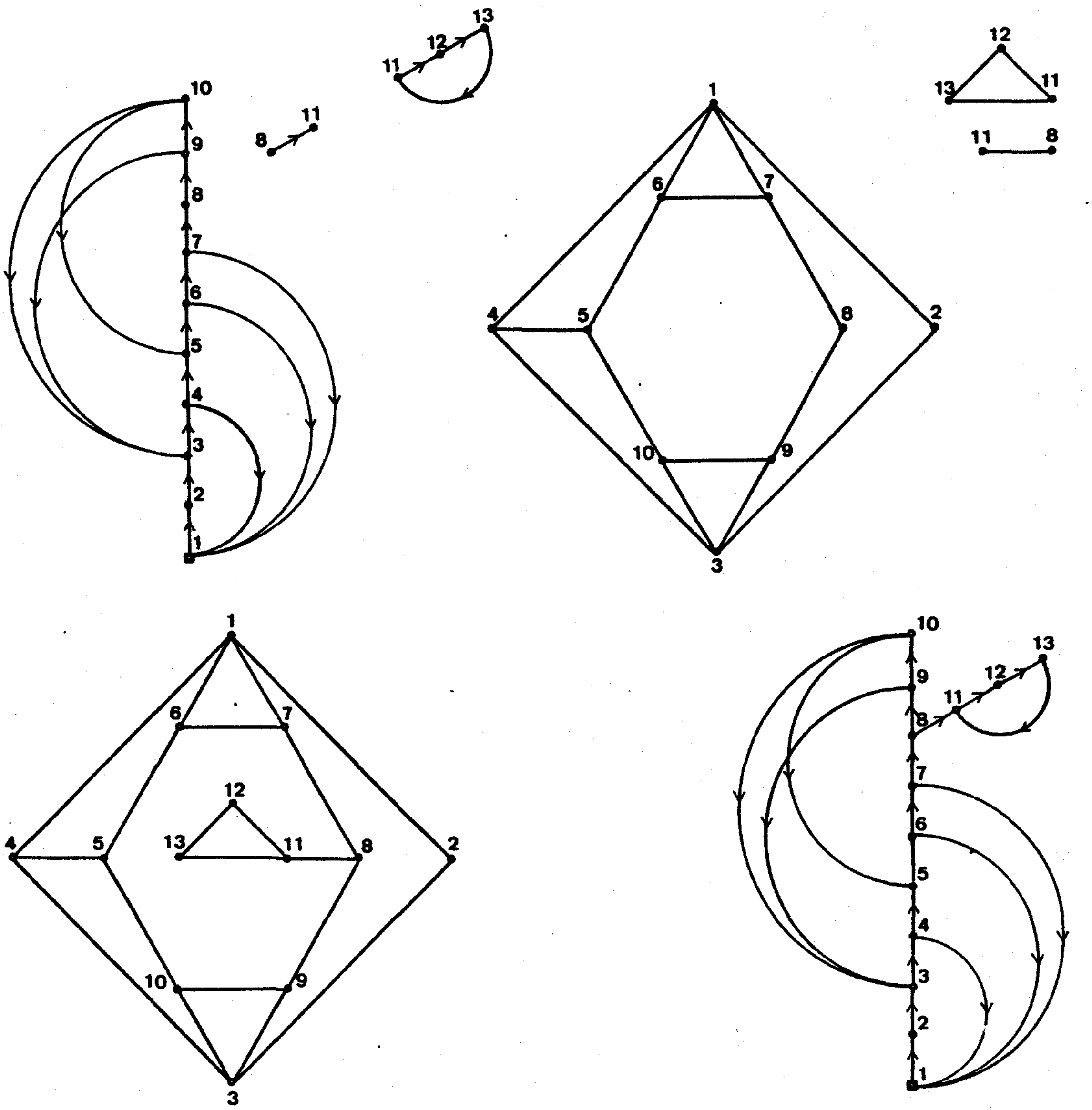
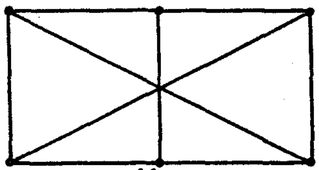
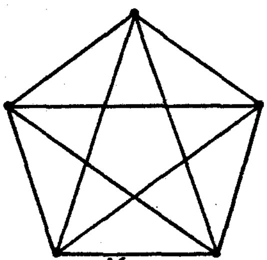


Figura 2

Si el grafo G_2 (conexo) no contiene ningún vértice de separación será además biconexo. Si G_2 no es biconexo lo subdividiremos en sus componentes biconexas. Para ello efectuaremos un nuevo recorrido sobre la palmera con arreglo a la técnica citada. Cada vez que alcanzamos un vértice de separación comienza una nueva componente biconexa. El vértice de separación para dos componentes biconexas estará incluído en ambas. Figura 2.



K33



K5

Figura 3

3.- Comprobación de la planaridad de cada una de las componentes biconexas.

Antes de efectuar la comprobación conforme a los procedimientos descritos, se puede realizar un control, rápido y cómodo, a la componente biconexa que posibilita el teorema de Kuratowski -Kuratowski (1930)- : 'Un grafo es planar si no contiene a los subgrafos K_{33} y K_5'. Figura 3.

El grafo nominado K_{33} contiene seis vértices y nueve aristas y el grafo K_5 posee cinco vértices y diez aristas. Cualquier componente biconexa que tenga menor número de vértices o menor número de aristas será por tanto planar y no habrá que aplicar sobre ella ninguna otra comprobación de planaridad.

A la componente biconexa que no cumpla estas características se le aplicará el procedimiento general.

Podríamos describirlo con mayor detalle del siguiente modo:

- Aplicamos una "depth first search" sobre la componente biconexa que se analiza transformándola en una palmera.

- Volvemos a aplicar esta misma técnica para obtener todos los caminos disjuntos sobre esta palmera. La construcción de caminos se caracteriza porque cuando se alcanza una rama de la palmera , ésta constituye la última -y a veces la única-arista orientada del presenta camino. Empezaría entonces un nuevo camino, con un nuevo vértice inicial en el vértice origen de la rama.

El primer camino que se genera comienza y termina en el vértice inicial de la componente biconexa. Sería un circuito; lo llamaremos circuito original.

- Suprimimos el circuito original. Cada uno de los segmentos resultantes constará, o bien de una sóla rama de la palmera, (W,V), o de un arco del árbol, (V,W), más un subárbol de origen W y todas las ramas que salen de él.

- Analizamos cada uno de los segmentos en el orden inverso a como han sido generados. Por el teorema de la Curva de Jordan, un segmento puede ser representado en el plano bien por el exterior, o por el interior del circuito original, sin que se produzcan cruces de sus aristas.

La localización en el plano de un nuevo segmento puede llevar consigo que algunos segmentos, ya situados, deban ser movidos del interior al exterior del circuito original, o viceversa.

- Si se ha podido ubicar todos y cada uno de los segmentos, la componente biconexa es planar. Basta que un segmento no pueda ser situado para que dicha componente biconexa no sea planar. En éste último caso, tendríamos que sustituir un arco o rama de este segmento por otro distinto para poder continuar el proceso.

El proceso completo se recoge en la Figura 4.

La operación de generación de caminos disjuntos y representación o "empotramiento" de los segmentos pueden ser realiza

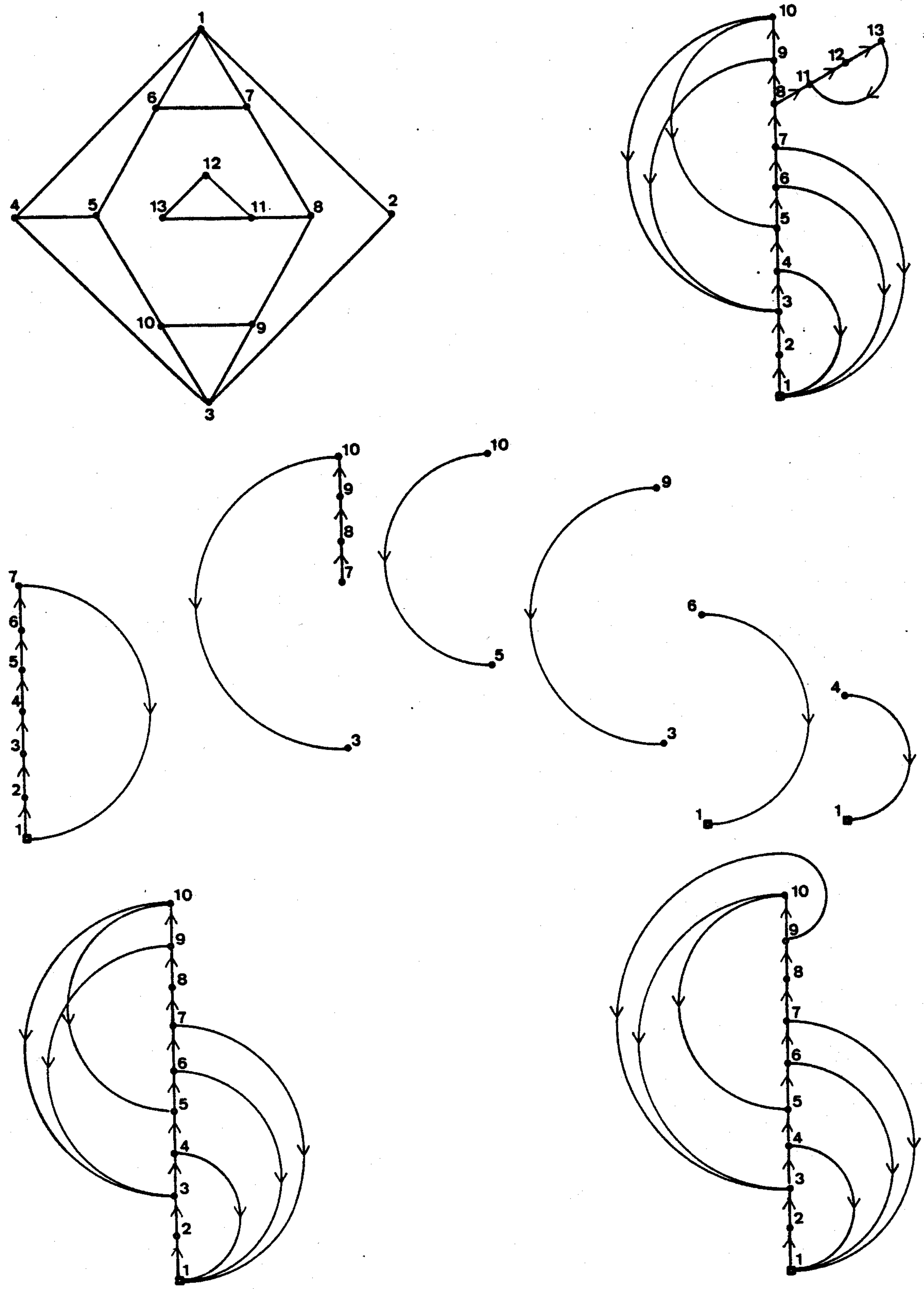


Figura 4

das simultáneamente. Así ha sido propuesto por Hopcroft y Tarjan (1974) y así ha sido recogida en este trabajo.

Después de esta etapa el grafo, G_3, será geométrico conexo y planar.

III.1.4. Análisis de la triconectividad

El procedimiento que estamos desarrollando limitará progresivamente el número de soluciones posibles por la introducción de nuevos requisitos de adyacencia y orientación. Pero la adición de nuevas restricciones ha de realizarse a partir de trazados geométricos planos del grafo o de sus subgrafos. De este modo, la inclusión de nuevas relaciones puede hacerse de forma racional para no producir alteración de la planaridad.

El teorema de Whithney nos dice: 'Un grafo ha de ser necesariamente triconexo para que tenga una representación única en el plano para cada una de sus caras o regiones'.

Así pués, si el grafo no es triconexo, añadiremos una serie de aristas al grafo para que tenga esta propiedad.

Un trazado en el plano claro y sencillo del grafo, ya triconexo, facilitaría la introducción de nuevos requisitos de adyacencia y orientación, sin necesidad de tener que efectuar la comprobación de la planaridad. Esto nos permitiría restringir paulatinamente el problema y llegar a una solución: una representación en planta.

Pero la inclusión de nuevas aristas para hacer el grafo triconexo, requiere a su vez de trazados geométricos planos con las características citadas. La primera operación que debemos realizar, por tanto, será la determinación de las componentes triconexas del grafo. Esto se hará distinguiendo las componentes triconexas de cada una de las componentes biconexas que se han obtenido en la etapa anterior, previo al estudio de la planaridad (3).

El procedimiento que se ha seguido se basa, igualmente, en las investigaciones de Hopcroft y Tarjan (1973,b). Comprendería las siguientes operaciones:

- Aplicación de una "depth first search" sobre la componente biconexa para transformarla en una palmera.
- Utilización de nuevo de esta técnica para obtener todos los caminos disjuntos en dicha componente.
- Determinación del circuíto original y de cada uno de los segmentos.
- Exploración de los segmentos en orden decreciente a su generación para la obtención de los pares de separación.
- Clasificación de los arcos y ramas de la palmera en clases de separación correspondientes a cada uno de los pares de separación.
- Elaboración de las componentes escindidas.
- Determinación de las componentes triconexas a partir de las escindidas.

Hay muchas formas de escindir un grafo y determinar sus componentes escindidas. Éstas, por tanto, no tienen porqué ser necesariamente únicas. Responderán a unos de estos tres tipos. Figura 5.

(3) Sólo tendrá sentido esta descomposición cuando la componente biconexa tenga más de tres vértices.

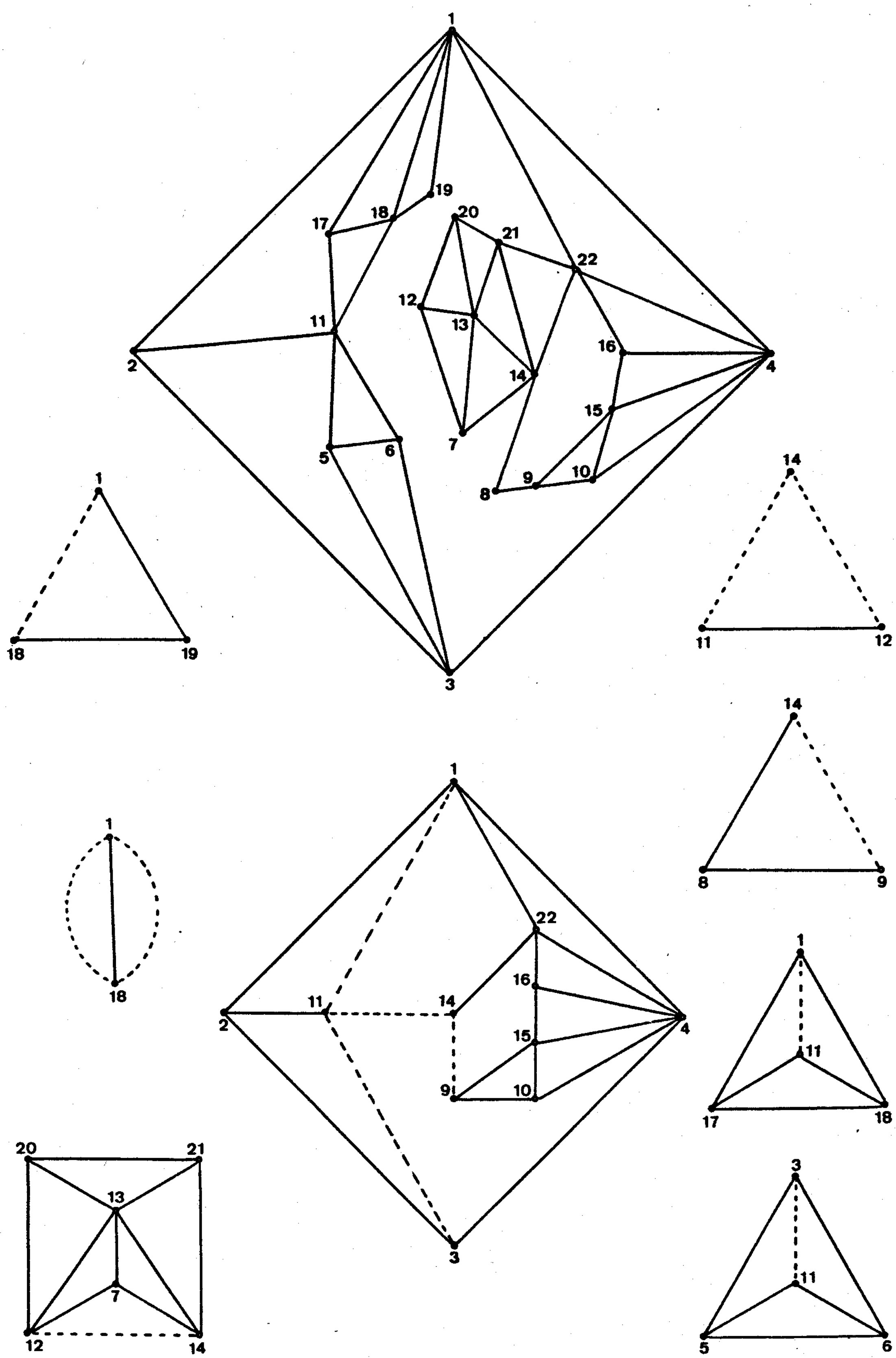


Figura 5

- Bucles triples: B_3 ({a,b}, {(a,b),(a,b),(a,b)})

- Triángulos: T ({a,b,c}, {(a,b), (b,c), (c,a)})

- Componentes triconexas.

Una vez que se han obtenido las componentes triconexas para cada componente, podemos fijar un método de adición progresiva de restricciones en dos estados sucesivos.

Primer estado: Adición de las aristas necesarias para convertir cada componente biconexa en un subgrafo triconexo.

Segundo estado: Inclusión de aristas para pasar de varios subgrafos triconexos articulados a un sólo grafo triconexo.

Para la representación en el plano usamos un algoritmo formulado a partir de otro que propone Tutte (1963) para grafos triconexos. Se basa en el teorema de Fary: 'Un grafo planar puede ser trazado en el plano de forma que sus aristas sean segmentos de línea recta y no se crucen'.

El algoritmo completo realiza las siguientes operaciones sobre una componente (grafo o subgrafo) triconexa:

- Genera un circuito de forma que dos vértices del mismo, no contiguos, no estén enlazados por ninguna arista. Si en la componente analizada figuran algunos vértices exteriores, éstos pertenecerán a dicho circuito. En el caso particular de que incluya a los cuatro puntos cardinales, el circuito estará formado sólo por estos cuatro vértices y las aristas que los unen.

- Se disponen los vértices de dicho circuito según los vértices de un polígono regular con el mismo número de lados que aristas tiene el circuito.

- Se elaboran las ecuaciones que permiten obtener las coordenadas de los restantes vértices. Para ello, se impone que cada vértice ha de localizarse en el centroide de sus vértices adyacentes.

Para un vértice V_i:

$$\rho_i x_i = \sum_{j=1}^{|V|} a_{ij} x_j$$

$$\rho_i y_i = \sum_{j=1}^{|V|} a_{ij} y_j$$

siendo:

$A (a_{ij})$: matriz de adyacencia de la componente

ρ_i : valencia del vértice V_i

- Se resuelve, por último, el sistema de ecuaciones lineales correspondiente por el método de eliminación de Gauss. Se obtiene, pues, una representación plana de la componente con la propiedad adicional de que todas las regiones finitas, en las que queda subdividida, tienen forma convexa. A partir de los distintos trazados planos podemos efectuar la introducción de las aristas en los dos estados citados:

Primer estado:

Si alguna componente triconexa posee una arista virtual, la primera operación que hay que realizar es transformarla en real (4). Las aristas de cada componente biconexa podemos entonces clasificarlas en uno de estos tres tipos: aristas pertenecientes a una componente triconexa, aristas que enlazan dos componentes triconexas, y aristas no enmarcadas en los tipos anteriores. Figura 5.

Para fusionar dos componentes triconexas en una sola hemos de introducir aristas que las enlacen. Su número estará en función del grado de relación que haya entre ambas. Pueden presentarse los siguientes casos.

1.- Hay sólo una arista que une a las dos componentes triconexas.

2.- Existen dos aristas de enlace y no tienen vértices comunes.

3.- Dos o más aristas ligan a las dos componentes pero todas tienen un vértice en común.

4.- Tres o más aristas relacionan a dos componentes y todas menos una tienen un vértice en común.

La adición de aristas sería distinta en cada una de ellas. Se haría así (5):

- Caso 1: Se incluirán dos aristas más que unan a las dos componentes. No podrán tener vértices comunes entre ellas, ni con la arista de enlace precedente. Figura 6.

(4) Esta operación no supone limitación alguna a la generalidad, pues, como veremos posteriormente, se presenta a lo largo del proceso la oportunidad de sustituir una relación introducida por otra distinta.

(5) Su justificación está en las definiciones de vértices de separación y componentes triconexas.

Figura 6

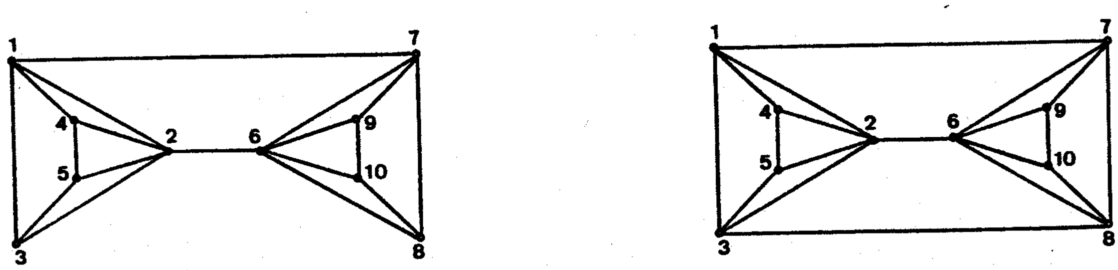


Figura 7

Figura 8

Figura 9

- Caso 2: Se añadirá una nueva arista entre las dos componentes. No podrá poseer por extremos ninguno de los vértices sobre los que inciden las restantes aristas de enlace. Figura 7.

- Caso 3: Se introducirán dos nuevas aristas que relacionen a las dos componentes. Sus vértices extremos serán distintos, y distintos también al vértice común de las dos aristas originales. Figura 8.

- Caso 4: Se tendrá que añadir una nueva arista entre componentes. No podrán incidir sobre el vértice común de las aristas en enlace ni a ninguno de los extremos de la única arista de enlace que no incide sobre el vértice común. Figura 9.

Después de ser fusionadas dos componentes triconexas, obtenemos, una nueva componente triconexa. Aplicándole las operaciones descritas podría integrarse con otra componente triconexa. La repetición sucesiva llevaría a una componente triconexa única. No obstante, puede suceder que no incluya a todos los vértices y aristas de la componente biconexa. En este caso subdividiremos las aristas restantes en dos clases: aristas aisladas y triángulos. A éstas las podemos considerar componentes triconexas "contraídas" y aplicar operaciones similares a las descritas con anterioridad.

La adición de aristas, en todo caso, se hará conforme a las posibilidades que ofrece la aplicación del teorema de la curva de Jordan. Figuras 5 y 10.

Segundo estado:

Una vez convertidas cada una de las componentes biconexas en subgrafo triconexo, se trazarían en el plano conforme al procedimiento descrito. A partir de estas representaciones se -

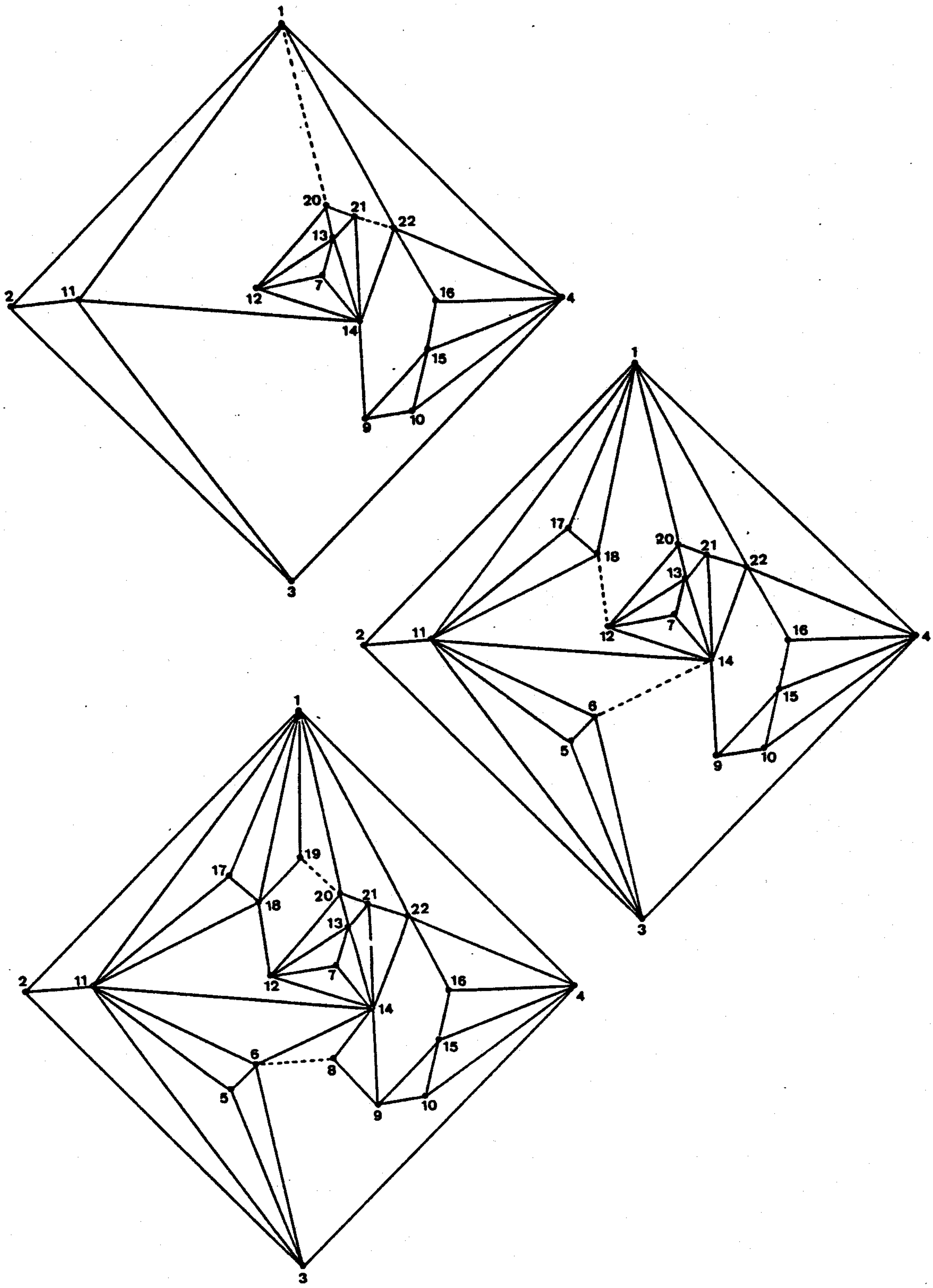

1
20
21
22
13
2
11
12
7
16
4
14
15
9
10
3
1
17
20
21
18
22
13
2
11
12
7
16
4
14
6
15
5
9
10
3
1
19
17
20
21
18
22
13
2
11
12
7
16
4
14
6
15
8
5
10
9
3

Figura 10

introducen otras nuevas limitaciones para transformar el grafo en triconexo.

Si queremos englobar dos subgrafos que tienen en común un vértice de separación, hemos de añadir dos aristas cualesquiera que los enlacen y no alteren la planaridad. Estas aristas no podrán tener un vértice extremo común ni incidir en el vértice de separación.

Las componentes biconexas arista aislada o triángulo pueden ser consideradas como subgrafos triconexos "contraídos" y tratarse de igual modo.

La característica especial de los grafos que manejamos, de tener siempre un circuito exterior formado por los cuatro puntos cardinales y las adyacencias entre ellos, hace que sea apropiado realizar el ensamble de las componentes biconexas, convertidas en subgrafos triconexos, a partir precisamente de ese circuito exterior.

El grago, G_4, después de esta etapa será pues triconexo.

III.1.5. Segunda comprobación de la planaridad

Como para hacer triconexo el grafo ha sido necesario añadirle aristas, hay que volverlo a someter al test de planaridad (cfr. III.1.3) por si se ha errado en la introducción de alguna de ellas. En el caso de que así sea, tendremos que sustituir aristas pertenecientes a segmentos no "empotrados" por otros, observando, de nuevo, los trazados planos conseguidos en la etapa anterior.

El grafo geométrico G_5, triconexo y planar, podrá ser entonces representado en su totalidad usando el algoritmo gráfico.

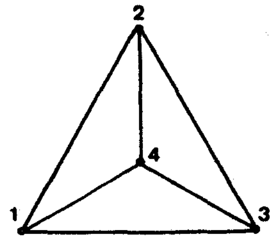


Figura 11

III.1.6. Comprobación de rectangularidad

Puesto que se impone la limitación de que tanto el contorno del esquema como cada uno de los locales sea rectangular, se hace necesario comprobar que el grafo lo permite, tras las - limitaciones que sucesivamente se han ido imponiendo.

Como quedó dicho en II.1.1, diversos àutores han estudiado - este problema, dejando establecido que el grafo cumplirá la condición de rectangularidad si no contiene ningún "grafo - completo de cuatro vértices". Figura 11.

El algoritmo que permite hacer esta comprobación utiliza la matriz de adyacencia, $A(a_{ij})$. Se trata de buscar un vértice V_i en cuya fila existan tres términos, a_{ij}, a_{ik} y a_{il}, distintos de cero; V_i es entonces adyacente a V_j, V_k y V_l. Si - en las filas j, k y l ocurre que los términos a_{ji}, a_{jk}, a_{jl}; a_{ki}, a_{kj}, a_{kl} y a_{li}, a_{lj}, a_{lk} son todos distintos de cero, - entre los vértices citados, V_i, V_j, V_k y V_l existe un grafo completo.

La comprobación se simplifica por la naturaleza simétrica de la matriz de adyacencia. Basta, pues, que para toda cuaterna i, j, k, l, se cumpla que a_{ij}, a_{ik}, a_{il}; a_{ji}, a_{jk}, a_{jl} sean todos distintos de cero. En caso contrario, el grafo G_6 cumple el test de rectangularidad.

Cuando el grafo no pasa el test, el programa avisa qué vértices son los causantes para que el operador revise consecuentemente las relaciones entre ellos.

III.1.7. Triangulación del grafo

Tras las transformaciones sufridas hata ahora, el grafo G_6 tiene ya una representación única pero puede corresponderle un número indefinido de duales. Para definir una solución única es preciso transformarlo aún en un "grafo planar máximo".

Este grafo cumple al límite la condición de Euler, $|A| = 3\ |V| - 6$, y todas sus caras tienen valencia tres. Es lo que se llama una "triangulación", según los teoremas de Baglivo y Graver (1983):

1.- Un grafo planar que es una triangulación para una de sus representaciones en el plano es un grafo planar máximo.

2.- Todas las representaciones en el plano de un grafo planar máximo con más de dos vértices son triangulaciones.

3.- Un grafo planar con más de dos vértices es un grafo planar máximo si y sólo si $|A| = 3\ |V| - 6$.

4.- A una triangulación le corresponde una disposición arquitectónica fundamental.

La característica especial del grafo que manejamos, con un circuito exterior formado por cuatro aristas que expresan las relaciones entre puntos cardinales, impide poder triangular todas las caras del grafo, pues no podemos incluir una arista que ligue a dos puntos cardinales no contiguos. La región exterior del grafo será, por tanto, la única que no podrá tener valencia tres, y la fórmula de Euler quedaría corregida así: $|A| = 3\ |V|\ -7$.

El procedimiento para lograr una triangulación es el siguiente:

1.- Introducción de nuevas relaciones de orientación. Aumentará así el número de locales definidos como "exteriores" y se indicarán, si no se hizo antes, cuáles van a ocupar las esquinas en el esquema de planta (orientaciones compuestas).

2.- Introducción de nuevas adyacencias entre locales.

3.- Comprobación de la condición de triangulación citada. Si es $|A| > 3\,|V| - 7$, hemos añadido demasiadas aristas y, en consecuencia, el grafo ya no es planar. Si $|A| < 3|V| - 7$, no se han introducido relaciones suficientes, deben seguirse añadiendo aristas.

Hecho ésto, se puede volver a trazar el grafo G_7, ya triangulado, sin necesidad de modificar las coordenadas de los vértices puesto que su número no ha variado.

Las aristas añadidas no hacen irreversible el proceso: se pueden hacer modificaciones siempre que no cambie el número de aristas.

III.1.8. Transformación de coordenadas de vértices

Antes de la obtención del grafo dual de G_7, y para que la disposición arquitectónica fundamental se ajuste a las condiciones señaladas para la forma del contorno y de los locales, conviene transformar las coordenadas de los vértices que representan a los distintos espacios (los del grafo g_7, subgrafo de G_7). El grafo transformado será g_8 y su dual, g_8^* lo será también de G_7 (considerando la región exterior), siendo por tanto el esquema adimensional buscado.

La transformación de coordenadas se efectúa según estos cri-

Figura 12

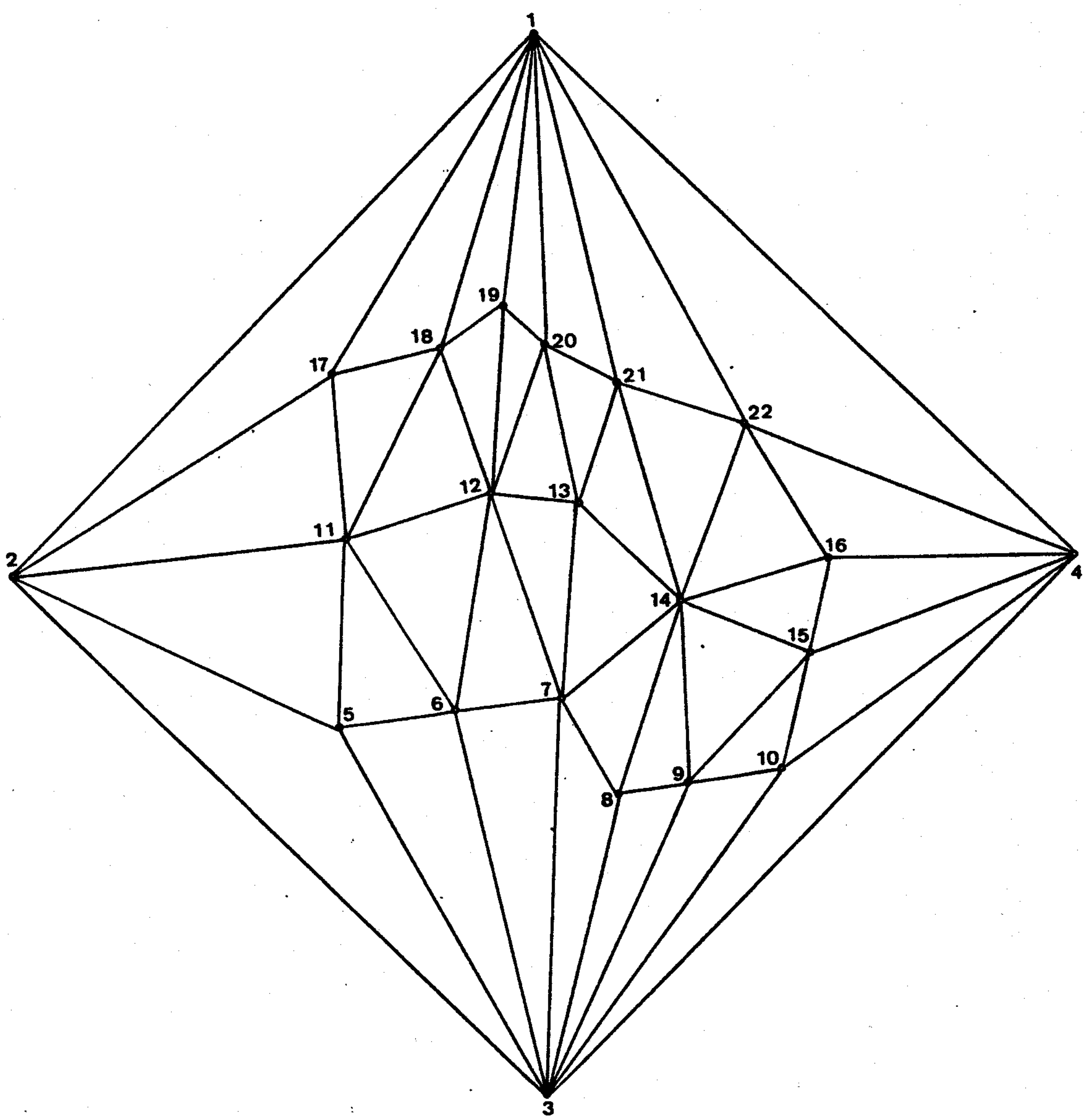


Figura 13

terios:

- Todas las aristas del grafo transformado serán ortogonales. Para ello, a un vértice de g_7 puede corresponderle más de uno en g_8.

- El contorno será rectangular, correspondiendo una orientación a cada uno de sus lados. Los vértices sobre un mismo lado representan locales de igual orientación, y los de las esquinas espacios con doble orientación.

- Los vértices que representan los puntos cardinales, pertenecientes al grafo G_7 pero no al g_7, servirán para "orientar" la transformación.

Para hacer que todas las aristas sean ortogonales, lo que simplifica enormemente la obtención del grafo dual, se opera un desdoblamiento de vértices cuando es necesario, como se ve en la figura 12.

El procedimiento completo se describe a continuación y se ilustra con una aplicación sobre el grafo de la figura 13.

1. Obtención del circuito exterior del grafo g_7, a partir del vértice nordeste y en sentido horario. Se opera sobre la matriz de incidencia.

2. El circuito exterior contiene cuatro tramos o caminos, comprendidos entre los cuatro vértices de doble orientación. Se toma como "lado inicial" aquél de estos tramos que contenga mayor número de vértices.

3. Los vértices del lado inicial se colocan uniformemente espaciados sobre un segmento de recta y se les asignan coordenadas. Figura 14.

4. Los vértices adyacentes a los del lado inicial se sitúan

Figura 14

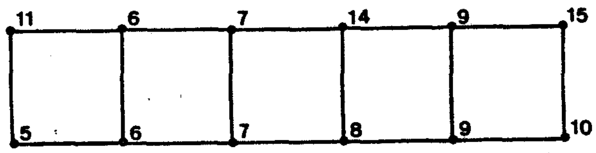


Figura 15

en una recta paralela, formando un nuevo lado enlazado al - inicial mediante aristas.

Para que un vértice pueda ser transformado ha de ser el único adyacente a un determinado vértice del lado inicial. El - resto del lado se completará con la repetición de los vértices que aparecen en el lado inicial y que no son adyacentes a ningún vértice con esas características. Así pues, como ya se ha dicho, a un vértice en el grafo g_7 le puede corresponder más de uno en el transformado g_8 con distintas representaciones. Figura 15.

5. Test 1. Una vez generado este nuevo lado hay que efectuar una primera comprobación por si hay que corregirlo.

Sucederá cuando dos vértices, adyacentes en el grafo g_7, no aparecen como tales en el presente lado. Este caso sólo se - puede dar cuando ya han sido transformado -y están en lados anteriores- los vértices que aparecen entre los dos afectados y que impiden esa adyacencia. Procederemos entonces a la supresión de éstos. Esto no supone alteración alguna de las condiciones de adyacencia por la forma de extracción del - dual que se describe en la etapa siguiente. Figura 16.

6. Test 2. Con este test se comprueba si se ha transformado ya algún vértice que pertenezca al tramo de orientación opuesta al lado inicial del grafo g_7. Se señalarán los que cumplan estas condiciones.

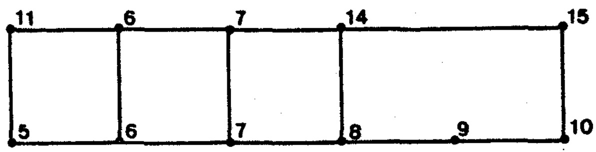


Figura 16

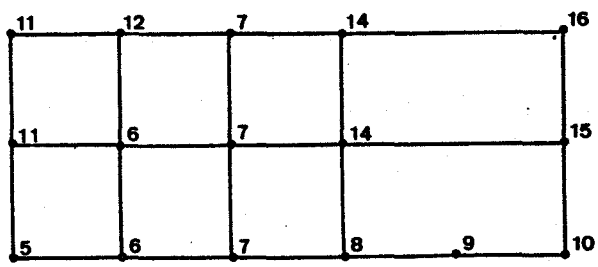


Figura 17

7. Transformación de coordenadas de vértices adyacentes a - los del lado generado en último lugar y que no se han transformado aún.

Si un vértice es adyacente a un solo vértice del último lado, y éste no ha sido marcado por el test 2, aparece transformado en el nuevo lado sobre una recta paralela al último. Se - completa este nuevo lado con la repetición de los vértices - correspondientes del lado anterior. Figura 17.

8. Test 3. Corrección si aparecen dos lados sucesivos con - los mismos vértices.

Si no se ha podido generar ningún nuevo lado a partir del último, querrá decir que no hay ningún vértice -no transformado aún- que sea el único adyacente a uno de los vértices del lado en cuestión.

Tendremos que proceder entonces al desdoblamiento de uno de los vértices del último lado. De esta forma puede haber ya - dos vértices que sean adyacentes a ese vértice duplicado. Si, a pesar de esta operación, sigue sin poderse generar un nuevo lado, habrá que efectuar un nuevo desdoblamiento del mismo vértice, siendo entonces tres los vértices que pueden - ser adyacentes al vértice elegido. Se procedería así hasta - la consecución de un nuevo lado.

El desdoblamiento de un vértice de un determinado lado obliga también a duplicar los vértices afectados situados en lados paralelos y que han sido generados con anterioridad. Sólo de esta forma se conserva la ortogonalidad de las aristas.

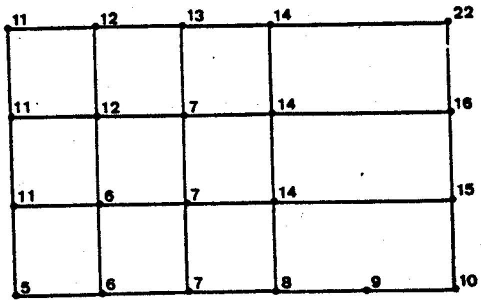
Para el desdoblamiento de los vértices se elegirán primero - los vértices extremos del último lado. Si éstos pertenecen - al lado opuesto al inicial (6) se elegirán entonces uno cualquiera de los interiores que no pertenezcan a ese lado y que sea adyacente a algún vértice no transformado. La razón de - elegir en primer lugar los vértices extremos obedece a la necesidad de obtener como vértices extremos del lado opuesto - al inicial los vértices "doblemente orientados".

9. Se repetirán cíclicamente las operaciones 5,6,7 y 8 hasta haber transformado todos los vértices del grafo g_7. Se obtiene entonces el grafo g_8. Figura 18.

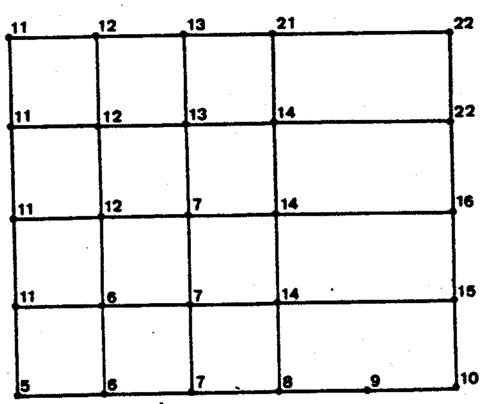
El grafo g_8 resultante tendrá mayor número de vértices y - aristas que el grafo g_7, pues, como ya se ha dicho, un vértice del grafo g_7 puede dar lugar a más de uno en el transformado g_8 (expansión del vértice); todos ellos estarán unidos mediante aristas.

---

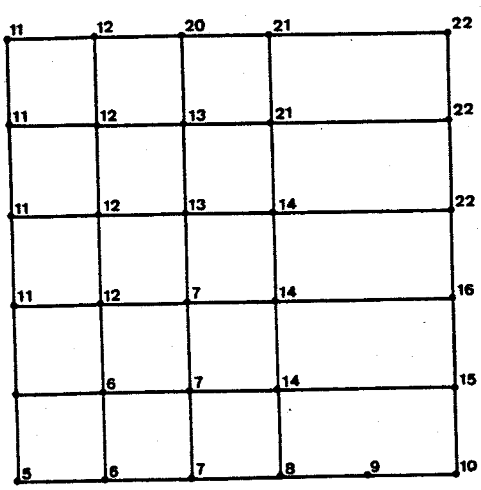
(6) Esto querrá decir que a partir de ellos no puede generarse ningún - otro.



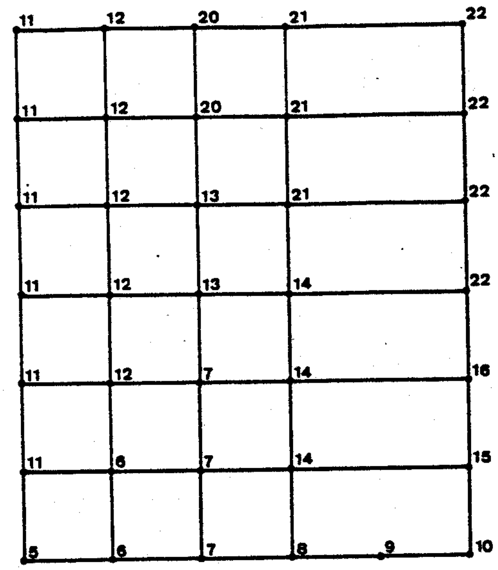
18.a



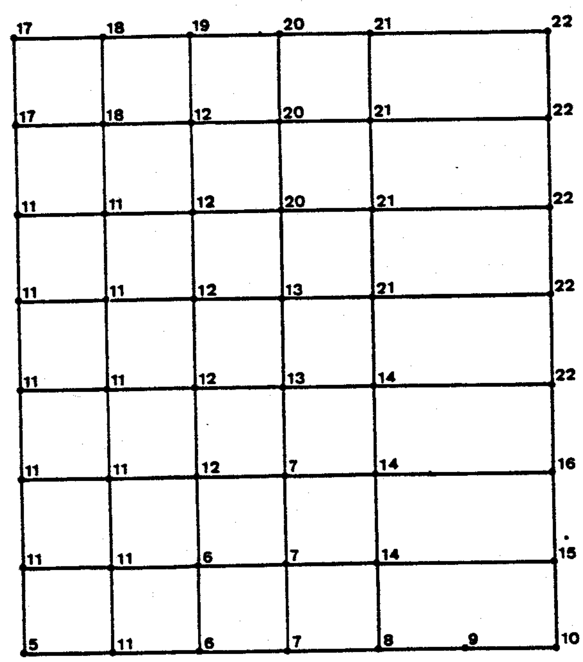
18.b



18.c



18.d



18.e

Figura 18

G.σ - grupo transformado

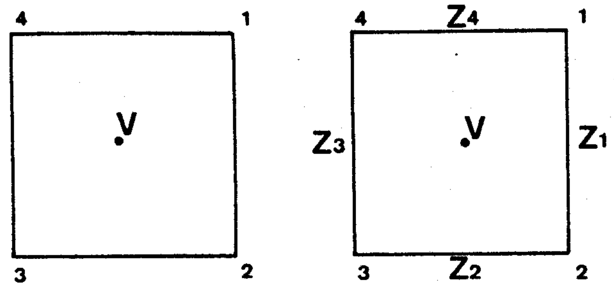


Figura 19

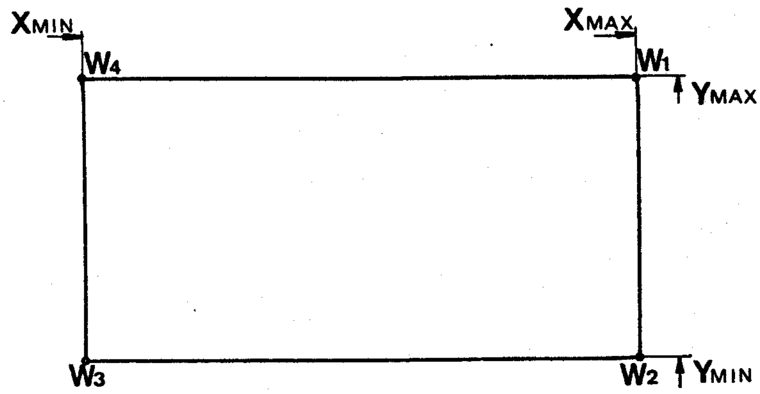


Figura 20

III.1.9. <u>Obtención del grafo dual</u>

La obtención del grafo dual a partir del grafo transformado - g_8 es inmediata. Se extrae el grafo dual g_8^* asignando una región rectangular a cada uno de los vértices del grafo g_8.

Cada región rectangular se definirá por las coordenadas de - sus cuatro vértices extremos. A cada uno de ellos le corresponden dos coordenadas pero al estar relacionadas son sólo - cuatro las magnitudes que hay que determinar para cada vértice en g_8 . Figura 19.

Para obtenerlas realizamos las siguientes operaciones:

1. Determinamos las coordenadas mínimas y máximas, X_{min}, X_{max}, Y_{min}, Y_{max} del grafo g_8 que, lógicamente, son la de los vértices de las esquinas. Figura 20.

2. Obtenemos los cuatro paráme tros correspondientes a los - vértices interiores del grafo g_8. Aquellos vendrán definidos por las coordenadas del - vértice analizado y de sus - vértices adyacentes. Si dos - vértices adyacentes tienen la misma abcisa, el parámetro co rrespondiente se logra por la semisuma de la abcisa de ambos. Si tienen igual ordenada, la semisuma de las ordenadas nos dará entonces aquél. Figura 21.

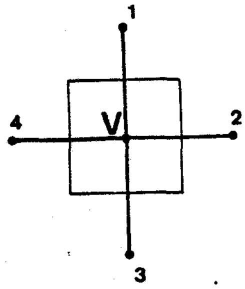


Figura 21

Si un vértice interior no es - adyacente a otros cuatro vérti ces, el parámetro correspondiente que falte -a lo sumo uno- se obtiene sumándole a su abcisa u ordenada, según el ca so, la mitad de la distancia - entre el lado al cual pertenece y el lado al que éste dio - lugar. Figura 22.

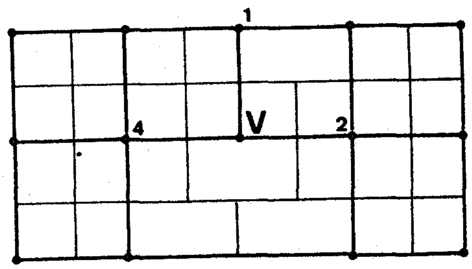


Figura 22

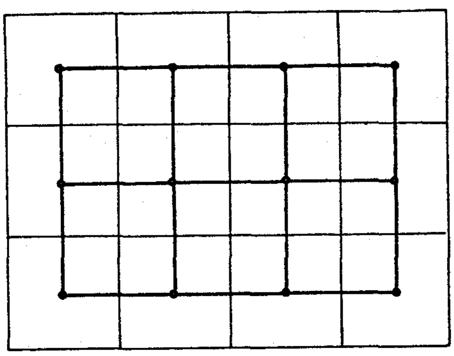


Figura 23

3. Calculamos los parámetros - correspondientes a los vértices exteriores del grafo g_8. El proceso será el mismo que - en el caso anterior salvo para el parámetro correspondiente a la orientación -u orientaciones en el caso de vértice en esquina-. Este vendrá determinado -

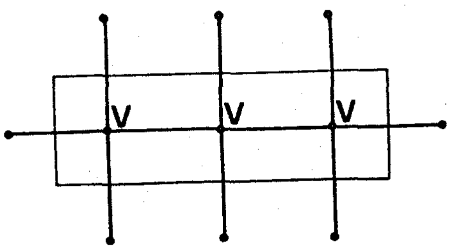


Figura 24

por las coordenadas X_{max}, X_{min}, Y_{max}, Y_{min}, obtenidas anteriormente, sumándoles o restándoles, según el caso, una determinada magnitud fija, para lograr que el contorno sea rectangular.

4. La unión de todas las regiones, correspondientes a cada uno de los vértices del grafo g_8, nos daría el dual de éste g_8^*. Al ser la distancia entre dos lados paralelos, y sucesivos en su generación, una magnitud constante, la forma de consecución del dual nos garantiza la no existencia de regiones que no correspondan a ninguno de los vértices del grafo geométrico. Además, la intersección de dos regiones, correspondientes a vértices adyacentes en dicho grafo, nos dará siempre la arista común. Figura 23.

Si suprimimos en dicho grafo dual g_8^* la arista intersección de dos regiones que responden a un mismo vértice en el grafo g_8, obtendremos un grafo g_9, que podemos llamar G_9 si tenemos en cuenta la región exterior. Este grafo geométrico G_9 será el esquema en planta adimensional buscado. Figura 24.

Este esquema, que es adimensional, cumple todas las relaciones de adyacencia y orientación impuestas (y es la única solución que las cumple), es rectangular en su contorno y en los espacios componentes, y no tiene espacios vacíos.

El esquema trazado puede ser el punto final del proceso. Pero podemos decidir que la solución no es satisfactoria, y desear introducir modificaciones. Esto puede hacerse cuantas veces se quiera, anulando una o varias relaciones y sustituyéndolas por un número igual de relaciones nuevas. Si ésto se hace a la vista del esquema obtenido, que es suficientemente expresivo, no se alterará la planaridad ni la triconectividad del grafo. Cuando el esquema sea satisfactorio podrá darse por terminado el proceso.

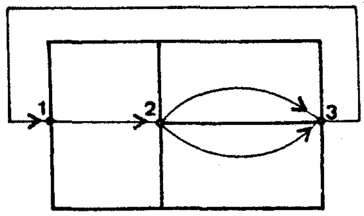


Figura 25.a

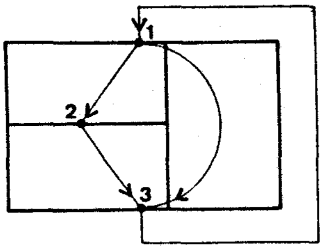


Figura 25.b

III.2. DIMENSIONAMIENTO

El procedimiento para dimensionar esquemas de contorno y componentes rectangulares comprende las etapas siguientes:

III.2.1. Sustitución por dos grafos dirigidos

El esquema se sustituye por dos grafos dirigidos, uno "horizontal" y otro "vertical".

Los vértices de grafo "vertical" representan a dos segmentos verticales del esquema, que son cerramientos y particiones paralelos entre sí. Figura 25.a.

La región exterior y los locales se sustituyen por arcos que unen los vértices correspondientes, orientados de izquierda a derecha en los locales y en sentido contrario en la región exterior.

Procedimiento análogo se seguirá para el grafo "horizontal" a partir de los segmentos horizontales del esquema y orientando de arriba abajo los arcos que representan locales, y de abajo arriba el de la región exterior. Figura 25.b.

Cuando se desea que los límites de dos espacios adyacentes estén en prolongación, ambos límites se definen por un solo vértice en el grafo que proceda. Figura 26. Esto es especialmen-

Figura 26

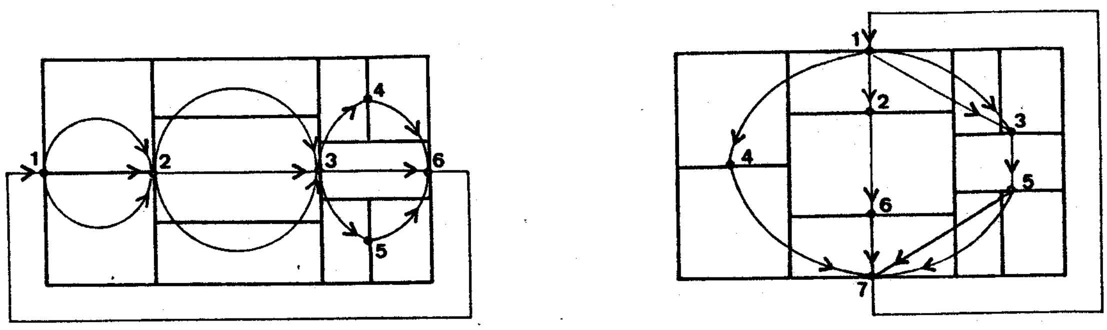


Figura 27

te importante en el contorno del esquema. Figura 27.

Como puede verse, estos grafos dirigidos pueden no ser simples, al contener arcos múltiples o paralelos. Serán en general fuertemente conexos.

El número de vértices de cada grafo igualará al de segmentos de contorno y particiones en la dirección correspondiente, y el de arcos se corresponderá con el número de locales, más el que representa la región exterior.

Ambos grafos vendrán definidos por su matriz de incidencia B (b_{ij}) de dimensión $(|A|,|V|)$, cuyos términos serán $b_{ij}=0$ si el arco $\underline{i}$ no incide en el vértice $\underline{j}$; $b_{ij} > 0$ si el vértice $\underline{j}$ es origen del arco $\underline{i}$, y $b_{ij} < 0$, cuando el vértice $\underline{j}$ sea el extremo del arco $\underline{i}$. Estos valores distintos de cero se definirán más adelante.

Cualquier criterio de numeración es válido para señalar los vértices. La numeración de los arcos estarán en correspondencia con la utilizada para designar los vértices en el problema adimensional. El arco que representa a la región exterior se numerará en último lugar.

III.2.2. Definición de las matrices de adyacencia

Elaborados los dos grafos dirigidos, han de fijarse las adyacencias entre locales, especialmente en cuanto sean relaciones de acceso. Se ha considerado que dos estancias pueden comunicarse si tienen al menos 1 m. de contorno común (hueco de paso).

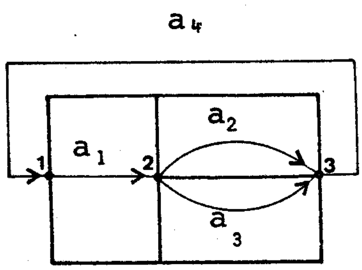
Este concepto de adyacencia, entendido como accesibilidad, viene definido en la correspondiente matriz A (a_{ij}) de dimensión $(|A|,|A|)$, cuyos términos serán $a_{ij} = 0$ si no se establece accesibilidad entre los locales <u>i</u> y <u>j</u>, y $a_{ij} = 1$ en caso positivo. Como el acceso es simétrico, será $a_{ij} = a_{ji}$, y A $\{a_{ij}\}$ será simétrica.

III.2.3. Restricciones métrico-geométricas

Las limitaciones más usuales que en la práctica se imponen a los locales de una distribución suelen ser las dimensiones lineales o superficies, mínimas o máximas, que cada espacio deba tener.

Las dimensiones mínimas y/o máximas de los locales o del contorno, que son restricciones lineales, se reflejan en la matriz de incidencia del grafo dirigido que corresponde a la dirección en que opera cada limitación.

Si se introducen limitaciones de superficie, que son no lineales, el problema deberá ser resuelto por programación no lineal; estas limitaciones pueden introducirse indistintamente en una u otra de las dos matrices de incidencia.



$y_1 = y_2 + y_3$

$y_1 = y_4$

$x_1 + x_2 = x_4$

$x_2 = x_3$

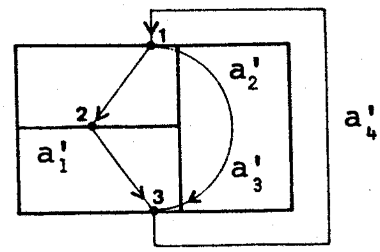


Figura 28

III.2.4. Expresiones que traducen las restricciones

Las limitaciones dimensionales dan lugar a ecuaciones o inecuaciones lineales, lo mismo que las condiciones de accesibilidad, como en seguida veremos. Las limitaciones de superficie dan - ecuaciones o inecuaciones no lineales.

Las limitaciones dimensionales se traducen en un sistema de - ecuaciones y/o inecuaciones lineales, aplicando la primera ley de Kirchhoff para redes eléctricas. Como es sabido, esta ley - dice que, en cada vértice de la red, la suma algebraica de las intensidades de corriente de las ramas concurrentes es nula. Traducida a nuestro grafos dirigidos, la primera ley de Kirchhoff diría que la suma de los valores dimensionales (longitud o anchura de un local) atribuídos a los arcos que concúrren en un vértice es igual a la de los que parten de él. Figura 28.

De este modo, en el grafo horizontal, se haría: $B_1 \cdot X = 0$. En esta expresión $B_1 = \{b'_{ij}\}$ es una matriz obtenida trasponiendo la de incidencia, a la que previamente se ha suprimido la última columna (que es combinación lineal de las demás), y dividido - cada término por su valor absoluto (con lo que B_1 sólo contendrá valores 0,1 y -1). X es una matriz columna, cuyos términos x_1, x_2 ... x_m representan las longitudes de cada local y del - contorno.

Del mismo modo, en el grafo vertical se hará $B_2 \cdot Y = 0$. Donde -

B_2 es una matriz obtenida de la de incidencia del grafo vertical por el mismo procedimiento que se siguió para B_1. Y donde Y es una matriz columna cuyos términos son los anchos de locales y contorno.

Para elaborar las inecuaciones que expresan las condiciones de accesibilidad (valores no nulos en la matriz de adyacencia de cada grafo) se procede del modo siguiente en cada uno de los dos grafos dirigidos:

- Como cada local se define por un arco, se busca el vértice común a los dos espacios adyacentes (término positivo y negativo, respectivamente, en la matriz de incidencia).

- Se buscan, a continuación, todos los arcos que concurren en ese vértice.

- Se analiza el centro geométrico de cada local en el esquema adimensional. (7)

- En el grafo horizontal, $\underline{i}$ representará el local cuyo centro geométrico tiene mayor ordenada, y convendremos en que el local $\underline{j}$ es accesible al $\underline{i}$.

Las inecuaciones que expresan la accesibilidad entre los locales $\underline{i}$ y $\underline{j}$, serán:

$$X_i + \Sigma X_{SUPi} - \Sigma X_{SUPj} \geq 1$$

$$X_j - \Sigma X_{SUPi} + \Sigma X_{SUPj} \geq 1$$

X_i, X_j longitud de los locales $\underline{i}$ y $\underline{j}$

(7) Esto no supone un contrasentido, como veremos después, sólo servirá para situar 'relativamente' unas estancias respecto de otras.

X_{SUPi}, X_{SUPj}: longitud del local cuyo centro geométrico tiene su abcisa X mayor que el de i, j.

El término independiente 1 indica la anchura del hueco de paso mínimo.

En el grafo dirigido vertical, llamamos local i a aquel cuyo centro geométrico tiene mayor abcisa X. Convendremos que el local j es accesible al i.

Las inecuaciones que expresan la accesibilidad entre dichos locales serán:

$$Y_i + \Sigma\ Y_{SUPi} - \Sigma\ Y_{SUPj} \geq 1$$

$$Y_j - \Sigma\ Y_{SUPi} + \Sigma\ Y_{SUPj} \geq 1$$

Y_i, Y_j anchura de los locales i y j

Y_{SUPi}, Y_{SUPj}: anchura del local cuyo centro geométrico tiene su ordenada Y mayor que el de i, j.

Si todos los términos de la ecuación son positivos se anula esta restricción por ser obvia. (8)

En un esquema de espacios y contornos rectangulares, estas inecuaciones expresan siempre las condiciones de accesibilidad entre dos locales cualesquiera. En efecto, podemos entender estas expresiones, para el grafo horizontal, considerando sólo dos grandes espacios: uno, el que comprende al local i y a todos los que tienen centro de abcisa mayor que él; y otro, el

(8) En las restricciones métrico-geométricas, se impondrá una dimensión para cada estancia que, lógicamente, será mayor que la longitud de un hueco de paso.

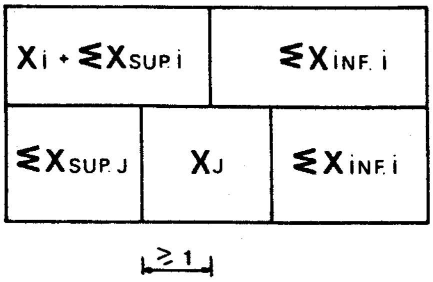


Figura 29.a

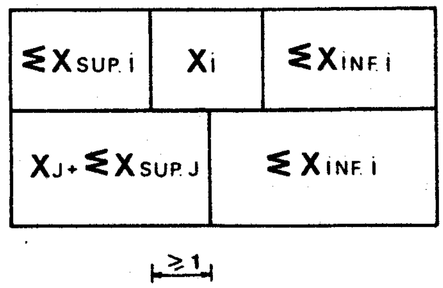


Figura 29.b

que contiene todos los espacios con centro de abcisa mayor que la de j. Figura 29.a.

Para la segunda inecuación se contempla la otra posibilidad: un gran espacio que comprende a todos los locales con centros de abcisa superior a la de i, y otro que englobe al j y a todos los locales con centros de abcisa superior a j. Figura - 29.b.

La operación realizada puede hacerse análogamente en el grafo vertical.

Estas inecuaciones pueden expresarse de otra forma.

En efecto: Figura 29

$$X_i + \Sigma X_{SUPi} = L - \Sigma X_{INFi} \Rightarrow \Sigma X_{SUPi} = L - X_i - \Sigma X_{INFi}$$

$$X_j + \Sigma X_{SUPj} = L - \Sigma X_{INFj} \Rightarrow \Sigma X_{SUPj} = L - X_j - \Sigma X_{INFj}$$

Las inecuaciones quedarían:

$$X_i + L - X_i - \Sigma X_{INFi} - L + X_j + \Sigma X_{INFj} = X_j - \Sigma X_{INFi} + \Sigma X_{INFj} \geq 1$$

$$X_j + L - X_j - \Sigma X_{INFj} - L + X_i + \Sigma X_{INFj} = X_i - \Sigma X_{INFj} + \Sigma X_{INFi} \geq 1$$

L: Longitud del contorno.

$X_{INF\,i}$, $X_{INF\,j}$: longitud de un gran espacio que comprende a aquellos cuyos centros tienen abcisa menor que la del local i o j.

Igualmente podríamos expresar estas inecuaciones para el otro grafo dirigido. Unas y otras, como puede observarse, son similares.

El tercer y último tipo de requisitos corresponde a las restricciones de superficie impuestas a los locales y al contorno. Podemos clasificarlas así:

a) Restricciones para el problema lineal:

Acotación superior e inferior de las variables longitud y anchura de locales y del contorno:

$$a_i \geq x_i \geq a'_i$$

$$b_i \geq y_i \geq b'_i$$

Si se quiere dar a una variable un valor fijo, bastará con hacer iguales sus límites superior e inferior.

Los valores límites a_i, a'_i, b_i, b'_i están recogidos en la matriz de incidencia de los grafos dirigidos.

Si alguna variable no está acotada inferiormente se le asignará un valor mínimo igual a 1 m.

b) Restricciones para el problema no lineal:

Acotación inferior de la longitud y anchura de cada local y del contorno:

$x_i \geq l_i$

$y_i \geq l'_i$

Los valores l_i, l'_i vienen reflejados en las matrices de incidencia de los grafos dirigidos.

Al igual que en el caso anterior, si una variable no está - limitada inferiormente, se le asignaría un valor mínimo.

En realidad estas inecuaciones pretenden fijar las dimensiones mínimas de locales y del contorno, sólo que dicho valor, si se quiere, puede ser distinto en una dirección que en - otra.

Acotación inferior de la superficie de cada una de las estancias y del contorno.

$x_i \cdot y_i \geq s_i$

Los valores s_i estarán reflejados en la matriz de incidencia de los grafos dirigidos.

Todas las restricciones impuestas, lineales y no lineales, figurarán en una matriz: la matriz de restricciones $R=(r_{ij})$ de - dimensión $(|R|, 2\cdot|A|)$, siendo:

$|R|$: número de restricciones.

$|A|$: número de locales más uno (correspondiente al contorno).

Los términos independientes de cada una de las restricciones - se localizarán en una matriz-columna; el tipo (9) de cada una de ellas se expresará en una matriz-fila.

(9) El tipo de una restricción se refiere a su forma de expresión, según se trate de ecuación o inecuación, y dentro de esta última, si es del género mayor o igual, o menor o - igual.

III.2.5. Naturaleza y costes de la función-objetivo

Las funciones-objetivo previstas son las siguientes:

- Funciones-objetivo lineales:

1) Minimizar la longitud o anchura del contorno. Su expresión sería: Minimizar x_m.

2) Con una longitud o anchura del contorno determinada, obtener estas mismas variables para cada uno de los locales, con arreglo a su importancia, referida ésta a sus dimensiones relativas dentro del esquema. Su expresión sería:

$$\text{Minimizar} \sum_{i=1}^{m-1} c_i \cdot x_i$$

c_i: coste que indica la importancia relativa del local i. Por tratarse de un problema de minimización se le asignará un valor negativo.

x_i: longitud o anchura de locales exclusivamente.

Para utilizar este último tipo de problema lineal, habrá que definir una nueva restricción métrico-geométrica muy particular: se fijará la dimensión del contorno.

- Funciones-objetivo no lineales:

1) Minimizar la superficie del contorno. Su expresión sería: Minimizar $x_m \cdot y_m$.

2) Con una superficie del contorno determinada, obtener la superficie de cada local, de acuerdo con su entidad dimensional relativa. Su expresión sería:

$$\text{Minimizar} \sum_{i=1}^{m-1} c_i \cdot x_i \cdot y_i$$

Para emplear este segundo tipo de problema no lineal, habrá que definir previamente la superficie del contorno.

III. 2.6. Resolución del problema por los métodos de programación

El dimensionamiento es un problema de programación, donde se busca un resultado óptimo y cuya estructura general consta de una función-objetivo, que se quiere maximizar o minimizar, sujeta a una serie de restricciones.

Un problema de programación lineal es aquél que consta de - una función-objetivo lineal sujeta a una serie de restricciones, también lineales. Cuando, o bien la función-objetivo o alguna de las restricciones -o ambas cosas a la vez- sean no lineales, la programación será no lineal. En el caso particular de que la función objetivo sea no lineal y todas las restricciones sí lo sean, la programación no lineal recibe el - nombre de cuadrática.

Al ser siempre expresiones lineales las derivadas de aplicar la primera ley de Kirchhoff a los dos grafos dirigidos, y - las que reflejan la accesibilidad entre espacios, la linealidad o no linealidad del problema vendrá dada: por un lado, por la función objetivo; por otro, por las restricciones métrico-geométricas.

Pasemos a describir los dos métodos de resolución para los - problemas de programación.

a) Programación lineal:

Su estructura general será:

$$\begin{array}{lll} \text{Min.} & C \cdot X & \\ \text{s.a.} & A \cdot X = 0 & \text{.......... (1ª ley de Kirchhoff)} \\ & B \cdot X - 1 \geq 0 & \text{.......... (accesibilidad)} \\ & X^T \geq D & \\ & X^T \leq E & \text{.......... (métrico-geométricas)} \end{array}$$

Para la resolución del problema de programación lineal se ha utilizado un algoritmo clásico conocido con el nombre de - "simplex". (Larrañeta, 1977).

Aplicando el método de resolución indicado, se obtendrá la - longitud -variable en una dirección- independientemente de - la anchura -variable en la dirección ortogonal- de cada local y del contorno.

b) Programación no lineal:

Su estructura general será:

$$\begin{array}{lll} \text{Min.} & C \cdot X \cdot Y & \\ \text{s.a.} & A \cdot X = 0 & \\ & & \text{......... (1ª ley de Kirchhoff)} \\ & A' \cdot Y = 0 & \\ & B \cdot X - 1 \geq 0 & \\ & & \text{......... (accesibilidad)} \\ & B' \cdot Y - 1 \geq 0 & \\ & X^T \geq D & \\ & Y^T \geq D' & \text{......... (métrico-geométricas)} \\ & X \cdot Y \geq E & \end{array}$$

Para la resolución del problema no lineal se ha utilizado el algoritmo de Polak-Ribière (1971), que utiliza el método de los gradientes conjugados para minimizar funciones no necesariamente convexas. Se ha demostrado asímismo (Rao, 1978), - que cualquier método de minimización que usa las direcciones conjugadas es convergente de forma cuadrática, lo que disminuye de forma considerable el número de iteraciones hasta - llegar al óptimo.

Se ha utilizado el algoritmo de Polak-Ribière y no el tradicional de Fletcher-Reeves porque aquél supone una mejora de este último. En efecto, Polak (1971) -citando un estudio de Lootsme- indica que los métodos de penalización de funciones, exterior e interiormente, convergen más rápidamente que aquéllos que sólo penalizan exterior o interiormente. En consecuencia, su método es una versión de aquél que usa los gradientes conjugados con penalización de la función objetivo, tanto exterior como interiormente.

Aplicando ese método de resolución se obtienen las dos variables -longitud y anchura-, para cada local y el contorno, - conjuntamente.

Para ajustar aún más los resultados obtenidos por este último método, una vez obtenidos los valores para las distintas variables se eliminan las restricciones no lineales. Se aplica entonces el algoritmo "simplex" al ser ya un problema lineal. Los valores de las cotas inferiores serán ahora los resultados obtenidos para las distintas variables. Las restricciones no lineales serán satisfechas entonces y, por tanto, pueden ser eliminadas. Al ser el algoritmo "simplex" más potente los resultados obtenidos se ajustarán mejor.

Una vez resuelto el problema de dimensionamiento por los métodos de programación adecuados a cada caso, podemos dibujar el esquema dimensionado, concluyendo entonces el proceso completo de su generación.

La posibilidad teórica de generar y optimizar esquemas de distribución de espacios había sido el punto de partida de nuestra tesis; posibilidad formulada con anterioridad por diversos autores.

Tras el análisis riguroso de los procesos sugeridos por las investigaciones sobre el tema y su crítica en función de los objetivos, el trabajo que presentamos consiste, esencialmente, en la propuesta de un nuevo método de generación y optimización completo y coherente.

La elaboración de este método lleva consigo el desarrollo de algunos algoritmos completamente originales, así como la adaptación de otros tradicionalmente aplicados.

El procedimiento que proponemos utiliza la técnica clásica de representación mediante el grafo dual.

El conjunto de algoritmos desarrollado para la generación de esquemas adimensionales ofrece una serie de ventajas: no limita el número de espacios componentes; es un proceso dinámico en el que las condiciones pueden introducirse de modo progresivo; es, además, un método flexible que permite en todo momento la revisión de las condiciones impuestas; y por último, es aplicable de un modo general -cualquiera que sea la estructura del grafo de partida-.

La comprobación de la planaridad y K-conectividad del grafo se basa en los estudios de Hopcroft y Tarjan. Los algoritmos mencionados permiten analizar de modo progresivo cada una de estas propiedades. Se ha utilizado el algoritmo que desarrolla Tutte para el trazado de grafos triconexos, con las adaptaciones pertinentes para una adecuada acción integrada usuario-ordenador.

El procedimiento empleado para la obtención del grado dual es una aportación original que consideramos de interés.

El problema de dimensionamiento de esquemas se aborda siguiendo una concepción clásica, recogida al menos teóricamente por la mayor parte de los investigadores: la de considerarlo como problema de optimización, empleando técnicas de programación lineal y no lineal.

Las condiciones de accesibilidad entre locales vienen expresadas por inecuaciones para cuya elaboración se propone un algoritmo nuevo. Los restantes requisitos o bien se formulan directamente o se establecen por algoritmos ya conocidos.

Se ha formulado también un procedimiento iterativo, basado en los estudios de Polak, para la resolución de problemas mediante programación no lineal, que dimensiona eficazmente los esquemas de distribución, con ajuste y trazado del mismo.

El trabajo desarrollado en esta tesis pretende, en suma, servir de aportación a las formulaciones teóricas necesarias para la revisión de la utilidad del ordenador en el diseño arquitectónico. Con ello contribuimos, tal vez, a la incorporación de este poderoso instrumento a una disciplina que hasta ahora se ha mantenido apartada de su inexorable influencia.

ANEXO: I — ALGUNOS CONCEPTOS DE LA TEORÍA DE GRAFOS

1. GRAFO

El concepto intuitivo de grafo es el de unos puntos unidos mediante líneas. Lo que un grafo expresa no son relaciones métricas de posición de los puntos o forma de las líneas, sino topológicas: qué pares de puntos son los que están en relación.

Un grafo G, pues, se define por una pareja de conjuntos. El de los puntos o *vértices*, conjunto finito y no vacío V (G), y el de relaciones o *aristas* A (G), que puede ser vacío, y cuyos elementos son pares desordenados de vértices.

Si los vértices tienen asignadas coordenadas tenemos un *grafo geométrico*. Si no, un *grafo abstracto*.

2. SUBGRAFO

Un subgrafo G' de un grafo G es otro grafo cuyos vértices son V (G') $\subseteq$ V (G) y cuyas aristas son A (G') $\subseteq$ A (G) y unen sólo vértices del conjunto V (G').

3. VÉRTICES Y ARISTAS

Dos vértices son *adyacentes* cuando están unidos por una arista.

Una arista es *incidente* a un vértice si lo tiene como extremo.

Si más de una arista incide en la misma pareja de extremos, se trata de *aristas paralelas o múltiples*. Si una arista une un vértice a sí mismo, es un *bucle o lazo*.

Un grafo sin aristas múltiples ni bucles es un *grafo simple*. Un grafo simple en el que todos los vértices son adyacentes es un *grafo completo*.

4. CAMINOS Y CIRCUITOS

Un *camino elemental* es una secuencia de aristas de un grafo, de forma que cada una de ellas tenga un vértice común con la precedente, salvo la primera, y otro con la siguiente, salvo la última. Un camino elemental cerrado, porque los vértices inicial y final de la secuencia coincidan, reciben el nombre de *circuito elemental*.

Por *longitud* de un camino o de un circuito entendemos el número de aristas que lo forman.

5. CONECTIVIDAD

Un grafo será *conexo*, si entre todo par de vértices existe - un camino. En caso contrario será *inconexo*.

Componente conexa de un grafo es el subgrafo que contiene un vértice dado y todos los vértices que pueden enlazarse a él mediante un camino.

Grafo por supresión de vértice G-v es el que se obtiene de G eliminando al vértice v y las aristas incidentes a él.

Grafo por supresión de arista G-a es el que se obtiene de G eliminando la arista a.

Si G es un grafo conexo y G-v no lo es, el vértice v es un *vértice de corte o separación.*

Un grafo es *biconexo* si no tiene vértices de corte.

Si G es un grafo conexo, una *componente biconexa* será el subgrafo biconexo que contenga el mayor número posible de vértices y aristas.

Si G es un grafo biconexo y contiene cuatro vértices distintos, de modo que cualquier camino que se pueda establecer entre ellos pasa necesariamente por dos vértices determinados, estos dos últimos constituyen un *par de separación.*

Si el grafo es biconexo y no contiene ningún par de separación, es además *triconexo.*

Una *componente triconexa* de un grafo biconexo G, será un subgrafo triconexo de éste con el máximo número posible de vértices y aristas.

Si (a,b) es un par de separación del grafo biconexo G y dividimos las aristas en clases de equivalencia: A_1, A_2, A_3,..., A_n, de manera que dos aristas sólo podrán estar en una misma clase; si existe un camino que las contenga y que no pase por a o b -salvo como vértices extremos-, estas clases serán, asímismo, *de separación* respecto del par (a,b).

Si (a,b) es un par de separación del grafo biconexo G y A_1, A_2, ..., A_n las clases de separación del mismo, los *grafos* -

escindidos G_1 (V_1, A_1) y G_2 (V_2, A_2) de G respecto del par (a,b) serán aquellos que cumplan:

$V_1 = V(A')$, $A_1 = A' \cup (a,b)$

$V_2 = V(A'')$, $A_2 = A'' \cup (a,b)$

siendo:

$A' = \bigcup_{i=1}^{k} A_i$ $\quad A'' = \bigcup_{i=k+1}^{n} A_i$ $\quad$ (a,b): arista que incide sobre los vértices a y b.

Las nuevas aristas añadidas a G_1 y G_2: (a,b) se denominan *aristas virtuales.*

En general, podemos decir que un grafo con al menos K+1 vértices es *K-conexo* si no puede ser hecho inconexo por la supresión de K-1, o menos vértices, y las aristas que inciden en ellos.

La conectividad de G, expresada por K (G) es el mayor valor de *K* para el cual G es K-conexo.

6. PLANARIDAD

Un grafo es *planar* cuando puede ser representado en el plano de forma que no se intersecten geométricamente dos aristas en puntos distintos de los vértices.

Si un grafo es planar y la inclusión de una arista entre dos vértices cualesquiera -que no sea un bucle o una arista paralela- lo convierte en no planar, se tratará de un *grafo planar máximo.*

Por *valencia de un vértice* entendemos el número de aristas que inciden en él. *La valencia de una cara o región* en un grafo geométrico, será el número de aristas que la limitan.

Un grafo geométrico conexo y planar donde cada una de sus regiones tiene valencia tres, es una *triangulación*.

7. DUALIDAD

Si G es un grafo geométrico conexo y planar, puede extraerse a partir de él su grafo dual G*. Cada región del primero vendrá caracterizada por un vértice en el segundo y viceversa. Una arista que relaciona dos vértices en el grafo G indicará que las dos regiones correspondientes, en el grafo dual G*, serán adyacentes.

8. GRAFO DIRIGIDO

Grafo dirigido D{V (D), A (D)} es aquel cuyas aristas A (D), ahora llamadas *arcos*, son pares ordenados de vértices.

Un arco tiene, pues, un *vértice origen*, V, y un *vértice extremo*, W. Ambos vértices son *adyacentes* y el arco será *incidente desde V* e *incidente a W*.

Los arcos que, partiendo de un mismo vértice origen, inciden a un mismo extremo se denominan *arcos paralelos* y el arco que une un vértice a sí mismo es un *bucle*. Cuando el grafo no contiene arcos paralelos ni bucles, es un *grafo dirigido simple*.

El *camino* en un grafo dirigido es una secuencia de arcos tales que el extremo de cada uno es origen del siguiente.

Cuando el extremo del último arco es a la vez origen del primero estamos ante un *circuito*.

Un grafo dirigido D será *fuertemente conexo* si para todo par de vértices, *V* y *W*, del mismo hay un camino en el grafo D desde el primero al segundo.

El *grafo subyacente* del D es el G que tiene sus mismos vértices y cuyas aristas coinciden con los arcos de aquél.

9. ÁRBOLES

Un *árbol* es un grafo conexo que tiene al menos dos vértices y ningún circuito.

Si un grafo A' es subgrafo del árbol A, será también un *subárbol* del mismo.

Si G es un grafo y A es un árbol y a la vez un subgrafo G - que contiene a todos sus vértices, diremos que A es un *árbol extendido sobre* el grafo G.

Un *árbol dirigido* será aquél grafo dirigido cuya estructura subyacente es un árbol. Idénticas definiciones podemos dar - para *subárbol dirigido* y *árbol extendido sobre un grafo dirigido.*

Un *árbol dirigido con origen* se caracteriza por haberse distinguido de los demás uno de los vértices, llamado origen. Cualquier vértice de este árbol puede ser alcanzado por un - camino que parte del vértice origen; ningún arco será incidente al vértice origen y sólo habrá un arco que incida al - resto de los vértices del grafo.

Dado un grafo dirigido D, podemos establecer una partición - del mismo en dos clases distintas: un conjunto de arcos del árbol extendido sobre D, y un conjunto de *ramas* que completan el número de arcos del grafo dirigido D. Si llamamos *V* - al vértice origen de una rama y *W* a su vértice extremo, y -

existe un camino en el árbol extendido sobre D que va desde W hasta V, y esto ocurre para cada una de las ramas, a este grafo dirigido se le denomina *palmera*.

En este anexo se presentan los dos programas que se han realizado y puesto a punto para cumplimentar los procesos descritos: generación de esquemas distributivos y dimensionamiento de los mismos.

Los dos programas se han realizado en lenguaje FORTRAN IV para un ordenador Hewlett Packard de la serie 1000-M, con un sistema operativo RTE-IV.B.

Para cada uno de los dos procedimientos se ofrece en este anexo un organigrama general en el que se destaca sobre todo la interacción usuario-ordenador, un listado y algunos ejemplos que se han implementado utilizando estos programas.

A.II.1. GENERACIÓN DE ESQUEMAS ADIMENSIONALES

A.II.1.1. Manual de usuario

Con carácter general han de tenerse en cuenta los siguientes criterios:

1. Los datos pueden escribirse en cualquier posición de una tarjeta o en pantalla, separados por caracteres "blanco" ( ), o "coma" (,). En el caso de utilizar la "coma" como separador, sólo podrá emplearse una para separar dos datos consecutivos.

2. Entre los datos, pueden incluirse tantos comentarios como se deseen, con la condición de que figure en la primera - columna el carácter barra (/).

   Sin embargo, este carácter no debe ser utilizado en la expresión que se emplea para titular el problema.

3. Aquellos datos que puedan tomar un valor real pueden escribirse con o sin punto decimal. En último caso, la conversión la efectúa el programa en el momento de la lectura.

4. Aquellos datos que vengan referidos por una lista de variables necesitarán tantas tarjetas o registros como variables haya. Al final de la lista se introducirá en otro registro un "0" para indicar que se ha terminado con esa orden de lectura.

5. La entrada de datos puede realizarse por tarjetas, pantalla o cinta magnética.

La secuencia de tarjetas o registros para la entrada de datos del programa PLP es costante a lo largo del mismo por su interactividad. Comprendería una lectura previa y una lectura a la vista de los mensajes que el ordenador envía.

La lectura previa se realiza introduciendo los siguientes datos:

1. Titulación del problema.

   FORMATO : (un máximo de 80 caracteres alfanuméricos)

2.a Identificación de los espacios. (a = 1,2, ... n; n = n°de espacios)

   FORMATO : (un máximo de 80 caracteres alfanuméricos)

3.a Matriz de incidencia (a = 1,2, ... n; n = n°de espacios)

   FORMATO: LA, LO, LF.

   LA : n° de la arista (a partir del 5)
   LO : n° de un vértice extremo (a partir del 5)
   LF : n° del otro vértice extremo (a partir del 5)

La lectura a lo largo del programa responde a los mensajes - que el ordenador envía. Fundamentalmente es de dos tipos según se trate de añadir nuevas aristas o de sustituir aristas ya introducidas:

1.a Introducción de nuevas aristas. (a = 1,2, ... m; m = n°de aristas introducidas)

   FORMATO: LA, LO, LF.

   LA : n° de la arista (a partir de la última que se había numerado)
   LO : n° de un vértice extremo.
   LF : n° del otro vértice extremo.

2.a Sustitución de aristas (a = 1,2, ... h; h = n° de aristas sustituídas)

FORMATO: LA, LO, LF, MO, MF.

LA : n° de la arista que se sustituye.

LO : n° de un vértice extremo de la arista, antes de la sustitución .

LF : n° del otro vértice extremo.

MO : n° del vértice extremo de la arista después de la - sustitución.

MF : n° del otro vértice extremo.

Una vez que se ha obtenido una solución el ordenador pregunta si se desea terminar o, por el contrario, se quiere continuar. En el primer caso se introduce un: "0",en el segundo - un: "1".

Además de la entrada de datos propiamente dicha el usuario - presenta una serie de opciones para la salida de resultados que conviene destacar. Estas operaciones se les ofrece al - pulsar distintos "siwtchs" del ordenador. Son las siguientes:

SIWTCH 1 : Se desea imprimir las coordenadas de los grafos o subgrafos geométricos trazados.

SIWTCH 2 : No se quiere trazar en el "plotter" el grafo o subgrafo geométrico analizado.

SIWTCH 3 : No se desea trazar el grafo ya triangulado.

SIWTCH 4 : Se indica que se trace el grafo triangulado - después de transformadas las coordenadas de - sus vértices.

SIWTCH 5 : Se desea imprimir las coordenadas de los vértices del grafo dual.

A lo largo de la ejecución del programa el ordenador puede enviar mensajes de errores. Obecederán a las siguientes causas:

ERROR 1 : El número de vértices o aristas es mayor que el permitido.

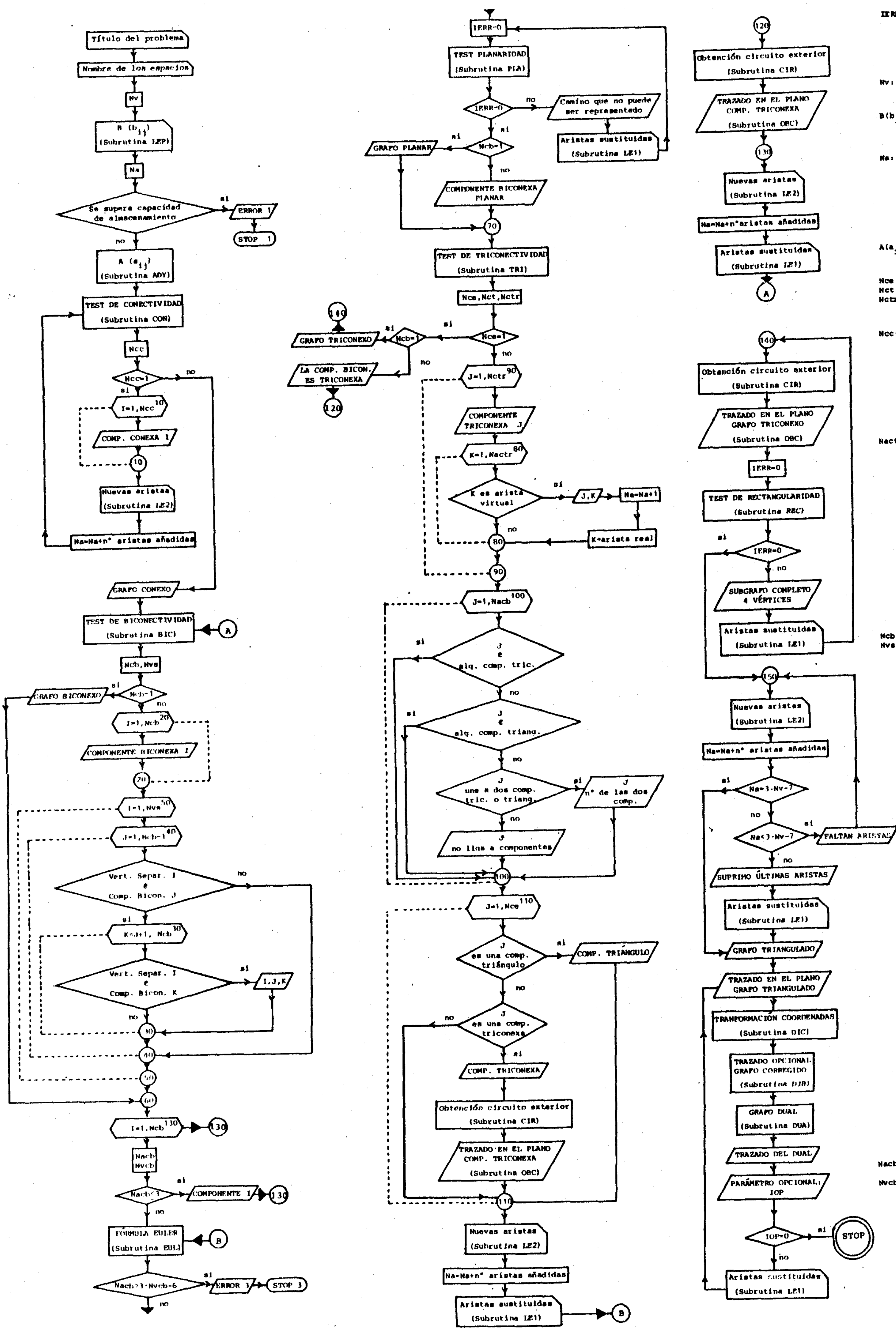
ERROR 2 : La valencia de un vértice es mayor de lo que se ha previsto.

ERROR 3 : La componente biconexa analizada no cumple la relación de Euler y, por tanto, no es planar.

ERROR 4 : No existe un circuito de vértices orientados al exterior.

El programa que se presenta está elaborado para un número máximo de 150 aristas y 50 vértices.

A.II.1.2. Organigrama general



IERR: variable de error.

Nv: n° de vértices.

B(b_{ij}): matriz de incidencia.

Na: n° de aristas.

A(a_{ij}): matriz de adyacencia.

Nce: n° de componentes escindidas.
Nct: n° de componentes triángulo.
Nctr: n° de componentes triconexas.

Ncc: n° de componentes conexas.

Nactr: n° de aristas de la componente triconexa.

Ncb: n° de componentes biconexas.
Nvs: n° de vértices de separación.

Nacb: n° de aristas de la componente biconexa.
Nvcb: n° de vértices de la componente biconexa.

A.II.1.3. Listado del Programa

```
FTN4
$EMA(IJK,3)
C
C
C      ***********
       PROGRAM PLP
C      ***********
C
C -----------------------------------------------------------------------
C        PROGRAMA PARA GENERACION DE ESQUEMAS DISTRIBUTIVOS EN PLANTA
C -----------------------------------------------------------------------
C
C PROGRAMA PRINCIPAL ................................. LECTURA DE DATOS
C
C TABLAS UTILIZADAS:
C
C IB(150,50) ... MATRIZ DE INCIDENCIA
C IA(100,50) ... MATRIZ DE ADYACENCIA
C IAA(50,10) ... MATRIZ DE ADYACENCIA ORDENADA
C
C
C LISTADO DE ERRORES:
C
C ERROR 1 ...... EL NUMERO DE VERTICES O ARISTAS ES MAYOR QUE EL PERMI-
C                TIDO.
C ERROR 2 ...... LA VALENCIA DE UN VERTICE ES MAYOR DE LO QUE SE HA PRE-
C                VISTO.
C ERROR 3 ...... LA COMPONENTE BICONEXA ANALIZADA NO CUMPLE LA RELACION
C                DE EULER Y, POR TANTO, NO ES PLANAR.
```

```
C ERROR 4 ...... NO EXISTE UN CIRCUITO DE VERTICES ORIENTADOS AL EXTE -
C                RIOR.
C
C
C ESCRITURAS OPCIONALES:
C
C SWITCH 1 ..... SE DESEA IMPRIMIR LAS COORDENADAS DE LOS GRAFOS O SUB-
C                GRAFOS GEOMETRICOS TRAZADOS.
C SWITCH 2...... NO SE DESEA TRAZAR EL GRAFO O SUBGRAFO GEOMETRICO ANA-
C                LIZADO.
C SWITCH 3 ..... NO SE DESEA DIBUJAR EL GRAFO YA TRIANGULADO.
C SWITCH 4 ..... SE DESEA TRAZAR EL GRAFO TRIANGULADO TRANSFORMADO.
C SWITCH 5 ..... SE DESEA IMPRIMIR LAS COORDENADAS DE LOS VERTICES DEL
C                GRAFO DUAL.
C
C
      COMMON IPAR(4),NAME(3)
      COMMON IENT,ISAL,IEAUX,ISAUX,NV,NA,KKTT
      COMMON NOM(40),JAUX(2200)
      COMMON /IJK/ IB(150,50),IA(100,50),X(300),Y(300),IAA(50,10)
C
C PARAMETROS DE ENTRADAS Y SALIDAS
C
      CALL RMPAR(IPAR)
      IENT=IPAR(1)
      ISAL=IPAR(2)
      IEAUX=IPAR(3)
      ISAUX=IPAR(4)
      IF(IENT.EQ.0) IENT=1
      IF(ISAL.EQ.0) ISAL=6
      IF(IEAUX.EQ.0) IEAUX=1
      IF(ISAUX.EQ.0) ISAUX=6
C
C  ** LLAMADA A LA SUBRUTINA DE LECTURA **
C
      CALL LEP(IENT,ISAL,ISAUX,NV,NA,NOM,IB)
C
C  ** LLAMADA AL SEGMENTO PL1 QUE APLICA AL GRAFO EL TEST DE CONECTIVI-
C     DAD **
C
      NAME(1)=2HPL
      NAME(2)=2H1
      NAME(3)=2H
      CALL EXEC(8,NAME)
      STOP
      END
C
C
      SUBROUTINE LEP(IENT,ISAL,ISAUX,N,M,NOM,IB)
C     ****************************************
C
C SUBRUTINA PARA LECTURA DE DATOS
C
      EMA IB(150,50)
      DIMENSION NOM(40)
      WRITE(ISAL,1001)
      WRITE(ISAL,2001)
      READ(IENT,2002) NOM
      WRITE(ISAL,2003) NOM
C
```

```
C PUESTA A CERO
C
      DO 50 I=1,150
      DO 50 J=1,50
      IB(I,J)=0
   50 CONTINUE
      DO 60 I=1,40
      NOM(I)=2H
   60 CONTINUE
      WRITE(ISAL,4001)
      DO 80 I=5,60
      READ(IENT,2002) NOM
      IF(NOM(1).EQ.2H0 ) GO TO 85
      WRITE(ISAL,4002) I,NOM
      DO 75 J=1,40
      NOM(J)=2H
   75 CONTINUE
   80 CONTINUE
   85 N=I-1
C
C LECTURA DE LA MATRIZ DE INCIDENCIA
C
   90 IB(1,1)=1
      IB(1,2)=1
      IB(2,2)=1
      IB(2,3)=1
      IB(3,3)=1
      IB(3,4)=1
      IB(4,4)=1
      IB(4,1)=1
      WRITE(ISAL,3001)
      WRITE(ISAL,3002)
      DO 100 I=5,110
      READ(IENT,*)LA,LO,LF
      IF(LA.EQ.0) GO TO 105
      WRITE(ISAUX,5001) LA,LO,LF
      IB(LA,LO)=1
      IB(LA,LF)=1
  100 CONTINUE
  105 M=I-1
C
C CONTROL SI SE SUPERA LA CAPACIDAD DE ALMACENAMIENTO
C
      IF(M.LE.150.AND.N.LE.50) 10,20
   10 IF(3*N-7.LE.150) GO TO 30
   20 IERR=MER(ISAL,1)
      STOP 0001
   30 RETURN
C
C FORMATOS
C
 1001 FORMAT(1H1)
 2001 FORMAT(/1X"MENSAJE ::ED:: TITULA EL PROBLEMA"/)
 2002 FORMAT(40A2)
 2003 FORMAT(1X,40A2,/1X,79"*",//)
 3001 FORMAT(//1X"MENSAJE ::ED:: MATRIZ DE INCIDENCIA"//16X"LA NUMERACIO
     &N DE LOS VERTICES  Y ARISTAS SE HARA A PARTIR DEL 5."/16X"LOS 4 PR
     &IMEROS VERTICES Y ARISTAS CORRESPONDEN A LOS PUNTOS"/16X"CARDINALE
     &S Y SUS RELACIONES. ESTAN INTRODUCIDOS AUTOMATICAMENTE."/16X"AL FI
     &NAL DE LA LISTA INTRODUCE: 0")
 3002 FORMAT(//1X"  A  V  V"/1X"  -  -  -")
```

```
 4001 FORMAT(//1X"MENSAJE ::ED:: ASIGNE UN NOMBRE A CADA UNO DE LOS ESPA
     &CIOS"//16X"AL FINAL DE LA LISTA INTRODUCE:0"//1X"ESP  NOMBRE"/1X"-
     &--  ------")
 4002 FORMAT(I4,2X,40A2)
 5001 FORMAT(I4,2I3)
      END
C
C
      SUBROUTINE LE1(IEAUX,ISAL,ISAUX,K,IA,IB)
C     ****************************************
C
C SUBRUTINA QUE LEE LAS ARISTAS QUE SUSTITUYEN A OTRAS
C
      EMA IB(150,50),IA(100,50)
      WRITE(ISAL,3001)
   10 READ(IEAUX,*) LA,MD,MF,LD,LF
      IF(LA.EQ.0) GO TO 50
      WRITE(ISAUX,5001) LA,MD,MF,LD,LF
      IB(LA,MD)=0
      IB(LA,MF)=0
      IA(MD,MF)=0
      IA(MF,MD)=0
      IB(LA,LD)=(-1)**K
      IB(LA,LF)=(-1)**K
      IA(LD,LF)=1
      IA(LF,LD)=1
      GO TO 10
   50 RETURN
C
C FORMATOS
C
 3001 FORMAT(/16X"LOS DATOS SE DARAN EN EL SIGUIENTE ORDEN:"/16X"- NUMER
     &O DE LA ARISTA"/16X"- VERTICE PRIMITIVO"/16X"- VERTICE PRIMITIVO"/
     &16X"- VERTICE ACTUAL"/16X"- VERTICE ACTUAL"/16X"AL FINAL DE LA LIS
     &TA INTRODUCE: 0"//1X"  A  VP  VP  VA  VA"/1X"  -  --  --  --  --")
 5001 FORMAT(5I4)
      END
C
C
      SUBROUTINE LE2(IEAUX,ISAL,ISAUX,K,NA,NV,IA,IB)
C     **********************************************
C
C SUBRUTINA QUE LEE LAS NUEVAS ARISTAS INTRODUCIDAS
C
      EMA IB(150,50),IA(100,50)
      WRITE(ISAL,3001) NA
      WRITE(ISAL,3002)
   10 READ(IEAUX,*) LA,LD,LF
      IF(LA.EQ.0) GO TO 50
      WRITE(ISAL,5001) LA,LD,LF
      NA=NA+1
      DO 20 J=1,NV
      IB(NA,J)=0
   20 CONTINUE
      IB(NA,LD)=(-1)**K
      IB(NA,LF)=(-1)**K
      IA(LD,LF)=1
      IA(LF,LD)=1
      GO TO 10
   50 RETURN
```

```
C
C FORMATOS
C
 3001 FORMAT(//1X"MENSAJE ::SR:: EL NUMERO DE ARISTAS ACTUALES ES:"I4)
 3002 FORMAT(//1X"MENSAJE ::ED:: INTRODUCE LAS NUEVAS ARISTAS"//16X"AL F
     &INAL DE LA LISTA INTRODUCE: 0"//1X"   A   V   V"/1X"   -   -   -")
 5001 FORMAT(I4,2I3)
      END
C
C
      SUBROUTINE AD1(EV,A,B,FIC,PTR,V,E,IB)
C     ************************************
C
C SUBRUTINA QUE ALMACENA UN ELEMENTO NUEVO EN UN CONJUNTOAS
C
      EMA IB(150,50)
      INTEGER A,B,PTR,V,E,EV
      INTEGER FIC(EV)
      DO 50 I=1,E
      IF(IB(I,A).EQ.0.OR.IB(I,B).EQ.0) GO TO 50
      PTR=PTR+1
      FIC(PTR)=I
      GO TO 60
   50 CONTINUE
   60 RETURN
      END
C
C
      SUBROUTINE AD2(EE,A,B,FIC,PTR)
C     ******************************
C
C SUBRUTINA QUE ALMACENA DOS ELEMENTOS NUEVOS EN UN CONJUNTO
C
      INTEGER EE,A,B,PTR
      INTEGER FIC(EE)
      PTR=PTR+2
      FIC(PTR-1)=A
      FIC(PTR)=B
      RETURN
      END
C
C
      SUBROUTINE AD3(FF,II,JJ,KK,B,K)
C     *******************************
C
C SUBRUTINA QUE ALMACENA TRES ELEMENTOS EN UN NUEVO CONJUNTO
C
      INTEGER FF
      INTEGER B(K)
      B(FF+1)=II
      B(FF+2)=JJ
      B(FF+3)=KK
      FF=FF+3
      RETURN
      END
C
C
      SUBROUTINE SU2(I,FF,B,K)
C     ************************
C
C SUBRUTINA QUE SUPRIME DOS ELEMENTOS CONSECUTIVOS DE UN CONJUNTO
```

```
C
      INTEGER FF
      INTEGER B(K)
      IF(FF.LE.2) GO TO 20
      IF(I.EQ.FF.OR.I.EQ.0) GO TO 15
      DO 10 J=I-1,FF-2
      B(J)=B(J+2)
   10 CONTINUE
   15 B(FF)=0
      B(FF-1)=0
      FF=FF-2
      GO TO 30
   20 FF=0
      B(1)=0
      B(2)=0
   30 RETURN
      END
C
C
      SUBROUTINE SU3(I,FF,B,K)
C     ************************
C
C SUBRUTINA QUE SUPRIME TRES ELEMENTOS CONSECUTIVOS DE UN CONJUNTO
C
      INTEGER FF
      INTEGER B(K)
      IF(FF.LE.3) GO TO 20
      IF(I.EQ.FF.OR.I.EQ.0) GO TO 15
      DO 10 J=I-2,FF-3
      B(J)=B(J+3)
   10 CONTINUE
   15 B(FF)=0
      B(FF-1)=0
      B(FF-2)=0
      FF=FF-3
      GO TO 30
   20 FF=0
      B(1)=0
      B(2)=0
      B(3)=0
   30 RETURN
      END
C
C
      SUBROUTINE VA1(SAVE,KKK)
C     ************************
C
C SUBRUTINA DE OPERACION AUXILIAR
C
      INTEGER SAVE
      SAVE=KKK
      RETURN
      END
C
C
      SUBROUTINE VA2(MMM,NNN)
C     ***********************
C
C SUBRUTINA DE OPERACION AUXILIAR
C
```

```
      NNN=NNN
      RETURN
      END
C
C
      SUBROUTINE VA3(NNN,NNN,SAVE)
C     ****************************
C
C SUBRUTINA DE OPERACION AUXILIAR
C
      INTEGER SAVE
      NNN=NNN
      NNN=SAVE
      RETURN
      END
C
C
      SUBROUTINE CA1(NV,PTBJ1,PTBJ2,ESTA,IAA)
C     ***************************************
C
C SUBRUTINA QUE MODIFICA LA ORDENACION DE LA PLANARIDAD PARA ADAPTARLA A
C LA NECESARIA PARA EL ANALISIS DE TRICONECTIVIDAD
C
      ENA IAA(50,10)
      INTEGER U,W
      INTEGER PTBJ1(NV),PTBJ2(NV),ESTA(NV)
      DO 10 I=1,NV
      ESTA(I)=0
   10 CONTINUE
      DO 200 I=1,NV
      DO 100 J=1,9
      IF(IAA(I,J).EQ.0) GO TO 110
      W=IAA(I,J)
      IF(I.LT.W) GO TO 60
      ESTA(W)=2*W+1
      GO TO 100
   60 IF(PTBJ2(W).LT.I) ESTA(W)=2*PTBJ1(W)
      IF(PTBJ2(W).GE.I) ESTA(W)=2*PTBJ1(W)+1
  100 CONTINUE
  110 DO 150 J=2,9
      IF(IAA(I,J).EQ.0) GO TO 200
      W=IAA(I,J)
      U=IAA(I,J-1)
      IF(ESTA(W).GE.ESTA(U)) GO TO 150
      IAVE=IAA(I,J)
      IAA(I,J)=IAA(I,J-1)
      IAA(I,J-1)=IAVE
      GO TO 110
  150 CONTINUE
  200 CONTINUE
      RETURN
      END
C
C
      SUBROUTINE CA2(NV,II,NUMB,VECT,VAUX)
C     ************************************
C
C SUBRUTINA QUE CAMBIA EL ORDEN DE ALMACENAMIENTO DE UN VECTOR PARA HA-
C CERLA SEGUN LA ANTIGUA NUMERACION
C
      ENA VAUX(NV),VECT(NV)
```

```
      DIMENSION NUMB(NV)
      DO 10 I=1,NV
   10 VAUX(I)=0
      DO 100 I=1,II
      DO 50 J=1,NV
      IF(NUMB(J).NE.I) GO TO 50
      VAUX(J)=VECT(I)
      GO TO 100
   50 CONTINUE
  100 CONTINUE
      DO 150 I=1,NV
      VECT(I)=VAUX(I)
  150 CONTINUE
      RETURN
      END
C
C
      SUBROUTINE CA3(NV,VECT,NUENU,VAUX)
C     **********************************
C
C SUBRUTINA PARA CAMBIAR VECTORES DE ACUERDO A UNA NUMERACION
C
      INTEGER VECT(NV),VAUX(NV),NUENU(NV)
      DO 10 I=1,NV
   10 VAUX(I)=0
      DO 50 I=1,NV
      IF(VECT(I).LE.0) GO TO 50
      VAUX(NUENU(I))=VECT(I)
   50 CONTINUE
      DO 60 I=1,NV
      VECT(I)=VAUX(I)
   60 CONTINUE
      RETURN
      END
C
C
      SUBROUTINE CA4(NV,IAA,NUENU,VAUX,MM)
C     ************************************
C
C SUBRUTINA PARA CAMBIAR EL ORDEN DE LAS FILAS DE LA MATRIZ DE ORDENA -
C CION
C
      ENA IAA(50.10)
      INTEGER VAUX(10),NUENU(NV)
      DO 1 I=1,MM
    1 IAA(I,10)=0
      DO 5 I=1,NV
      DO 3 J=1,10
      IF(IAA(I,J).EQ.0) GO TO 5
      IF(IAA(I,J).LT.0) IAA(I,J)=-IAA(I,J)
    3 CONTINUE
    5 CONTINUE
      DO 10 I=1,10
   10 VAUX(I)=0
      K=0
   20 K=K+1
      IF(K.GT.MM) GO TO 70
      DO 30 I=1,9
      VAUX(I)=IAA(K,I)
   30 CONTINUE
```

```
      DO 40 I=1,NV
      IF(K.EQ.NUENU(I)) GO TO 50
   40 CONTINUE
   50 J=I
   51 IF(IAA(J,10).EQ.0) GO TO 55
      J=IAA(J,10)
      GO TO 51
   55 DO 60 I=1,9
      IAA(K,I)=IAA(J,I)
      IAA(J,I)=VAUX(I)
   60 CONTINUE
      IAA(K,10)=J
      GO TO 20
   70 RETURN
      END
C
C
      SUBROUTINE PRO(V,EE,EEV,PUNTO,VAL,FRENE,SIG,CABE)
C     ****************************************************
C
C SUBRUTINA QUE ELABORA EL CONJUNTO CABE PARA ALMACENAR LAS INCIDENCIAS
C
      INTEGER V,EE,EEV,FRENE,PUNTO,VAL
      INTEGER SIG(EEV),CABE(EE)
      FRENE=FRENE+1
      SIG(FRENE)=SIG(PUNTO)
      SIG(PUNTO)=FRENE
      CABE(FRENE-V)=VAL
      RETURN
      END
C
C
      SUBROUTINE TRA(V,E,EE,LISA,IB)
C     ******************************
C
C SUBRUTINA QUE ELABORA EL LISTADO DE ARISTAS POR SUS VERTICES EXTREMOS
C
      ENA IB(150,50)
      INTEGER V,E,EE
      INTEGER LISA(EE)
      K=0
      DO 100 I=1,E
      DO 50 J=1,V
      IF(IB(I,J).EQ.0) GO TO 50
      K=K+1
      LISA(K)=J
   50 CONTINUE
  100 CONTINUE
      RETURN
      END
C
C
      SUBROUTINE ORD(ISAL,NV,NA,K2,FROND,NUMB,PTBJ1,PTBJ2,FI,IB,IAA)
C     ****************************************************************
C
C SUBRUTINA QUE ORDENA LOS TERMINOS DE LA NUEVA MATRIZ DE ADYACENCIA  -
C USANDO EL D.F.S.
C
      ENA IB(150,50),IAA(50,10)
      INTEGER V,W
      INTEGER PTBJ1(NV),PTBJ2(NV),NUMB(NV),FI(NA),FROND(NA)
```

```
      DO 80 I=1,NA
      V=0
      W=0
      DO 30 J=1,NV
      IF(IB(I,J).LT.0.AND.V.EQ.0) GO TO 20
      IF(IB(I,J).LT.0.AND.V.NE.0) GO TO 40
      GO TO 30
   20 V=J
   30 CONTINUE
   40 W=J
      DO 50 J=1,K2
      IF(I.EQ.FROND(J)) GO TO 60
   50 CONTINUE
      IF(NUMB(V).LT.NUMB(W)) GO TO 55
      LL=V
      V=W
      W=LL
   55 IF(PTBJ2(W).GE.NUMB(V)) FI(I)=2*PTBJ1(W)
      IF(PTBJ2(W).LT.NUMB(V)) FI(I)=2*PTBJ1(W)+1
      GO TO 80
   60 IF(NUMB(V).GT.NUMB(W)) GO TO 65
      LL=V
      V=W
      W=LL
   65 FI(I)=2*NUMB(W)
   80 CONTINUE
      DO 200 J=1,2*NV+1
      DO 190 I=1,NA
      IF(FI(I).NE.J) GO TO 190
      VW=I
      V=0
      W=0
      DO 120 JJ=1,NV
      IF(IB(I,JJ).LT.0.AND.V.EQ.0) GO TO 110
      IF(IB(I,JJ).LT.0.AND.V.NE.0) GO TO 130
      GO TO 120
  110 V=JJ
  120 CONTINUE
  130 W=JJ
      DO 140 JJ=1,K2
      IF(VW.EQ.FROND(JJ)) GO TO 150
  140 CONTINUE
      IF(NUMB(V).LT.NUMB(W)) GO TO 160
      LL=V
      V=W
      W=LL
      GO TO 160
  150 IF(NUMB(V).GT.NUMB(W)) GO TO 160
      LL=V
      V=W
      W=LL
  160 DO 170 JJ=1,9
      IF(IAA(V,JJ).EQ.0) GO TO 180
  170 CONTINUE
C
C SALIDA DE ERROR SI LA VALENCIA ES MAYOR DE LO QUE PERMITE LA CAPACIDAD
C DE ALMACENAMIENTO
C
      IERR=NER(ISAL,2)
      STOP 0002
```

```
  180 IAA(V,JJ)=W
  190 CONTINUE
  200 CONTINUE
      RETURN
      END
C
C
      SUBROUTINE PAT(UU,K1,T,CAN,EEV)
C     ******************************
C
C SUBRUTINA QUE ESTABLECE UNA OPERACION DETERMINADA SEGUN DOS VERTICES
C ESTEN O NO SOBRE UN MISMO CAMINO
C
      INTEGER UU,T,EEV
      INTEGER CAN(EEV)
      DO 100 I=1,K1-2
      J1=0
      J2=0
      IF(CAN(I).GE.0) GO TO 100
      DO 50 J=I+1,K1-1
      IF(CAN(J).LT.0) GO TO 60
      IF(CAN(J).EQ.UU.AND.J2.EQ.0) J1=1
      IF(CAN(J).EQ.T.AND.J1.NE.0) J2=1
   50 CONTINUE
   60 IF(J1.NE.0.AND.J2.NE.0) GO TO 110
  100 CONTINUE
      GO TO 120
  110 UU=T
  120 RETURN
      END
C
C
      SUBROUTINE PA1(FLAG,JJ,SAV1,SAV2,HH,T,U,COMPO,GRADO,NV,EEE)
C     ************************************************************
C
C SUBRUTINA QUE REALIZA OPERACIONES AUXILIARES
C
      INTEGER FLAG,SAV1,SAV2,HH,T,U,EEE
      INTEGER GRADO(NV),COMPO(EEE)
      FLAG=0
      JJ=JJ+1
C
C  ** LLAMADAS A LA SUBRUTINA QUE ANADE DOS ELEMENTOS AL CONJUNTO COMPO
C     **
C
      CALL AD2(EEE,SAV1,SAV2,COMPO,HH)
      CALL AD2(EEE,T,U,COMPO,HH)
      CALL AD2(EEE,T,U,COMPO,HH)
      HH=HH+1
      COMPO(HH)=-JJ
      GRADO(T)=GRADO(T)-1
      GRADO(U)=GRADO(U)-1
      RETURN
      END
C
C
      SUBROUTINE PA2(GG,U,T,JJ,ESTA,GRADO,PAD,UU,K1,CAN,V,EEV,NV,EE)
C     ***************************************************************
C
C SUBRUTINA QUE REALIZA OPERACIONES AUXILIARES
C
```

```
      INTEGER GG,U,T,UU,V,EEV,EE
      INTEGER PAD(NV),GRADO(NV),CAM(EEV),ESTA(EE)
C
C  ** LLAMADA A LA SUBRUTINA QUE ANADE DOS ELEMENTOS AL CONJUNTO ESTA **
C
      CALL AD2(EE,U,T,ESTA,GG)
      GRADO(T)=GRADO(T)+1
      GRADO(U)=GRADO(U)+1
      PAD(T)=U
C
C  ** LLAMADA A LA SUBRUTINA QUE REALIZA UNA SERIE DE OPERACIONES SEGUN
C     DOS VERTICES ESTEN O NO SOBRE UN MISMO CAMINO GENERADO **
C
      CALL PAT(UU,K1,T,CAM,EEV)
      V=T
      RETURN
      END
C
C
      SUBROUTINE MUL(EE,EEE,NV,GG,V,W,HH,JJ,ESTA,GRADO,COMPO,IOP)
C     ***********************************************************
C
C SUBRUTINA PARA ELIMINAR LAS ARISTAS MULTIPLES DE LAS COMPONENTES TRI-
C CONEXAS
C
      INTEGER V,W,GG,HH,XX,YY,EE,EEE
      INTEGER ESTA(EE),COMPO(EEE),GRADO(NV)
      DO 100 I=GG,1,-2
      XX=ESTA(I-1)
      YY=ESTA(I)
      IF((XX.NE.V.OR.YY.NE.W).AND.(XX.NE.W.OR.YY.NE.V)) GO TO 100
      JJ=JJ+1
C
C  ** LLAMADAS A LA SUBRUTINA QUE ANADE DOS ELEMENTOS AL CONJUNTO COMPO
C     **
C
      CALL AD2(EEE,XX,YY,COMPO,HH)
      CALL AD2(EEE,V,W,COMPO,HH)
      CALL AD2(EEE,V,W,COMPO,HH)
      HH=HH+1
      COMPO(HH)=-JJ
      IF(IOP.EQ.2) GO TO 50
C
C  ** LLAMADA A LA SUBRUTINA QUE SUPRIME DOS ELEMENTOS DEL CONJUNTO ESTA
C     **
C
      CALL SU2(I,GG,ESTA,EE)
   50 GRADO(V)=GRADO(V)-1
      GRADO(W)=GRADO(W)-1
  100 CONTINUE
      RETURN
      END
C
C
      SUBROUTINE TUT(NA,NV,R,NUME,X,Y,IB,LN,ISAL)
C     *******************************************
C
C DIBUJA EL GRAFO O SUBGRAFOS POR EL ALGORITMO DE TUTTE
C
      EMA IB(150,50),X(NV),Y(NV)
```

```
      DIMENSION NUME(NV)
      CALL PLTLU(10)
      WRITE(ISAL,3001)
      CALL PLOT(R,R+0.40,-3)
C
C  NUMERACION DE LOS VERTICES
C
      IF(LN.LT.5) GO TO 5
      CALL PLOT(10.8,-44.8,-3)
      LN=0
      GO TO 7
    5 IF(LN.NE.0) CALL PLOT(0.,11.2,-3)
      LN=LN+1
    7 DO 10 J=1,NV
      IF(NUME(J).EQ.0) GO TO 10
      RJ=J
      CALL NUMB(X(J)-0.25,Y(J)-0.25,0.15,RJ,0.,-1)
   10 CONTINUE
C
C TRAZADO DE LAS ARISTAS
C
      DO 100 I=1,NA
      J1=0
      J2=0
      DO 50 J=1,NV
      IF(IB(I,J).EQ.0) GO TO 50
      IF(IB(I,J).NE.-1)GO TO 100
      IF(J1.NE.0) GO TO 60
      J1=J
   50 CONTINUE
   60 J2=J
      X1=X(J1)
      Y1=Y(J1)
      X2=X(J2)
      Y2=Y(J2)
      CALL PLOT(X1,Y1,3)
      CALL PLOT(X2,Y2,2)
  100 CONTINUE
      CALL PLOT(-R,-R-0.40,-3)
      RETURN
C
C FORMATOS
C
 3001 FORMAT(/1X"MENSAJE !!SR!! SE DIBUJA EN EL PLOTTER")
      END
C
C
      SUBROUTINE CIR(NV,NA,V1,NUMB,PILA,PAD,CIRCO,K,CODE,IA)
C     ****************************************************
C
C  SUBRUTINA QUE DETERMINA UN CIRCUITO PARA EL TRAZADO DEL GRAFO POR TU-
C  TTE
C
      EMA IA(100,50)
      INTEGER V1,V,W,CODE,S
      INTEGER NUMB(V1),PILA(V1),PAD(NV),CIRCO(NV)
C
C PUESTA A CERO
C
      DO 5 I=1,NV
      CIRCO(I)=0
```

```
      NUMB(I)=0
      PAD(I)=0
      PILA(I)=0
    5 CONTINUE
      NUMB(V1)=0
      PILA(V1)=0
C
C DETERMINACION DEL VERTICE INICIAL DEL CIRCUITO. OBTENCION DE ESTE UL-
C TIMO
C
      I1=IA(1,2)
      I2=IA(3,4)
      I3=IA(1,3)
      I4=IA(2,4)
      DO 20 I=1,NV
      DO 20 J=1,NV
      IF(IA(I,J).EQ.1) GO TO 30
   20 CONTINUE
   30 S=I
      LN=0
      V=S
      CODE=1
      KONT=1
      NUMB(V)=CODE
      PILA(KONT)=V
   40 KT=0
   45 DO 50 J=1,NV
      IF(IA(V,J).NE.1) GO TO 50
      IF(NUMB(J).NE.0.AND.KT.EQ.0) GO TO 50
      IA(V,J)=0
      IA(J,V)=0
      W=J
      GO TO 60
   50 CONTINUE
      IF(KT.NE.0) GO TO 55
      KT=KT+1
      GO TO 45
   55 IF(KONT.EQ.1) GO TO 80
      PILA(KONT)=0
      KONT=KONT-1
      V=PILA(KONT)
      GO TO 40
   60 IF(NUMB(W).NE.0) GO TO 70
      CODE=CODE+1
      KONT=KONT+1
      NUMB(W)=CODE
      PILA(KONT)=W
      PAD(W)=V
      V=W
      GO TO 40
   70 IF(W.NE.LN) GO TO 75
      IF(NUMB(W).LT.NUMB(JS)) GO TO 45
   75 LN=V
      JS=W
      GO TO 45
   80 IF(I1.NE.1.OR.I2.NE.1) GO TO 85
      IF(I3.EQ.1.OR.I4.EQ.1) GO TO 85
      CIRCO(1)=1
      CIRCO(2)=4
      CIRCO(3)=3
```

```
      CIRCO(4)=2
      K=4
      GO TO 110
   85 K=1
      CIRCO(K)=JS
      K=K+1
      CIRCO(K)=LN
      IS=LN
   90 K=K+1
      IF(PAD(IS).EQ.JS) GO TO 100
      CIRCO(K)=PAD(IS)
      IS=PAD(IS)
      GO TO 90
  100 K=K-1
C
C RENUMERACION DE LOS VERTICES
C
  110 KT=0
      DO 180 J=1,CODE
      DO 150 I=1,NV
      IF(NUMB(I).NE.J) GO TO 150
      DO 130 IT=1,K
      IF(I.NE.CIRCO(IT)) GO TO 130
      KT=KT+1
      GO TO 180
  130 CONTINUE
      NUMB(I)=NUMB(I)-KT
      GO TO 130
  150 CONTINUE
  180 CONTINUE
      DO 200 I=1,K
      NUMB(CIRCO(I))=CODE-K+I
  200 CONTINUE
      RETURN
      END
C
C
      SUBROUTINE OBC(NV,NA,K,CODE,NUMB,ICIRC,XC,YC,BX,BY,VAUX,A,IB,NN,
     &ISAL,ISAUX)
C     ****************************************************************
C
C     SUBRUTINA QUE OBTIENE LAS COORDENADAS DE LOS VERTICES SEGUN EL AL-
C     GORITMO DE TUTTE
C
      ENA IB(150,50),A(50,50),XC(NV),YC(NV),BX(NV),BY(NV)
      INTEGER CODE,VAUX(NV)
      DIMENSION NUMB(NV),ICIRC(K)
C
C PUESTA A CERO
C
      DO 10 I=1,NV
      BX(I)=0
      BY(I)=0
      XC(I)=0
      YC(I)=0
      DO 10 J=1,NV
   10 A(I,J)=0
C
C COORDENADAS DE LOS VERTICES DEL CIRCUITO
C
      ANG=6.2831853/K
```

```
      R=5.
      DO 50 II=1,K
      I=ICIRC(II)
      ANG1=-ANG*(II-1)+3.1415927/2.
      XC(NUMB(I))=R*COS(ANG1)
      YC(NUMB(I))=R*SIN(ANG1)
   50 CONTINUE
C
C ELABORACION DE LA SUBMATRIZ DE ADYACENCIA DE ACUERDO A LA NUEVA NUME-
C RACION DE LOS VERTICES
C
      DO 200 I=1,NA
      KT=0
      JJ=0
      DO 150 J=1,NV
      IF(IB(I,J).NE.-1) GO TO 150
      DO 100 L=1,K
      IF(J.EQ.ICIRC(L)) GO TO 110
  100 CONTINUE
      GO TO 120
  110 JJ=NUMB(J)
      GO TO 150
  120 IF(KT.NE.0) GO TO 160
      KT=KT+1
      J1=NUMB(J)
  150 CONTINUE
      IF(KT.EQ.0) GO TO 200
      BX(J1)=BX(J1)+XC(JJ)
      BY(J1)=BY(J1)+YC(JJ)
      A(J1,J1)=A(J1,J1)+1
      GO TO 200
  160 J2=NUMB(J)
      A(J1,J2)=-1.
      A(J2,J1)=-1.
      A(J1,J1)=A(J1,J1)+1
      A(J2,J2)=A(J2,J2)+1
  200 CONTINUE
C
C PIVOTAMIENTO DE LA MATRIZ Y DE LOS VECTORES
C
      IF(CODE-K.EQ.1) GO TO 360
      DO 350 J=1,CODE-K
      I=J+1
  310 RM=A(I,J)/A(J,J)
      A(I,J)=0
      L=J+1
  320 A(I,L)=A(I,L)-RM*A(J,L)
      IF(L.EQ.CODE-K) GO TO 330
      L=L+1
      GO TO 320
  330 BX(I)=BX(I)-RM*BX(J)
      BY(I)=BY(I)-RM*BY(J)
      IF(I.EQ.CODE-K) GO TO 340
      I=I+1
      GO TO 310
  340 IF(J.EQ.CODE-K-1) GO TO 360
  350 CONTINUE
C
C  ** LLAMADAS A LAS SUBRUTINAS QUE DETERMINAN LAS COORDENADAS DE LOS -
C     VERTICES SOLUCIONANDO EL SISTEMA POR EL METODO DE ELIMINACION DE
```

```
C      GAUSS **                                                          OBC-936
C                                                                        OBC-937
  360 CALL SOL(CODE-K,A,BX,XC)                                           OBC-938
      CALL SOL(CODE-K,A,BY,YC)                                           OBC-939
C                                                                        OBC-940
C  ** LLAMADA A LA SUBRUTINA QUE ASIGNA LAS COORDENADAS A LOS VERTICES   OBC-941
C      DE ACUERDO A LA PRIMITIVA NUMERACION **                           OBC-942
C                                                                        OBC-943
      CALL CA2(NV,CODE,NUMB,XC,BX)                                       OBC-944
      CALL CA2(NV,CODE,NUMB,YC,BY)                                       OBC-945
      NAA=0                                                              OBC-946
C                                                                        OBC-947
C  ** LLAMADA A LA SUBRUTINA QUE DIBUJA EL GRAFO O SUBGRAFO POR EL ALGO- OBC-948
C      RITMO DE TUTTE **                                                 OBC-949
C                                                                        OBC-950
      IF(ISSW(1).LT.0) 370,400                                           OBC-951
  370 WRITE(ISAUX,5001)                                                  OBC-952
      DO 380 I=1,NV                                                      OBC-953
      WRITE(ISAUX,5002) I,XC(I),I,YC(I)                                  OBC-954
  380 CONTINUE                                                           OBC-955
  400 IF(ISSW(2).LT.0) GO TO 450                                         OBC-956
      CALL TUT(NA,NV,R,NUMB,XC,YC,IB,NN,ISAL)                            OBC-957
  450 RETURN                                                             OBC-958
C                                                                        OBC-959
C FORMATOS                                                               OBC-960
C                                                                        OBC-961
 5001 FORMAT(/1X"COORDENADAS DE LOS VERTICES:"/)                         OBC-962
  502 FORMAT(1X"X("I2")="F6.2,3X"Y("I2")="F6.2)                          OBC-963
 5002 FORMAT(1X"X("I2")="F6.2,3X"Y("I2")="F6.2)                          OBC-963
      END                                                                OBC-964
C                                                                        SOL-965
C                                                                        SOL-966
      SUBROUTINE SOL(N,A,B,Z)                                            SOL-967
C     **********************                                             SOL-968
C                                                                        SOL-969
C SUBRUTINA QUE SOLUCIONA, POR EL METODO DE ELIMINACION DE GAUS, LAS     SOL-970
C ECUACIONES DE TUTTE EN X E Y, OBTENIENDO POR TANTO LAS COORDENADAS     SOL-971
C DE LOS VERTICES INTERIORES.                                            SOL-972
C                                                                        SOL-973
      ENA A(50,50),B(N),Z(N)                                             SOL-974
      DO 50 I=1,N                                                        SOL-975
   50 Z(I)=0.                                                            SOL-976
      Z(N)=B(N)/A(N,N)                                                   SOL-977
      IF(N.EQ.1) GO TO 110                                               SOL-978
      DO 100 I=N-1,1,-1                                                  SOL-979
      J=I+1                                                              SOL-980
      S=0                                                                SOL-981
   80 S=S+A(I,J)*Z(J)                                                    SOL-982
      IF(J.EQ.N) GO TO 90                                                SOL-983
      J=J+1                                                              SOL-984
      GO TO 80                                                           SOL-985
   90 Z(I)=(B(I)-S)/A(I,I)                                               SOL-986
      IF(I.EQ.1) RETURN                                                  SOL-987
  100 CONTINUE                                                           SOL-988
  110 RETURN                                                             SOL-989
      END                                                                SOL-990
C                                                                        NER-991
C                                                                        NER-992
      FUNCTION NER(ISAL,IERR)                                            NER-993
C     ***********************                                            NER-994
C                                                                        NER-995
```

```
C FUNCION QUE DETERMINA EL TIPO DE ERROR                              NER-996
C                                                                     NER-997
      WRITE(ISAL,3001) IERR                                           NER-998
      NER=IERR                                                        NER-999
      RETURN                                                          NER-000
C                                                                     NER-001
C FORMATOS                                                            NER-002
C                                                                     NER-003
 3001 FORMAT(//'ERROR ............... 'I3)                            NER-004
      END                                                             NER-005
$                                                                     NER-006
```

```
FTN4
$EMA(IJK,3)
C
C
C     **************
      PROGRAM PL1(5)
C     **************
C
C
C SEGMENTO 1......................... APLICACION DEL TEST DE CONECTIVIDAD
C
C
      COMMON IPAR(4),NAME(3)
      COMMON IENT,ISAL,IEAUX,ISAUX,NV,NA,KKTT
      COMMON NOM(40),JAUX(2200)
      COMMON /IJK/ IB(150,50),IA(100,50),X(300),Y(300),IAA(50,10)
C
C  ** LLAMADA A LA SUBRUTINA QUE OBTIENE LA MATRIZ DE ADYACENCIA **
C
      CALL ADY(NV,NA,IA,IB)
C
C CONTADOR PARA EL DIBUJO DEL GRAFO
C
      KKTT=0
C
C  ** LLAMADA A LA SUBRUTINA QUE COMPRUEBA LA CONECTIVIDAD DEL GRAFO **
C
  170 CALL CON(NV,NA,2*NA,2*NA+NV,NV+1,NA+NV,K1,JAUX(1),JAUX(NA+2*NV+1),
     &JAUX(3*NA+3*NV+1),JAUX(3*NA+4*NV+2),JAUX(3*NA+5*NV+3),JAUX(5*NA+5*
     &NV+3),IB)
C
C SI SOLO HAY UNA COMPONENTE CONEXA EL GRAFO ES CONEXO
C EN CASO CONTRARIO, HAY QUE INTRODUCIR NUEVAS ARISTAS
C
      IF(JAUX(1).EQ.NA) GO TO 200
      WRITE(ISAL,3001)
      K2=1
      KONT=1
  180 WRITE(ISAL,3002) KONT
      WRITE(ISAL,3003) (JAUX(J),J=K2+1,K2+JAUX(K2))
      IF(K2+JAUX(K2).GE.K1) GO TO 190
      KONT=KONT+1
      K2=K2+JAUX(K2)+1
      GO TO 180
  190 WRITE(ISAL,3004)
C
C  ** LLAMADA A LA SUBRUTINA QUE LEE LAS NUEVAS ARISTAS INTRODUCIDAS **
C
      CALL LE2(IEAUX,ISAL,ISAUX,2,NA,NV,IA,IB)
C
C UNA VEZ INTRODUCIDAS LAS ARISTAS SE VOLVERIA A COMPROBAR LA CONECTIVI-
C DAD.
C
      GO TO 170
  200 WRITE(ISAL,3005)
C
C  ** LLAMADA AL SEGMENTO PL2 QUE APLICA LOS TEST DE BICONECTIVIDAD,
C     PLANARIDAD Y TRICONECTIVIDAD Y OBTIENE EL GRAFO GEOMETRICO TRICO-
C     NEXO **
```

```
C
      NAME(2)=2H2
      NAME(3)=2H
      CALL EXEC(8,NAME)
      STOP
C
C FORMATOS
C
 3001 FORMAT(//1X"MENSAJE ::SR:: EL GRAFO NO ES CONEXO",//1X"LAS COMPONE
     &NTES CONEXAS SON:"/)
 3002 FORMAT(1X"COMPONENTE:"I3,/1X,14"-"/)
 3003 FORMAT(1X"ARISTAS:"10(I4"-"I2),20(/9X,10(I4"-"I2)))
 3004 FORMAT(//1X"MENSAJE ::SR:: HAS DE INTRODUCIR NUEVAS ARISTAS PARA L
     &IGAR TODAS LAS COMPONENTES")
 3005 FORMAT(//1X"MENSAJE ::SR:: EL GRAFO ES CONEXO"/)
      END
C
C
      SUBROUTINE ADY(NV,NA,IA,IB)
C     **************************
C
C SUBRUTINA QUE OBTIENE LA MATRIZ DE ADYACENCIA DEL GRAFO
C
      EMA IB(150,50),IA(100,50)
      DO 50 I=1,2*NV
      DO 50 J=1,NV
   50 IA(I,J)=0
      DO 150 I=1,NA
      KT=0
      DO 100 J=1,NV
      IF(IB(I,J).EQ.0) GO TO 100
      IF(KT.NE.0) GO TO 110
      KT=1
      JJ=J
  100 CONTINUE
  110 JJJ=J
      IA(JJ,JJJ)=1
      IA(JJJ,JJ)=1
  150 CONTINUE
      RETURN
      END
C
C
      SUBROUTINE CON(V,E,EE,EEV,V1,EV,CPTR,COM,SIG,NUMB,PILA,CABE,LISA,
     &IB)
C     ******************************************************************
C
C SUBRUTINA QUE OBTIENE LAS COMPONENTES CONEXAS DEL GRAFO
C
      EMA IB(150,50)
      INTEGER V,E,EE,EEV,V1,EV,CPTR,PUNTO,CODE,KONT,OLDTR,PT,OLDPT,V2
      INTEGER FRENE
      INTEGER LISA(EE),SIG(EEV),NUMB(V1),PILA(V1),CABE(EE),COM(EV)
C
C PUESTA A CERO
C
      DO 5 I=1,V1
      NUMB(I)=0
      PILA(I)=0
    5 CONTINUE
```

```
      DO 10 I=1,EEV
      SIG(I)=0
   10 CONTINUE
      DO 15 I=1,EE
      LISA(I)=0
      CABE(I)=0
   15 CONTINUE
      DO 18 I=1,EV
      COM(I)=0
   18 CONTINUE
C
C  ** LLAMADA A LA SUBRUTINA QUE ELABORA LA LISTA DE ARISTAS POR SUS -
C     VERTICES EXTREMOS **
C
      CALL TRA(V,E,EE,LISA,IB)
      FRENE=V
C
C  ** LLAMADA A LA SUBRUTINA QUE ELABORA EL CONJUNTO CABE **
C
      DO 20 I=1,E
      CALL PRO(V,EE,EEV,LISA(2*I-1),LISA(2*I),FRENE,SIG,CABE)
      CALL PRO(V,EE,EEV,LISA(2*I),LISA(2*I-1),FRENE,SIG,CABE)
   20 CONTINUE
      CPTR=0
      PUNTO=1
   35 IF(PUNTO.GT.V) GO TO 110
      CODE=1
      KONT=1
      NUMB(PUNTO)=CODE
      PILA(KONT)=PUNTO
      CPTR=CPTR+1
      OLDTR=CPTR
      PT=PUNTO
      OLDPT=0
   50 IF(SIG(PT).EQ.0) GO TO 60
      V2=CABE(SIG(PT)-V)
      SIG(PT)=SIG(SIG(PT))
      IF(NUMB(V2).GE.NUMB(PT).OR.V2.EQ.OLDPT) GO TO 50
C
C  ** LLAMADA A LA SUBRUTINA QUE ALMACENA EL CONJUNTO DE COMPONENTES CO-
C     NEXAS **
C
      CALL ADI(EV,PT,V2,COM,CPTR,V,E,IB)
      IF(NUMB(V2).NE.0) GO TO 50
      CODE=CODE+1
      KONT=KONT+1
      NUMB(V2)=CODE
      PILA(KONT)=V2
      OLDPT=PT
      PT=V2
      GO TO 50
   60 IF(KONT.EQ.1) GO TO 90
      PILA(KONT)=0
      KONT=KONT-1
      IF(KONT.EQ.1)70,80
   70 PT=1
      OLDPT=0
      GO TO 50
   80 PT=PILA(KONT)
      OLDPT=PILA(KONT-1)
      GO TO 50
```

```
   90 CON(OLDTR)=CPTR-OLDTR
  100 IF(NUMB(PUNTO).EQ.0) GO TO 35
      PUNTO=PUNTO+1
      GO TO 100
  110 RETURN
      END
$
```

```
FTN4
$EMA(IJK,3)
C
C
C     **************
      PROGRAM PL2(5)
C     **************
C
C
C SEGMENTO 2 ................................ APLICACION DE LOS TEST DE:
C                                              - PLANARIDAD
C                                              - BICONECTIVIDAD
C                                              - TRICONECTIVIDAD
C
C
      COMMON IPAR(4),NAME(3)
      COMMON IENT,ISAL,IEAUX,ISAUX,NV,NA,KKTT
      COMMON NOM(40),JAUX(2200)
      COMMON /IJK/ IB(150,50),IA(100,50),X(300),Y(300),IBA(50,10)
C
C  ** LLAMADA A LA SUBRUTINA QUE OBTIENE LAS COMPONENTES BICONEXAS **
C
  210 CALL BIC(NV,NA,2*NA,2*NA+NV,NV+1,NA+NV,K1,JONT,JAUX(1),JAUX(NA+NV+
     &1),JAUX(NA+2*NV+1),JAUX(3*NA+3*NV+1),JAUX(3*NA+4*NV+2),JAUX(3*NA+5
     &*NV+3),JAUX(5*NA+5*NV+3),JAUX(7*NA+5*NV+3),JAUX(9*NA+5*NV+3),IB)
      IF(JAUX(1).EQ.NA) GO TO 310
      WRITE(ISAL,3001)
      K2=1
      KONT=1
  260 WRITE(ISAL,3002) KONT
      WRITE(ISAL,3003) (JAUX(J),J=K2+1,K2+JAUX(K2))
      IF(K2+JAUX(K2).GE.K1) GO TO 270
      KONT=KONT+1
      K2=K2+JAUX(K2)+1
      GO TO 260
  270 WRITE(ISAL,3004)
      DO 200 I=1,JONT
      II=JAUX(NA+NV+I)
      K2=1
      KONT=1
   80 IF(K2.GE.K1) GO TO 200
      DO 100 J=K2+1,K2+JAUX(K2)
      JJ=JAUX(J)
      IF(IB(JJ,II).NE.0) GO TO 110
  100 CONTINUE
  105 KONT=KONT+1
      K2=K2+JAUX(K2)+1
      GO TO 80
  110 K3=K2+JAUX(K2)+1
      IONT=KONT+1
  120 IF(K3.GE.K1) GO TO 105
      DO 150 K=K3+1,K3+JAUX(K3)
      KK=JAUX(K)
      IF(IB(KK,II).NE.0) GO TO 160
  150 CONTINUE
      K3=K3+JAUX(K3)+1
      IONT=IONT+1
      GO TO 120
  160 WRITE(ISAL,3005) KONT,IONT,II
```

```
  200 CONTINUE
      GO TO 320
  310 WRITE(ISAL,3006)
C
C EN CADA COMPONENTE BICONEXA VAMOS A REALIZAR LAS SIGUIENTES OPERACIO-
C NES:
C - ESTUDIO DE LA PLANARIDAD
C - DESCOMPOSICION EN COMPONENTES TRICONEXAS
C - INTRODUCCION DE ARISTAS PARA HACER TRICONEXA LA COMPONENTE BICONEXA
C
  320 K2=1
      NCB=0
C
C PUESTA A CERO DE LA MATRIZ DE ADYACENCIA
C
  325 DO 330 I=1,2*NV
      DO 330 J=1,NV
      IA(I,J)=0
  330 CONTINUE
      DO 335 I=1,NA
      DO 335 J=1,NV
      IF(IB(I,J).NE.0) IB(I,J)=1
  335 CONTINUE
C
C HACEMOS NEGATIVOS LOS TERMINOS DE LA MATRIZ DE INCIDENCIA CORRESPON -
C DIENTES  A LA COMPONENTE BICONEXA. LOS DE LA MATRIZ DE ADYACENCIA LOS
C HACEMOS POSITIVOS
C
      IF(K2.GE.K1) GO TO 930
      NCB=NCB+1
      IF(JAUX(1).NE.NA) WRITE(ISAL,3007) NCB
      IF(JAUX(1).NE.NA) WRITE(ISAL,3003) (JAUX(J),J=K2+1,K2+JAUX(K2))
      DO 400 I=K2+1,K2+JAUX(K2)
      K=0
      J1=0
      J2=0
      DO 350 J=1,NV
      IF(IB(JAUX(I),J).EQ.0) GO TO 350
      IF(IB(JAUX(I),J).NE.0) IB(JAUX(I),J)=-1
      IF(K.EQ.0) 340,345
  340 J1=J
      K=K+1
      GO TO 350
  345 J2=J
      GO TO 360
  350 CONTINUE
      IF(J1.EQ.0.OR.J2.EQ.0) GO TO 400
  360 IA(J1,J2)=1
      IA(J2,J1)=1
  400 CONTINUE
C
C SALTO SI LA COMPONENTE BICONEXA ES UNA ARISTA O UN TRIANGULO
C
      IF(JAUX(K2).LE.3) GO TO 920
      II=0
      IT=0
      KKK=JAUX(K2)
C
C  ** LLAMADA A LA SUBRUTINA QUE APLICA LA FORMULA DE EULER A LA COMPO-
C     NENTE BICONEXA **
```

```
C
  410 CALL EUL(KKK,NV,II,IA)
C
C SALIDA DE ERROR SI NO ES PLANAR
C
      IF(II.EQ.0) GO TO 420
      IERR=NER(ISAL,3)
      STOP 0003
C
C  ** LLAMADA A LA SUBRUTINA QUE COMPRUEBA LA PLANARIDAD DE LA COMPONEN-
C     TE BICONEXA**
C
  420 CALL PLA(IS,NV,NA,2*NA,2*NA+NV,NV+1,N1,N2,II,IT,JAUX(NA+2*NV+1),
     &JAUX(3*NA+3*NV+1),JAUX(3*NA+4*NV+2),JAUX(3*NA+5*NV+3),JAUX(4*NA+5*
     &NV+3),JAUX(5*NA+5*NV+3),JAUX(6*NA+5*NV+3),JAUX(7*NA+5*NV+3),JAUX(9
     &*NA+7*NV+3),JAUX(9*NA+8*NV+3),JAUX(9*NA+9*NV+3),JAUX(9*NA+10*NV+3)
     &,JAUX(10*NA+10*NV+3),IAA,IA,IB,ISAUX)
C
C SALIDA DE ERROR SI NO ES PLANAR
C
      IF(II.EQ.0.AND.IT.EQ.0) GO TO 430
      II=NA+2*NV+II
      IT=NA+2*NV+IT
      WRITE(ISAL,3008) (JAUX(I),I=II,IT)
      WRITE(ISAL,3009)
C
C  ** LLAMADA A LA SUBRUTINA QUE SUSTITUYE UNAS RELACIONES POR OTRAS **
C
      CALL LE1(IEAUX,ISAL,ISAUX,1,IA,IB)
      GO TO 420
  430 IF(JAUX(1).EQ.NA) WRITE(ISAL,3010)
      IF(JAUX(1).NE.NA) WRITE(ISAL,3011)
C
C  ** LLAMADA A LA SUBRUTINA QUE DESCOMPONE UNA COMPONENTE BICONEXA EN
C     SUS COMPONENTES TRICONEXAS **
C
      CALL TRI(IS,NV,NA,2*NA,2*NA+NV,NV+1,6*NA,N1,N2,IONT,JAUX(NA+2*NV+1
     &),JAUX(3*NA+3*NV+1),JAUX(3*NA+4*NV+2),JAUX(3*NA+5*NV+3),JAUX(9*NA+
     &5*NV+3),JAUX(9*NA+6*NV+3),JAUX(9*NA+7*NV+3),JAUX(9*NA+8*NV+3),
     &JAUX(9*NA+9*NV+3),JAUX(9*NA+10*NV+3),JAUX(10*NA+10*NV+3),JAUX(11*N
     &A+10*NV+3),JAUX(13*NA+10*NV+3),JAUX(13*NA+11*NV+3),JAUX(13*NA+12*N
     &V+3),IAA,IB)
C
C SALTO SI EL GRAFO ES TRICONEXO
C
      IF(IONT-1.EQ.2*NA) GO TO 938
      KT1=3*NA+5*NV+2
      KF=3*NA+5*NV+2+IONT
      KT2=0
      KT3=0
      KT5=0
      NAA=NA
      DO 435 JJ=KT1+1,KF
      IF(JAUX(JJ).GE.0) GO TO 435
      KT5=KT5+1
  435 CONTINUE
C
C SI KT5=1 LA COMPONENTE BICONEXA ES TRICONEXA
C
      IF(KT5.EQ.1) GO TO 910
      IF(JAUX(1).EQ.NA) WRITE(ISAL,3027)
```

```
      IF(JAUX(1).NE.NA) WRITE(ISAL,3012)
      DO 460 JJ=KT1+1,KF
      IF(JAUX(JJ).GE.0) GO TO 460
      KT2=JJ
C
C SI SON MENOS DE CUATRO EL NUMERO DE ARISTAS NO ES UNA COMPONENTE TRI-
C CONEXA EN UN PRINCIPIO
C
      IF(KT2-KT1.LT.7) GO TO 454
      IF(KT2-KT1.EQ.7) GO TO 455
      KT3=KT3+1
      WRITE(ISAL,3013) KT3,(JAUX(J),J=KT1+1,KT2-1)
      DO 450 K=KT1+1,KT2-1,2
      J1=JAUX(K)
      J2=JAUX(K+1)
      IF(IA(J1,J2).GT.0) GO TO 450
      WRITE(ISAL,3014) KT3,J1,J2
C
C SE INTRODUCE LA ARISTA VIRTUAL QUE TIENE COMO EXTREMO J1,J2
C
      IA(J1,J2)=1
      IA(J2,J1)=1
      NA=NA+1
      DO 440 L=1,NV
  440 IB(NA,L)=0
      IB(NA,J1)=-1
      IB(NA,J2)=-1
  450 CONTINUE
      JAUX(KT2)=-KT3
      GO TO 456
  454 JAUX(KT2)=JAUX(KT2)-200
      GO TO 456
  455 JAUX(KT2)=JAUX(KT2)-100
  456 KT1=KT2
      IF(KT1.GE.KF) GO TO 465
  460 CONTINUE
C
C SI HAY UNA COMPONENTE ESCINDIDA QUE ES UN TRIANGULO DONDE TODAS LAS
C ARISTAS SON REALES Y NO INCLUIDAS EN NINGUNA COMPONENTE TRICONEXA LA
C SENALAMOS TAMBIEN
C
  465 KT1=3*NAA+5*NV+2
      DO 490 K=KT1+1,KF
      IF(JAUX(K).GT.-100.OR.JAUX(K).LT.-200) GO TO 490
      IF(JAUX(K-1).EQ.JAUX(K-5).AND.JAUX(K-2).EQ.JAUX(K-6)) GO TO 490
      IF(JAUX(K-1).EQ.JAUX(K-6).AND.JAUX(K-2).EQ.JAUX(K-5)) GO TO 490
      DO 480 JJ=K-6,K-1,2
      IX=JAUX(JJ)
      IY=JAUX(JJ+1)
      IF(IA(IX,IY).LE.0) GO TO 490
      KK1=KT1+1
      DO 475 II=KT1+1,KF
      IF(JAUX(II).GT.0) GO TO 475
      KK2=II-1
      IF(JAUX(II).LT.-100) GO TO 474
      DO 470 KK=KK1,KK2,2
      IF(IX.EQ.JAUX(KK).AND.IY.EQ.JAUX(KK+1)) GO TO 490
      IF(IY.EQ.JAUX(KK).AND.IX.EQ.JAUX(KK+1)) GO TO 490
  470 CONTINUE
  474 KK1=KK2+2
```

```
      IF(KK1.GE.KF) GO TO 489
  475 CONTINUE
  480 CONTINUE
      KT3=KT3+1
      JAUX(K)=-KT3
      WRITE(ISAL,3013) KT3,(JAUX(J),J=K-6,K-1)
  490 CONTINUE
      IF(NAA.NE.NA) WRITE(ISAL,3015)
C
C ARISTAS QUE LIGAN UNAS COMPONENTES CON OTRAS
C ARISTAS QUE NO LIGAN A NINGUNA COMPONENTE
C
      L1=9*NAA+10*NV+2
      DO 501 LL=L1+1,L1+2*NAA
      JAUX(LL)=0
  501 CONTINUE
      KT5=0
      KT1=3*NAA+5*NV+2
      KT2=0
      KT6=0
      KT9=0
      L2=0
      DO 517 JJ=KT1+1,KF
      IF(JAUX(JJ).GT.-100.OR.JAUX(JJ).LT.-200) GO TO 517
      DO 515 II=JJ-6,JJ-1,2
      IX=JAUX(II)
      IY=JAUX(II+1)
      IF(IA(IX,IY).LE.0) GO TO 515
      IF(L2.EQ.0) GO TO 505
      DO 504 LL=1,L2,2
      IF(IX.EQ.JAUX(LL+L1).AND.IY.EQ.JAUX(L1+LL+1)) GO TO 515
      IF(IY.EQ.JAUX(LL+L1).AND.IX.EQ.JAUX(L1+LL+1)) GO TO 515
  504 CONTINUE
  505 KT3=0
      KT4=0
      KT7=3*NAA+5*NV+2
      KT8=0
      DO 510 KK=KT7+1,KF
      IF(JAUX(KK).GT.0) GO TO 510
      KT8=KK
      IF(JAUX(KK).LT.-100) GO TO 509
      DO 506 MM=KT7+1,KT8-1
      IF(IX.EQ.JAUX(MM)) KT3=-JAUX(KT8)
      IF(IY.EQ.JAUX(MM)) KT4=-JAUX(KT8)
  506 CONTINUE
      IF(KT3.EQ.0.OR.KT4.EQ.0) GO TO 509
      IF(KT3.NE.KT4) GO TO 514
      GO TO 515
  509 KT7=KT8
      IF(KT7.GE.KF) GO TO 511
  510 CONTINUE
  511 IF(L2.EQ.0) GO TO 513
      DO 512 LL=1,L2,2
      IF(IX.EQ.JAUX(LL+L1).AND.IY.EQ.JAUX(LL+L1+1)) GO TO 515
      IF(IY.EQ.JAUX(LL+L1).AND.IX.EQ.JAUX(LL+L1+1)) GO TO 515
  512 CONTINUE
  513 IF(KT9.EQ.0) WRITE(ISAL,3016)
      KT9=1
      WRITE(ISAL,3017) IX,IY
      JAUX(L1+L2+1)=IX
      JAUX(L1+L2+2)=IY
```

```
      L2=L2+2
      GO TO 515
  514 JAUX(L1+L2+1)=IX
      JAUX(L1+L2+2)=IY
      L2=L2+2
      IF(KT9.EQ.1) WRITE(ISAL,3016)
      WRITE(ISAL,3018) IX,IY,KT3,KT4
      KT9=1
  515 CONTINUE
  517 CONTINUE
C
C DIBUJO DE CADA COMPONENTE TRICONEXA DE LA BICONEXA
C
      NCT=0
      KT1=3*NAA+5*NV+2
      DO 800 JJ=KT1+1,KF
      IF(JAUX(JJ).GT.0) GO TO 800
      KT2=JJ
      IF(JAUX(JJ).LT.-100) GO TO 760
      DO 650 I=1,NA
      J1=0
      J2=0
      DO 550 J=1,NV
      IF(IB(I,J).GT.0) GO TO 650
      IF(IB(I,J).EQ.0) GO TO 550
      IF(J1.EQ.0) 519,520
  519 J1=J
      GO TO 550
  520 J2=J
      GO TO 560
  550 CONTINUE
  560 KT3=0
      DO 600 K=KT1+1,KT2-1,2
      IF(JAUX(K).EQ.J1.AND.JAUX(K+1).EQ.J2) KT3=1
      IF(JAUX(K).EQ.J2.AND.JAUX(K+1).EQ.J1) KT3=1
      IF(KT3.EQ.1) GO TO 640
  600 CONTINUE
      IB(I,J1)=-2
      IB(I,J2)=-2
      IA(J1,J2)=0
      IA(J2,J1)=0
      GO TO 650
  640 IA(J1,J2)=1
      IA(J2,J1)=1
  650 CONTINUE
      NCT=NCT+1
C
      IF(KT2-KT1.NE.7) GO TO 690
      WRITE(ISAL,3019) NCT,(JAUX(IJ),IJ=KT1+1,KT2-1)
      WRITE(ISAL,3020)
      GO TO 698
  690 WRITE(ISAL,3021) NCT,(JAUX(IJ),IJ=KT1+1,KT2-1)
C
C
C  ** LLAMADA A LA SUBRUTINA QUE OBTIENE UN CIRCUITO EN LA COMPONENTE
C      TRICONEXA **
C
      CALL CIR(NV,NA,NV+1,JAUX(3*NA+3*NV+1),JAUX(3*NA+4*NV+2),JAUX(9*NA+
     &6*NV+3),JAUX(9*NA+7*NV+3),KK,LONT,IA)
C
```

```
C  ** LLAMADA A LA SUBRUTINA QUE OBTIENE LAS COORDENADAS DE LOS VERTICES
C     APLICANDO EL ALGORITMO DE TUTTE **
C
      CALL OBC(NV,NA,KK,LONT,JAUX(3*NA+3*NV+1),JAUX(9*NA+7*NV+3),X(1),
     &Y(1),X(NV+1),Y(NV+1),JAUX(9*NA+5*NV+3),IA,IB,KKTT,ISAL,ISAUX)
  698 DO 699 I=1,2*NV
      DO 699 J=1,NV
  699 IA(I,J)=0
      DO 750 I=1,NA
      J1=0
      J2=0
      DO 700 J=1,NV
      IF(IB(I,J).GE.-1) GO TO 700
      IB(I,J)=-1
      IF(J1.NE.0) GO TO 710
      J1=J
  700 CONTINUE
      GO TO 750
  710 J2=J
      IA(J1,J2)=1
      IA(J2,J1)=1
  750 CONTINUE
  760 KT1=KT2
      IF(KT1.GE.KF) GO TO 810
  800 CONTINUE
  810 WRITE(ISAL,3022)
      DO 815 I=1,NA
      J1=0
      J2=0
      DO 812 J=1,NV
      IF(IB(I,J).EQ.0) GO TO 812
      IF(IB(I,J).GT.0) GO TO 815
      IF(J1.NE.0) GO TO 813
      J1=J
  812 CONTINUE
  813 J2=J
      IA(J1,J2)=1
      IA(J2,J1)=1
  815 CONTINUE
C
C ** LLAMADA A LA SUBRUTINA QUE LEE LAS NUEVAS RELACIONES INTRODUCIDAS
C    QUE LIGAN A LAS COMPONENTES TRICONEXAS **
C
      CALL LE2(IEAUX,ISAL,ISAUX,1,NA,NV,IA,IB)
      WRITE(ISAL,3023)
C
C  ** LLAMADA A LA SUBRUTINA QUE SUSTITUYE ALGUNA DE LAS RELACIONES IN-
C     TRODUCIDAS POR OTRAS, POR EJEMPLO, LAS ARISTAS VIRTUALES HECHAS
C     REALES **
C
      CALL LE1(IEAUX,ISAL,ISAUX,1,IA,IB)
      KKK=0
      DO 900 I=1,NA
      KK1=0
      DO 850 J=1,NV
      IF(IB(I,J).GT.0) GO TO 900
      IF(IB(I,J).EQ.0) GO TO 850
      IF(KK1.EQ.0) KKK=KKK+1
      KK1=1
      IF(IB(I,J).LT.0) IB(I,J)=-1
  850 CONTINUE
```

```
  900 CONTINUE
      II=0
      GO TO 410
C
C DIBUJO DE LA COMPONENTE BICONEXA QUE ES ADEMAS TRICONEXA
C
C  ** LLAMADA A LA SUBRUTINA QUE OBTIENE UN CIRCUITO EN LA COMPONENTE
C      TRICONEXA **
C
  910 WRITE(ISAL,3024)
      CALL CIR(NV,NA,NV+1,JAUX(3*NA+3*NV+1),JAUX(3*NA+4*NV+2),JAUX(9*NA+
     &6*NV+3),JAUX(9*NA+7*NV+3),KK,LONT,IA)
C
C  ** LLAMADA A LA SUBRUTINA QUE OBTIENE LAS COORDENADAS DE LOS VERTICES
C      APLICANDO EL ALGORITMO DE TUTTE **
C
      CALL OBC(NV,NA,KK,LONT,JAUX(3*NA+3*NV+1),JAUX(9*NA+7*NV+3),X(1),
     &Y(1),X(NV+1),Y(NV+1),JAUX(9*NA+5*NV+3),IA,IB,KKTT,ISAL,ISAUX)
      K2=K2+JAUX(K2)+1
      GO TO 325
  920 IF(JAUX(K2).EQ.1) GO TO 925
      L1=JAUX(K2+1)
      L2=JAUX(K2+2)
      L3=JAUX(K2+3)
      WRITE(ISAL,3020)
      DO 923 J=1,NV
      IF(IB(L1,J).NE.0) IB(L1,J)=1
      IF(IB(L2,J).NE.0) IB(L2,J)=1
      IF(IB(L3,J).NE.0) IB(L3,J)=1
  923 CONTINUE
      GO TO 929
  925 L1=JAUX(K2+1)
      WRITE(ISAL,3020)
      DO 927 J=1,NV
      IF(IB(L1,J).NE.0) IB(L1,J)=1
  927 CONTINUE
  929 K2=K2+JAUX(K2)+1
      GO TO 325
  930 DO 935 I=1,NA
      DO 935 J=1,NV
      IF(IB(I,J).NE.0) IB(I,J)=1
  935 CONTINUE
      WRITE(ISAL,3025)
C
C  ** LLAMADA A LA SUBRUTINA QUE INTRODUCE NUEVAS ARISTAS PARA HACER
C      TRICONEXO AL GRAFO **
C
      CALL LE2(IEAUX,ISAL,ISAUX,2,NA,NV,IA,IB)
      WRITE(ISAL,3023)
C
C  ** LLAMADA A LA SUBRUTINA QUE SUSTITUYE ALGUNA DE LAS RELACIONES IN-
C      TRODUCIDAS POR OTRAS **
C
      CALL LE1(IEAUX,ISAL,ISAUX,2,IA,IB)
      GO TO 210
C
C DIBUJO DEL GRAFO TRICONEXO
C
C
C  ** LLAMADA A LA SUBRUTINA QUE OBTIENE UN CIRCUITO EN EL GRAFO TRICO-
```

```
C     NEXO **
C
  938 WRITE(ISAL,3026)
  940 CALL CIR(NV,NA,NV+1,JAUX(3*NA+3*NV+1),JAUX(3*NA+4*NV+2),JAUX(9*NA+
     &6*NV+3),JAUX(9*NA+7*NV+3),KK,LONT,IA)
C
C  ** LLAMADA A LA SUBRUTINA QUE OBTIENE LAS COORDENADAS DE LOS VERTICES
C     APLICANDO EL ALGORITMO DE TUTTE **
C
      CALL OBC(NV,NA,KK,LONT,JAUX(3*NA+3*NV+1),JAUX(9*NA+7*NV+3),X(1),
     &Y(1),X(NV+1),Y(NV+1),JAUX(9*NA+5*NV+3),IA,IB,KKTT,ISAL,ISAUX)
      DO 941 I=1,2*NV
      DO 941 J=1,NV
  941 IA(I,J)=0
      DO 944 I=1,NA
      J1=0
      J2=0
      DO 942 J=1,NV
      IF(IB(I,J).EQ.0) GO TO 942
      IF(J1.NE.0) GO TO 943
      J1=J
  942 CONTINUE
  943 J2=J
      IA(J1,J2)=1
      IA(J2,J1)=1
  944 CONTINUE
C
C  ** LLAMADA AL SEGMENTO PL3 QUE OBTIENE EL DUAL DEL GRAFO INICIAL **
C
  990 NAME(2)=2H3
      NAME(3)=2H
      CALL EXEC(8,NAME)
      STOP
C
C FORMATOS
C
 3001 FORMAT(/1X"MENSAJE ::SR:: EL GRAFO NO ES BICONEXO",//1X"LAS COMPON
     &ENTES BICONEXAS SON:")
 3002 FORMAT(/1X"COMPONENTE:"I3,/1X,14"-")
 3003 FORMAT(1X"ARISTAS:"17I4,15(/9X,17I4))
 3004 FORMAT(//1X"LOS VERTICES DE SEPARACION DE ESAS COMPONENTES SON:"/)
 3005 FORMAT(1X"COMPONENTES:"I2"-"I2" .... VERT. DE SEPAR:"I3)
 3006 FORMAT(/1X"MENSAJE ::SR:: EL GRAFO ES BICONEXO")
 3007 FORMAT(//37X"COMPONENTE BICONEXA:"I3,/1X,79"-"/)
 3008 FORMAT(//1X"MENSAJE ::SR:: EL GRAFO NO ES PLANAR"//1X"EL CAMINO QU
     &E NO PUEDE SER REPRESENTADO EN EL PLANO (DADO POR LOS VERTICES) ES
     &:"/,(26I3))
 3009 FORMAT(//1X"MENSAJE ::SR:: HA DE SUSTITUIRSE UNA DE LAS ARISTAS DE
     &L CAMINO POR OTRA."//16X"ESTA NUEVA ARISTA TIENE QUE INCIDIR EN DO
     &S VERTICES DE LA MISMA"/16X"COMPONENTE BICONEXA.")
 3010 FORMAT(//1X"MENSAJE ::SR:: EL GRAFO ES PLANAR")
 3011 FORMAT(//1X"MENSAJE ::SR:: LA COMPONENTE BICONEXA ES PLANAR")
 3012 FORMAT(//1X"MENSAJE ::SR:: LA COMPONENTE BICONEXA NO ES TRICONEXA"
     &//1X"LAS COMPONENTES TRICONEXAS Y TRIANGULOS (DONDE LAS ARISTAS VI
     &ENEN EXPRESADAS"/1X"POR SUS VERTICES EXTREMOS) SON:"/)
 3013 FORMAT(/1X"COMPONENTE:"I2,/1X"ARISTAS:"10(I4"-"I2),20(/9X,10(I4"-"
     &I2)))
 3014 FORMAT(//1X"MENSAJE ::SR:: ARISTA VIRTUAL EN LA COMPONENTE"I2":"I4
     &"-"I2)
 3015 FORMAT(//1X"MENSAJE ::SR:: LAS ARISTAS VIRTUALES CITADAS HAN PASAD
     &O A SER REALES.",/16X"POSTERIORMENTE,SI SE DESEA,PUEDEN SER CAMBIA
```

```
     &DAS POR OTRAS.")
3016 FORMAT(//1X"MENSAJE ::SR:: ADEMAS DE LAS ARISTAS PERTENECIENTES A
     &ALGUNA COMPONENTE",/16X"HAY OTRAS QUE NO PERTENECEN A NINGUNA DE E
     &LLAS.",//1X"LAS ARISTAS QUE RELACIONAN UNAS COMPONENTES CON OTRAS
     &Y LAS QUE NO ENLAZAN A"/1X"DOS COMPONENTES CUALESQUIERA SON:"/)
3017 FORMAT(1X,"ARISTA:"I4"-"I2,2X"(NO LIGA A DOS COMPONENTES)")
3018 FORMAT(1X,"ARISTA:"I4"-"I2,2X"COMPONENTE:"I2" Y"I2)
3019 FORMAT(//1X,25"-"/1X"COMPONENTE:"I2" (TRIANGULO)"/1X,25"-"//1X"ARI
     &STAS:"10(I4"-"I2),20(/9X,10(I4"-"I2)))
3020 FORMAT(/1X"MENSAJE ::SR:: NO SE DIBUJA POR SER ELEMENTAL")
3021 FORMAT(//1X,25"-"/1X"COMPONENTE:"I2" (TRICONEXA)"/1X,25"-"//1X"ARI
     &STAS:"10(I4"-"I2),20(/9X,10(I4"-"I2)))
3022 FORMAT(//1X"MENSAJE ::SR:: SE HAN DE INTRODUCIR NUEVAS ARISTAS PAR
     &A CONVERTIR EN TRICONEXA"/16X"LA COMPONENTE BICONEXA."//16X"LA ADI
     &CION SE HARA A LA VISTA DEL SUBGRAFO TRAZADO.",/16X"SE PUEDE REALI
     &ZAR ESCALONADAMENTE,INTRODUCIENDO SOLO PARTE DE"/16X"ELLAS, ANADIE
     &NDO EL RESTO A PARTIR DE UN NUEVO TRAZADO.")
3023 FORMAT(//1X"MENSAJE ::SR:: SI SE DESEA SE PUEDE SUSTITUIR ARISTAS"
     &)
3024 FORMAT(//1X"MENSAJE ::SR:: LA COMPONENTE BICONEXA ES ADEMAS TRICON
     &EXA")
3025 FORMAT(//1X"MENSAJE ::SR:: SE HAN DE ANADIR NUEVAS ARISTAS PARA QU
     &E EL GRAFO SEA TRICONEXO"//16X"ESTAS ARISTAS ENLAZARAN A LAS DISTI
     &NTAS COMPONENTES BICONEXAS"/16X"ENTRE SI.")
3026 FORMAT(//1X"MENSAJE ::SR:: EL GRAFO ES TRICONEXO")
3027 FORMAT(//1X"MENSAJE ::SR:: EL GRAFO NO ES TRICONEXO"//1X"LAS COMPO
     &NENTES TRICONEXAS Y TRIANGULOS (DONDE LAS ARISTAS VIENEN EXPRESADA
     &S POR SUS VERTICES EXTREMOS) SON:"/)
      END
C
C
      SUBROUTINE BIC(V,E,EE,EEV,V1,EV,BPTR,JONT,BICON,ARTIC,SIG,NUMB,
     &PILA,CABE,LISA,PIAR,PTBJ,IB)
C     ***************************************************************
C
C SUBRUTINA QUE OBTIENE LAS COMPONENTES BICONEXAS DE UN GRAFO
C
      ENA IB(150,50)
      INTEGER V,E,EPTR,BPTR,PUNTO,OLDPT,CODE,KONT,V2,JONT,OLDTR
      INTEGER EE,EEV,V1,EV,FRENE
      INTEGER LISA(EE),SIG(EEV),NUMB(V1),PILA(V1),PTBJ(V),CABE(EE)
      INTEGER PIAR(EE),ARTIC(V),BICON(EV)
C
C PUESTA A CERO C PUESTA A CERO
C
      DO 5 I=1,V
      NUMB(I)=0
      PILA(I)=0
      PTBJ(I)=0
      ARTIC(I)=0
    5 CONTINUE
      PILA(V1)=0
      NUMB(V1)=0
      DO 10 I=1,EEV
      SIG(I)=0
   10 CONTINUE
      DO 15 I=1,EE
      CABE(I)=0
      PIAR(I)=0
      LISA(I)=0
```

```
   15 CONTINUE
      DO 18 I=1,EV
      BICON(I)=0
   18 CONTINUE
C
C ** LLAMADA A LA SUBRUTINA QUE ELABORA LA LISTA DE ARISTAS POR SUS VER-
C    TICES EXTREMOS **
C
      CALL TRA(V,E,EE,LISA,IB)
      FRENE=V
C
C  ** LLAMADA A LA SUBRUTINA QUE ELABORA EL CONJUNTO CABE **
C
      DO 20 I=1,E
      CALL PRO(V,EE,EEV,LISA(2*I-1),LISA(2*I),FRENE,SIG,CABE)
      CALL PRO(V,EE,EEV,LISA(2*I),LISA(2*I-1),FRENE,SIG,CABE)
   20 CONTINUE
      JONT=0
      EPTR=0
      BPTR=0
      PUNTO=1
      OLDPT=0
      CODE=1
      KONT=1
      NUMB(PUNTO)=CODE
      PILA(KONT)=PUNTO
      PTBJ(PUNTO)=1
   40 IF(SIG(PUNTO).EQ.0) GO TO 70
      V2=CABE(SIG(PUNTO)-V)
      SIG(PUNTO)=SIG(SIG(PUNTO))
      IF(NUMB(V2).GE.NUMB(PUNTO).OR.V2.EQ.OLDPT) GO TO 40
C
C  ** LLAMADA A LA SUBRUTINA QUE ALMACENA ARISTAS POR SUS VERTICES EX-
C     TREMOS **
C
      CALL AD2(EE,PUNTO,V2,PIAR,EPTR)
      IF(NUMB(V2).EQ.0) 50,60
   50 CODE=CODE+1
      KONT=KONT+1
      NUMB(V2)=CODE
      PILA(KONT)=V2
      OLDPT=PUNTO
      PUNTO=V2
      PTBJ(PUNTO)=NUMB(OLDPT)
      GO TO 40
   60 IF(NUMB(V2).LT.PTBJ(PUNTO)) PTBJ(PUNTO)=NUMB(V2)
      GO TO 40
   70 IF(KONT.EQ.1) GO TO 150
      IF(PTBJ(PUNTO).NE.NUMB(OLDPT)) 80,110
   80 IF(PTBJ(PUNTO).LT.PTBJ(OLDPT)) PTBJ(OLDPT)=PTBJ(PUNTO)
      PILA(KONT)=0
      KONT=KONT-1
      PUNTO=PILA(KONT)
      OLDPT=PILA(KONT-1)
      GO TO 40
  110 DO 115 I=1,JONT
      IF(OLDPT.EQ.ARTIC(I)) GO TO 116
  115 CONTINUE
      JONT=JONT+1
C
C SE OBTIENEN LOS PUNTOS DE ARTICULACION
```

```
C
      ARTIC(JONT)=OLDPT
  116 BPTR=BPTR+1
      OLDTR=BPTR
  120 IF(NUMB(PIAR(EPTR-1)).GT.NUMB(OLDPT)) 130,140
C
C  ** LLAMADA A LA SUBRUTINA QUE ALMACENA EL CONJUNTO DE COMPONENTES BI-
C     CONEXAS **
C
  130 CALL AD1(EV,PIAR(EPTR-1),PIAR(EPTR),BICON,BPTR,V,E,IB)
      EPTR=EPTR-2
      GO TO 120
  140 CALL AD1(EV,OLDPT,PUNTO,BICON,BPTR,V,E,IB)
      EPTR=EPTR-2
      BICON(OLDTR)=BPTR-OLDTR
      GO TO 80
  150 RETURN
      END
C
C
      SUBROUTINE EUL(KA,NV,II,IA)
C     ***************************
C
C SUBRUTINA QUE APLICA LA FORMULA DE EULER COMO PRIMER TEST DE PLANARI-
C DAD
C
      EMA IA(100,50)
      KV=0
      DO 100 I=1,NV
      DO 50 J=1,NV
      IF(IA(I,J).LE.0) GO TO 50
      KV=KV+1
      GO TO 100
   50 CONTINUE
  100 CONTINUE
      IF(KA.GT.3*KV-6) II=1
      RETURN
      END
C
C
      SUBROUTINE PLA(S,NV,NA,EE,EEV,V1,K1,N2,JF,KF,CAN,NUMB,PILA,FI,
     &BUC,FROND,ARAR,ALMAC,ND,PTBJ1,PTBJ2,CADE,F,IAA,IA,IB,ISAUX)
C     ****************************************************************
C
C SUBRUTINA QUE CHEQUEA LA PLANARIDAD PARA UN GRAFO BICONEXO
C
      EMA IB(150,50),IA(100,50),IAA(50,10)
      INTEGER CODE,V,S,U,W,VW,EE,EEV,V1
      INTEGER ARAR(NA),FROND(NA),NUMB(V1),FI(NA)
      INTEGER PILA(V1),PTBJ1(NV),PTBJ2(NV),BUC(NA)
      INTEGER CAN(EEV),ND(NV),ALMAC(EE),CADE(NA),F(NA)
C
C PUESTA A CERO
C
      S=0
      DO 5 I=1,EEV
      CAN(I)=0
    5 CONTINUE
      DO 10 I=1,EE
      ALMAC(I)=0
```

```
   10 CONTINUE
      DO 15 I=1,50
      DO 15 J=1,10
      IAA(I,J)=0
   15 CONTINUE
      DO 20 I=1,NA
      ARAR(I)=0
      FROND(I)=0
      FI(I)=0
      BUC(I)=0
   20 CONTINUE
      DO 30 I=1,NV
      PTOJ1(I)=0
      PTOJ2(I)=0
      ND(I)=1
      NUMB(I)=0
      PILA(I)=0
      CADE(I)=0
      F(I)=0
   30 CONTINUE
      PILA(V1)=0
      NUMB(V1)=0
C
C DETERMINACION DEL VERTICE INICIAL
C
      DO 32 I=1,NV
      DO 31 J=1,NV
      IF(IA(I,J).LE.0) GO TO 31
      S=I
      GO TO 33
   31 CONTINUE
   32 CONTINUE
   33 CODE=1
      KONT=1
      V=S
      U=0
      N1=0
      N2=0
      JF=0
      KF=0
      NUMB(V)=CODE
      PILA(KONT)=V
      PTOJ1(V)=NUMB(V)
      PTOJ2(V)=NUMB(V)
C
C D.F.S.
C
   35 DO 100 J=1,NV
      IF(IA(V,J).LE.0) GO TO 100
      W=J
      IA(V,J)=-IA(V,J)
      IA(J,V)=-IA(J,V)
      DO 40 I=1,NA
      IF(IB(I,V).LT.0.AND.IB(I,W).LT.0) GO TO 50
   40 CONTINUE
   50 VW=I
      IF(NUMB(W).EQ.0) 60,70
   60 N1=N1+1
      ARAR(N1)=VW
      CODE=CODE+1
      KONT=KONT+1
```

```
      NUMB(W)=CODE
      PILA(KONT)=W
      PTBJ1(W)=NUMB(W)
      PTBJ2(W)=NUMB(W)
      U=V
      V=W
      GO TO 35
   70 N2=N2+1
      FROND(N2)=VW
      IF(NUMB(W).LT.PTBJ1(V))GO TO 80
      IF(NUMB(W).GT.PTBJ1(V)) PTBJ2(V)=MIN0(PTBJ2(V),NUMB(W))
      GO TO 100
   80 PTBJ2(V)=PTBJ1(V)
      PTBJ1(V)=NUMB(W)
  100 CONTINUE
      IF(KONT.EQ.1) GO TO 140
      PILA(KONT)=0
      KONT=KONT-1
      ND(U)=ND(U)+ND(V)
      IF(PTBJ1(V).LT.PTBJ1(U)) GO TO 110
      IF(PTBJ1(V).EQ.PTBJ1(U)) PTBJ2(U)=MIN0(PTBJ2(U),PTBJ2(V))
      IF(PTBJ1(V).GT.PTBJ1(U)) PTBJ2(U)=MIN0(PTBJ2(U),PTBJ1(V))
      GO TO 120
  110 PTBJ2(U)=MIN0(PTBJ1(U),PTBJ2(V))
      PTBJ1(U)=PTBJ1(V)
  120 IF(KONT.EQ.1)GO TO 130
      V=PILA(KONT)
      U=PILA(KONT-1)
      GO TO 35
  130 V=1
      U=0
      GO TO 35
  140 DO 150 I=1,NV
      DO 150 J=1,NV
      IF(IA(I,J).NE.0) IA(I,J)=1
  150 CONTINUE
C
C  ** LLAMADA A LA SUBRUTINA QUE ORDENA LOS TERMINOS DE LA NUEVA MATRIZ
C     DE ADYACENCIA USANDO EL D.F.S. **
C
      CALL ORD(ISAL,NV,NA,N2,FROND,NUMB,PTBJ1,PTBJ2,FI,IB,IAA)
C
C  ** LLAMADA A LA SUBRUTINA QUE OBTIENE EL CIRCUITO ORIGINAL Y LOS CA-
C     MINOS DE LOS DISTINTOS SEGMENTOS Y LOS REPRESENTA EN EL PLANO
C     COMPROBANDO, POR TANTO, LA PLANARIDAD **
C
      CALL CAD(S,NA,NV,EE,EEV,V1,N2,IAA,NUMB,CAM,PILA,FI,BOC,ALMAC,
     &F,CADE,K1,JF,KF)
      RETURN
      END
C
C
      SUBROUTINE CAD(S,NA,NV,EE,EEV,V1,NF,IAA,NUMB,CAM,PILA,SIG,B,
     &ALMAC,F,CADE,K1,JF,KF)
C     ************************************************************
C
C SUBRUTINA QUE OBTIENE EL CIRCUITO ORIGINAL Y LOS CAMINOS DE LOS DIS-
C TOS SEGMENTOS Y LOS REPRESENTA EN EL PLANO, COMPROBANDO, POR TANTO,
C LA PLANARIDAD
C
```

```
      EMA IAA(50,10)
      INTEGER S,U,V,W,VW,FF,FREE,K,Y,NE1,NE0,SAVE,RR,SS,SSTT,EE,EEV,V1
      INTEGER PILA(NV),CAM(EEV),NUMB(V1)
      INTEGER B(NA),ALMAC(EE),SIG(NA),F(NA),CADE(NA)
C
C RENUMERACION
C
      M=1
      DO 1 I=1,NV
      IF(NUMB(I).LE.0) GO TO 1
      IF(NUMB(I).GT.M) M=NUMB(I)
    1 CONTINUE
      MM=M
C
C  ** LLAMADA A LA SUBRUTINA QUE MODIFICA LA MATRIZ DE ORDENACION **
C
      CALL CA4(NV,IAA,NUMB,ALMAC,MM)
      DO 3 I=1,NV
      DO 2 J=1,9
      IF(IAA(I,J).EQ.0) GO TO 3
      IAA(I,J)=NUMB(IAA(I,J))
    2 CONTINUE
    3 CONTINUE
C
C PUESTA A CERO
C
      DO 5 I=1,V1
      PILA(I)=0
      ALMAC(I)=0
    5 CONTINUE
      DO 8 I=1,NA
      B(I)=0
      SIG(I)=0
    8 CONTINUE
      FF=0
      NE0=0
      NE1=0
      STA0=0
      FREE=1
      SS=0
      SSTT=0
      U=0
      V=S
      KONT=1
      PILA(1)=V
      CADE(V)=1
      K1=1
      K2=0
   10 K2=K2+1
      CAM(K1)=-K2
      K1=K1+1
   15 CAM(K1)=V
   20 DO 30 J=1,10
      IF(IAA(V,J).GT.0) GO TO 280
      IF(IAA(V,J).EQ.0) GO TO 40
   30 CONTINUE
   40 IF(KONT.LE.1) GO TO 500
C
C SUPRESION DE ENTRADAS A FICHERO Y BLOQUES CORRESPONDIENTES A VERTICES
C NO MAYORES QUE U.
C
```

```
   45 DO 100 I=FF,1,-2
      K=B(I-1)
      Y=B(I)
      IF(K.EQ.0) GO TO 60
      IF(Y.EQ.0) GO TO 70
      IF(ALMAC(K).GE.U.AND.ALMAC(Y).GE.U) GO TO 80
      IF(ALMAC(K).LT.U) GO TO 50
      B(I-1)=0
   50 IF(ALMAC(Y).LT.U) GO TO 100
      B(I)=0
      GO TO 100
   60 IF(Y.EQ.0) GO TO 80
      IF(ALMAC(Y).LT.U) GO TO 100
      GO TO 80
   70 IF(ALMAC(K).LT.U) GO TO 100
      GO TO 80
C
C  ** LLAMADA A LA SUBRUTINA QUE SUPRIME DOS ELEMENTOS DEL CONJUNTO B **
C
   80 CALL SU2(I,FF,B,NA)
      IF(FF.EQ.0) GO TO 110
      GO TO 45
  100 CONTINUE
  110 IF(NE1.EQ.0) GO TO 120
      IF(ALMAC(NE1).LT.U) GO TO 120
      NE1=SIG(NE1)
      GO TO 110
  120 IF(NE0.EQ.0) GO TO 130
      IF(ALMAC(NE0).LT.U) GO TO 130
      NE0=SIG(NE0)
      GO TO 120
  130 IF(CADE(U).EQ.CADE(V)) GO TO 270
C
C TODO SEGMENTO CON LA PRIMERA ARISTA (U,V) HA SIDO EMPOTRADO.
C NUEVOS BLOQUES DEBEN SER MOVIDOS DE DERECHA A IZQUIERDA.
C
      LL=0
  135 DO 200 I=FF,1,-2
      K=B(I-1)
      Y=B(I)
      IF(NE1.EQ.0) GO TO 146
      IF(ALMAC(NE1).EQ.0) GO TO 200
  141 IF(K.EQ.0) GO TO 147
      IF(ALMAC(K).GT.F(CADE(V))) GO TO 155
  142 IF(Y.EQ.0) GO TO 148
      IF(ALMAC(Y).GT.F(CADE(V))) 149,200
  146 IF(STA0.EQ.0) 200,141
  147 IF(STA0.GT.F(CADE(V))) 155,142
  148 IF(STA0.GT.F(CADE(V))) 149,200
  149 IF(LL.NE.0) GO TO 151
C
C  ** LLAMADAS A SUBRUTINAS QUE REALIZAN OPERACIONES AUXILIARES **
C
      CALL VA1(SAVE,NE0)
      CALL VA2(NE1,NE0)
  151 IF(LL.EQ.-1) CALL VA1(SAVE,NE1)
      IF(LL.LE.0) GO TO 152
      CALL VA1(SAVE,SIG(LL))
      CALL VA2(NE1,SIG(LL))
  152 IF(Y.EQ.0) CALL VA3(NE1,NE0,SAVE)
```

```
      IF(Y.EQ.-1) CALL VA3(NE1,NE1,SAVE)                                CAD-966
      IF(Y.GT.0) CALL VA3(NE1,SIG(Y),SAVE)                              CAD-967
      LL=Y                                                              CAD-968
      GO TO 160                                                         CAD-969
  155 IF(Y.EQ.0) GO TO 156                                              CAD-970
      IF(ALMAC(Y).GT.F(CADE(V))) GO TO 490                              CAD-971
      GO TO 158                                                         CAD-972
  156 IF(STA0.GT.F(CADE(V))) GO TO 490                                  CAD-973
  158 LL=K                                                              CAD-974
C                                                                       CAD-975
C  ** LLAMADA A LA SUBRUTINA QUE SUPRIME DOS ELEMENTOS DEL CONJUNTO B ** CAD-976
C                                                                       CAD-977
  160 CALL SU2(I,FF,B,NA)                                               CAD-978
      IF(FF.EQ.0) GO TO 260                                             CAD-979
      GO TO 135                                                         CAD-980
  200 CONTINUE                                                          CAD-981
C                                                                       CAD-982
C EL BLOQUE SOBRE B DEBE SER COMBINADO CON OTROS BLOQUES YA SUPRIMIDOS.  CAD-983
C                                                                       CAD-984
  210 DO 250 I=FF,1,-2                                                  CAD-985
      X=B(I-1)                                                          CAD-986
      Y=B(I)                                                            CAD-987
      IF(X.NE.0) GO TO 250                                              CAD-988
      IF(LL.NE.0.OR.Y.NE.0) GO TO 220                                   CAD-989
C                                                                       CAD-990
C  ** LLAMADA A LA SUBRUTINA QUE SUPRIME DOS ELEMENTOS DEL CONJUNTO B ** CAD-991
C                                                                       CAD-992
      CALL SU2(I,FF,B,NA)                                               CAD-993
      IF(FF.EQ.0) GO TO 260                                             CAD-994
      GO TO 210                                                         CAD-995
  220 B(I-1)=LL                                                         CAD-996
  250 CONTINUE                                                          CAD-997
C                                                                       CAD-998
C SUPRESION DE LA MARCA FIN DE FICHERO SOBRE EL FICHERO DE LA DERECHA.  CAD-999
C                                                                       CAD-000
  260 IF(NE1.EQ.0) CALL VA2(NE0,NE1)                                    CAD-001
      IF(NE1.GT.0) CALL VA2(SIG(NE1),NE1)                               CAD-002
  270 PILA(KONT)=0                                                      CAD-003
      KONT=KONT-1                                                       CAD-004
      V=PILA(KONT)                                                      CAD-005
      U=PILA(KONT-1)                                                    CAD-006
      GO TO 15                                                          CAD-007
  280 W=IAA(V,J)                                                        CAD-008
      IAA(V,J)=-IAA(V,J)                                                CAD-009
      K1=K1+1                                                           CAD-010
      CAN(K1)=W                                                         CAD-011
      IF(V.GT.W) GO TO 330                                              CAD-012
      U=V                                                               CAD-013
      V=W                                                               CAD-014
      KONT=KONT+1                                                       CAD-015
      PILA(KONT)=W                                                      CAD-016
      CADE(W)=K2                                                        CAD-017
      GO TO 20                                                          CAD-018
  330 F(K2)=W                                                           CAD-019
C                                                                       CAD-020
C CAMBIAR LOS BLOQUES DE ENTRADA DE IZQUIERDA A DERECHA DE FORMA QUE    CAD-021
C PUEDA SER EMPOTRADA POR LA IZQUIERDA                                  CAD-022
C                                                                       CAD-023
      LL=0                                                              CAD-024
      RR=-1                                                             CAD-025
  335 IF(LL.EQ.0) GO TO 340                                             CAD-026
```

```
      IF(LL.EQ.-1) GO TO 350
      IF(SIG(LL).EQ.0) GO TO 360
      IF(ALMAC(SIG(LL)).LE.W) 360,400
  340 IF(NEO.EQ.0) GO TO 360
      IF(ALMAC(NEO).LE.W) 360,400
  350 IF(NE1.EQ.0) GO TO 360
      IF(ALMAC(NE1).LE.W) 360,400
  360 IF(RR.EQ.0) GO TO 370
      IF(RR.EQ.-1) GO TO 380
      IF(SIG(RR).EQ.0) GO TO 460
      IF(ALMAC(SIG(RR)).LE.W) 460,400
  370 IF(NEO.EQ.0) GO TO 460
      IF(ALMAC(NEO).LE.W) 460,400
  380 IF(NE1.EQ.0) GO TO 460
      IF(ALMAC(NE1).LE.W) GO TO 460
  400 X=B(FF-1)
      Y=B(FF)
      IF(X.NE.0.AND.Y.NE.0) GO TO 420
      IF(X.EQ.0) GO TO 410
      IF(LL.EQ.0) GO TO 401
      IF(LL.EQ.-1) GO TO 402
      IF(SIG(LL).EQ.0) GO TO 460
      IF(ALMAC(SIG(LL)).LE.W) GO TO 460
      GO TO 405
  401 IF(NEO.EQ.0) GO TO 460
      IF(ALMAC(NEO).LE.W) GO TO 460
      GO TO 405
  402 IF(NE1.EQ.0) GO TO 460
      IF(ALMAC(NE1).LE.W) GO TO 460
C
C  ** LLAMADAS A UNA SERIE DE SUBRUTINAS QUE REALIZAN OPERACIONES AUXI-
C     LARES **
C
  405 IF(X.EQ.-1) CALL VA1(SAVE,NE1)
      IF(X.NE.-1) CALL VA1(SAVE,SIG(X))
      IF(RR.EQ.0.AND.X.EQ.-1) CALL VA2(NEO,NE1)
      IF(RR.GT.0.AND.X.EQ.-1) CALL VA2(SIG(RR),NE1)
      IF(RR.EQ.-1.AND.X.NE.-1) CALL VA2(NE1,SIG(X))
      IF(RR.GT.0.AND.X.NE.-1) CALL VA2(SIG(RR),SIG(X))
      IF(RR.EQ.0.AND.X.NE.-1) CALL VA2(NEO,SIG(X))
      IF(RR.EQ.-1.AND.LL.EQ.0) CALL VA3(NE1,NEO,SAVE)
      IF(RR.EQ.-1.AND.LL.EQ.-1) CALL VA3(NE1,NE1,SAVE)
      IF(RR.GT.0.AND.LL.EQ.0) CALL VA3(SIG(RR),NEO,SAVE)
      IF(RR.GT.0.AND.LL.GT.0) CALL VA3(SIG(RR),SIG(LL),SAVE)
      IF(RR.GT.0.AND.LL.EQ.-1) CALL VA3(SIG(RR),NE1,SAVE)
      IF(RR.EQ.0.AND.LL.GT.0) CALL VA3(NEO,SIG(LL),SAVE)
      IF(RR.EQ.-1.AND.LL.GT.0) CALL VA3(NE1,SIG(LL),SAVE)
      RR=X
      GO TO 440
  410 IF(RR.EQ.0) GO TO 411
      IF(RR.EQ.-1) GO TO 412
      IF(SIG(RR).EQ.0) GO TO 460
      IF(ALMAC(SIG(RR)).LE.W) GO TO 460
      GO TO 415
  411 IF(NEO.EQ.0) GO TO 460
      IF(ALMAC(NEO).LE.W) GO TO 460
      GO TO 415
  412 IF(NE1.EQ.0) GO TO 460
      IF(ALMAC(NE1).LE.W) GO TO 460
  415 IF(Y.NE.0) RR=Y
```

```
      GO TO 440
  420 IF(LL.EQ.0) NNN=NE0
      IF(LL.EQ.-1) NNN=NE1
      IF(LL.GT.0) NNN=SIG(LL)
      IF(NNN.EQ.0) SSTT=STA0
      IF(NNN.NE.0) SSTT=ALMAC(NNN)
      IF(SSTT.GT.W) GO TO 430
      LL=K
      RR=Y
      GO TO 440
  430 IF(RR.EQ.0) NNN=NE0
      IF(RR.EQ.-1) NNN=NE1
      IF(RR.GT.0) NNN=SIG(RR)
      IF(NNN.EQ.0) SSTT=STA0
      IF(NNN.NE.0) SSTT=ALMAC(NNN)
      IF(SSTT.GT.W) GO TO 550
C
C  ** LLAMADAS A UNA SERIE DE SUBRUTINAS QUE REALIZAN OPERACIONES AUXI-
C     LIARES **
C
      IF(RR.EQ.0) CALL VA1(SAVE,NE0)
      IF(RR.EQ.-1) CALL VA1(SAVE,NE1)
      IF(RR.GT.0) CALL VA1(SAVE,SIG(RR))
      IF(RR.EQ.0.AND.LL.EQ.-1) CALL VA3(NE0,NE1,SAVE)
      IF(RR.EQ.-1.AND.LL.EQ.0) CALL VA3(NE1,NE0,SAVE)
      IF(RR.GT.0.AND.LL.EQ.0) CALL VA3(SIG(RR),NE0,SAVE)
      IF(RR.GT.0.AND.LL.EQ.-1) CALL VA3(SIG(RR),NE1,SAVE)
      IF(RR.GT.0.AND.LL.GT.0) CALL VA3(SIG(RR),SIG(LL),SAVE)
      IF(RR.EQ.0.AND.LL.GT.0) CALL VA3(NE0,SIG(LL),SAVE)
      IF(RR.EQ.-1.AND.LL.GT.0) CALL VA3(NE1,SIG(LL),SAVE)
      SAVE=SIG(X)
      SIG(X)=SIG(Y)
      SIG(Y)=SAVE
      LL=Y
      RR=K
  440 II=FF
C
C  ** LLAMADA A LA SUBRUTINA QUE SUPRIME DOS ELEMENTOS DEL CONJUNTO B **
C
      CALL SU2(II,FF,B,NA)
      IF(FF.EQ.0) GO TO 460
      GO TO 335
  460 DO 467 I=K1,1,-1
      IF(CAN(I).NE.-K2) GO TO 467
      SS=CAN(I+1)
      GO TO 468
  467 CONTINUE
C
C ANADIR EL CAMINO AL FICHERO DE LA IZQUIERDA SI EL CAMINO ES NORMAL
C
  468 IF(F(CADE(SS)).GE.W.AND.K2.GT.1) GO TO 470
      IF(LL.EQ.0) LL=FREE
      ALMAC(FREE)=W
      SIG(FREE)=NE0
      NE0=FREE
      FREE=FREE+1
C
C ANADIR NUEVOS BLOQUES CORRESPONDIENTES A LOS ANTIGUOS BLOQUES COMBINA-
C DOS. NUEVOS BLOQUES PUEDEN SER VACIADOS SI EL SEGMENTO QUE CONTIENE AL
C CAMINO ACTUAL NO ES UNA SOLA RAMA.
C
```

```
  470 IF(RR.EQ.-1) RR=0
C
C  ** LLAMADA A LA SUBRUTINA QUE ALMACENA DOS NUEVOS VALORES EN EL CON-
C     JUNTO B **
C
      IF(LL.NE.0.OR.RR.NE.0.OR.V.NE.SS) CALL AD2(NA,LL,RR,B,FF)
      IF(V.EQ.SS) GO TO 480
C SI EL SEGMENTO QUE CONTIENE AL CAMINO ACTUAL NO ES UNA SOLA RAMA,
C ANADIR UNA MARCA FIN DE FICHERO AL FICHERO DE LA DERECHA.
      ALMAC(FREE)=0
      SIG(FREE)=NE1
      NE1=FREE
      FREE=FREE+1
  480 K1=K1+1
      GO TO 10
  490 KF=K1-2
      DO 495 I=K1-2,1,-1
      IF(CAN(I).LT.0) GO TO 496
  495 CONTINUE
  496 JF=I+1
      GO TO 560
  500 K1=K1-1
      CAN(K1+1)=0
      GO TO 600
  550 KF=K1
      DO 555 I=K1,1,-1
      IF(CAN(I).LT.0) GO TO 556
  555 CONTINUE
  556 JF=I+1
  560 DO 570 I=JF,KF
      DO 565 J=1,NV
      IF(CAN(I).EQ.NUMB(J)) GO TO 567
  565 CONTINUE
      GO TO 570
  567 CAN(I)=J
  570 CONTINUE
  600 RETURN
      END
C
C
      SUBROUTINE TRI(S,NV,NA,EE,EEV,V1,EEE,K1,N2,IONT,CAN,NUMB,PILA,
     &TICON,ALTO,PAD,ND,PTBJ1,PTBJ2,TSTA,INDT,ESTA,GRADO,NUENU,A1,IAA,
     &IB)
C     *****************************************************************
C
C SUBRUTINA QUE OBTIENE LAS COMPONENTES TRICONEXAS DE UN GRAFO BICONEXO
C
      ENA IB(150,50),IAA(50,10)
      INTEGER CODE,V,S,U,W,VW,EE,EEV,V1,EEE
      INTEGER NUMB(NV),ESTA(EE),TSTA(NA),INDT(NA)
      INTEGER PILA(V1),PTBJ1(NV),PTBJ2(NV),TICON(EEE),NUENU(NV)
      INTEGER CAN(EEV),ND(NV),PAD(NV),ALTO(NV),A1(NV),GRADO(NV)
C
C PUESTA A CERO
C
      DO 10 I=1,EE
      ESTA(I)=0
   10 CONTINUE
      DO 15 I=1,EEV
      CAN(I)=0
```

```
   15 CONTINUE
      DO 20 I=1,EEE
      TICON(I)=0
   20 CONTINUE
      DO 30 I=1,NV
      A1(I)=0
      GRADO(I)=0
      NUENU(I)=0
      PAD(I)=0
      ALTO(I)=0
      PILA(I)=0
   30 CONTINUE
      PILA(V1)=0
      DO 40 I=1,NA
      TSTA(I)=0
      INDT(I)=0
   40 CONTINUE
      N=1
      DO 50 I=1,NV
      IF(NUMB(I).GT.N) N=NUMB(I)
   50 CONTINUE
      NN=N
      DO 160 J=1,NV
      K=0
      DO 150 I=1,NA
      IF(IB(I,J).LT.0) K=K+1
  150 CONTINUE
      GRADO(J)=K
  160 CONTINUE
C
C  ** LLAMADA A LA SUBRUTINA QUE CAMBIA LOS VECTORES DE ACUERDO A LA
C     PRIMERA NUMERACION **
C     NUMERACION **
C
      CALL CA3(NV,ND,NUMB,ESTA)
      CALL CA3(NV,PTBJ1,NUMB,ESTA)
      CALL CA3(NV,PTBJ2,NUMB,ESTA)
      CALL CA3(NV,GRADO,NUMB,ESTA)
      DO 170 I=1,NV
      DO 165 J=1,9
      IF(IAA(I,J).EQ.0) GO TO 170
      IF(IAA(I,J).LT.0) IAA(I,J)=-IAA(I,J)
  165 CONTINUE
  170 CONTINUE
C
C  ** LLAMADA A LA SUBRUTINA QUE CAMBIA LA ORDENACION PARA EL ANALISIS
C     DE LA TRICONECTIVIDAD **
C     TRICONECTIVIDAD **
C
      CALL CA1(NV,PTBJ1,PTBJ2,ESTA,IAA)
C
C NUEVA NUMERACION DE LOS VERTICES
C
      N2=0
      V=1
      KONT=1
      PILA(KONT)=1
      NUENU(V)=1
      K1=1
      K2=0
  210 K2=K2+1
```

```
      CAN(K1)=-K2
      K1=K1+1
  215 CAN(K1)=V
  220 DO 250 J=1,9
      IF(IAA(V,J).EQ.0) GO TO 260
      IF(IAA(V,J).GT.0) GO TO 270
  250 CONTINUE
  260 IF(KONT.EQ.1) GO TO 290
      PILA(KONT)=0
      KONT=KONT-1
      N=N-1
      V=PILA(KONT)
      GO TO 215
  270 W=IAA(V,J)
      IAA(V,J)=-IAA(V,J)
      K1=K1+1
      CAN(K1)=W
      IF(NUENU(W).NE.0) GO TO 280
      NUENU(W)=N-ND(W)+1
      PAD(NUENU(W))=NUENU(V)
      KONT=KONT+1
      PILA(KONT)=W
      V=W
      GO TO 220
  280 IF(ALTO(NUENU(W)).EQ.0) ALTO(NUENU(W))=NUENU(V)
      K1=K1+1
      GO TO 210
  290 PILA(KONT)=0
C
C  ** LLAMADA A LA SUBRUTINA QUE CAMBIA LOS VECTORES ,FORMADOS A PARTIR
C     DE LA NUMERACION PRIMITIVA, SEGUN LA ULTIMA NUMERACION.
C
      CALL CA3(NV,ND,NUENU,ESTA)
      CALL CA3(NV,GRADO,NUENU,ESTA)
      CALL CA3(NV,PTBJ1,NUENU,ESTA)
      CALL CA3(NV,PTBJ2,NUENU,ESTA)
C
C CAMBIO DE LOS VECTORES PTBJ1 Y PTBJ2
C
      DO 295 I=1,NV
      IF(PTBJ1(I).LE.0) GO TO 295
      PTBJ1(I)=NUENU(PTBJ1(I))
      PTBJ2(I)=NUENU(PTBJ2(I))
  295 CONTINUE
C
C  ** LLAMADA A LA SUBRUTINA QUE CAMBIA LA MATRIZ DE ORDENACION SEGUN LA
C     NUEVA NUMERACION **
C
      CALL CA4(NV,IAA,NUENU,ESTA,MM)
      DO 305 I=1,NV
      DO 304 J=1,9
      IF(IAA(I,J).EQ.0) GO TO 305
      IAA(I,J)=NUENU(IAA(I,J))
  304 CONTINUE
  305 CONTINUE
      DO 350 I=1,K1
      IF(CAN(I).LE.0) GO TO 350
      CAN(I)=NUENU(CAN(I))
  350 CONTINUE
      DO 360 I=1,MM
```

```
      A1(I)=IAA(I,1)
  360 CONTINUE
C
C  ** LLAMADA A LA SUBRUTINA QUE OBTIENE LAS COMPONENTES ESCINDIDAS DE
C     GRAFO BCONEXO **
C
      CALL ESC(NA,NV,EE,EEV,V1,EEE,K1,N2,IONT,CAN,PILA,TSTA,INDT,ESTA,
     &ALTO,PAD,ND,GRADO,A1,PTBJ1,PTBJ2,TICON,IAA)
      DO 400 I=1,IONT
      IF(TICON(I).LT.0) GO TO 400
      DO 380 J=1,NV
      IF(NUENU(J).EQ.TICON(I)) GO TO 390
  380 CONTINUE
  390 JJ=J
      DO 395 J=1,NV
      IF(NUMB(J).EQ.JJ) GO TO 398
  395 CONTINUE
  398 TICON(I)=J
  400 CONTINUE
      KT=0
  420 RETURN
      END
C
C
      SUBROUTINE ESC(NA,NV,EE,EEV,V1,EEE,K1,NF,NN,CAN,PILA,TSTA,INDT,
     &ESTA,ALTO,PAD,ND,GRADO,A1,PTBJ1,PTBJ2,COMPO,IAA)
C     ***************************************************************
      EMA IAA(50,10)
      INTEGER FLAG,V,U,W,VW,FF,GG,HH,N,A,B,BB,T,XX,YY,SAV1,SAV2,FFF,Y
      INTEGER EE,EEV,V1,EEE
      INTEGER TSTA(NA),ESTA(EE),PAD(NV),GRADO(NV),A1(NV)
      INTEGER PTBJ1(NV),PTBJ2(NV),ND(NV),ALTO(NV),PILA(V1)
      INTEGER CAN(EEV),COMPO(EEE),INDT(NA)
      DO 10 I=1,EE
   10 ESTA(I)=0
      JJ=0
      FF=0
      GG=0
      HH=0
      KONT=1
      SAV1=0
      SAV2=0
      FLAG=0
      V=1
      U=0
      T=0
      PILA(KONT)=V
   40 DO 50 I=1,10
      IF(IAA(V,I).GT.0) GO TO 280
      IF(IAA(V,I).EQ.0) GO TO 60
   50 CONTINUE
   60 IF(KONT.LE.1) GO TO 700
      JJJ=JJ
C
C  ** LLAMADA A LA SUBRUTINA USADA PARA ELIMINAR ARISTAS MULTIPLES **
C     EN LAS COMPONENTES TRICONEXAS **
C
      CALL MUL(EE,EEE,NV,GG,U,V,HH,JJ,ESTA,GRADO,COMPO,2)
      IF(JJJ.NE.JJ) GO TO 63
C
C  ** LLAMADA A LA SUBRUTINA QUE ALMACENA DOS ELEMENTOS EN EL CONJUNTO
```

```
C     ESTA **
C
      CALL AD2(EE,U,V,ESTA,GG)
   63 IF(U.EQ.1) GO TO 205
   65 DO 200 I=FF,1,-3
      IF(FF.EQ.0) GO TO 75
      IF(INDT(FF).NE.0) GO TO 75
      N=TSTA(I-2)
      A=TSTA(I-1)
      B=TSTA(I)
      IF((GRADO(V).EQ.2.AND.A1(V).GT.V).OR.B.EQ.A) 70,200
C
C PAR DE SEPARACION TIPO 2
C
   70 IF(U.NE.A.OR.PAD(B).NE.A) GO TO 75
C
C  ** LLAMADA A LA SUBRUTINA EMPLEADA PARA ELIMINAR TRES ELEMENTOS DEL
C     CONJUNTO TSTA **
C
      CALL SU3(I,FF,TSTA,NA)
      GO TO 65
   75 IF(GRADO(V).EQ.2.AND.A1(V).GT.V) 80,113
   80 JJ=JJ+1
      IF(ESTA(GG-3).NE.V) T=ESTA(GG-3)
      IF(ESTA(GG-2).NE.V) T=ESTA(GG-2)
C
C  ** LLAMADAS A LA SUBRUTINA QUE ANADE DOS ELEMENTOS AL CONJUNTO COMPO
C     **
C
      CALL AD2(EEE,U,V,COMPO,HH)
      CALL AD2(EEE,V,T,COMPO,HH)
      CALL AD2(EEE,U,T,COMPO,HH)
      HH=HH+1
      COMPO(HH)=-JJ
C
C  ** LLAMADAS A LA SUBRUTINA QUE SUPRIME DOS ELEMENTOS DEL CONJUNTO ES-
C     TA **
C
      CALL SU2(0,GG,ESTA,EE)
      CALL SU2(0,GG,ESTA,EE)
      GRADO(V)=GRADO(V)-2
      GRADO(U)=GRADO(U)-1
      GRADO(T)=GRADO(T)-1
      DO 100 J=GG,1,-2
      XX=ESTA(J-1)
      YY=ESTA(J)
      IF(XX.EQ.T.AND.YY.EQ.U) 90,100
   90 FLAG=1
      SAV1=XX
      SAV2=YY
C
C  ** LLAMADA A LA SUBRUTINA QUE SUPRIME DOS ELEMENTOS DEL CONJUNTO ESTA
C     **
C
      CALL SU2(J,GG,ESTA,EE)
      GO TO 160
  100 CONTINUE
      GO TO 170
  113 IF(FF.EQ.0) GO TO 205
      IF(INDT(FF).NE.0) GO TO 205
```

```
      JJ=JJ+1
C
C  ** LLAMADA A LA SUBRUTINA QUE SUPRIME TRES ELEMENTOS DEL CONJUNTO
C     TSTA **
C
      CALL SU3(I,FF,TSTA,NA)
  116 IF(GG.EQ.0) GO TO 155
      DO 150 J=GG,1,-2
      XX=ESTA(J-1)
      YY=ESTA(J)
      IF(A.LE.XX.AND.XX.LE.N.AND.A.LE.YY.AND.YY.LE.N) 120,150
  120 IF((XX.EQ.A.AND.YY.EQ.B).OR.(XX.EQ.B.AND.YY.EQ.A)) GO TO 130
C
C  ** LLAMADA A LA SUBRUTINA QUE SUPRIME DOS ELEMENTOS DEL CONJUNTO ESTA
C     **
C
      CALL SU2(J,GG,ESTA,EE)
C
C  ** LLAMADA A LA SUBRUTINA QUE ANADE DOS ELEMENTOS AL CONJUNTO COMPO
C     **
C
      CALL AD2(EEE,XX,YY,COMPO,HH)
      GRADO(XX)=GRADO(XX)-1
      GRADO(YY)=GRADO(YY)-1
      GO TO 116
  130 FLAG=1
      SAV1=XX
      SAV2=YY
C
C  ** LLAMADA A LA SUBRUTINA QUE SUPRIME DOS ELEMENTOS DEL CONJUNTO ESTA
C     **
C
      CALL SU2(J,GG,ESTA,EE)
      GO TO 116
  150 CONTINUE
C
C  ** LLAMADA A LA SUBRUTINA QUE ANADE DOS ELEMENTOS AL CONJUNTO COMPO
C     **
C
  155 CALL AD2(EEE,A,B,COMPO,HH)
      HH=HH+1
      COMPO(HH)=-JJ
      T=B
C
C  ** LLAMADAS A SUBRUTINAS QUE REALIZAN OPERACIONES AUXILIARES **
C
  160 IF(FLAG.EQ.1) CALL PA1(FLAG,JJ,SAV1,SAV2,HH,T,U,COMPO,GRADO,NV,
     &EEE)
  170 CALL PA2(GG,U,T,JJ,ESTA,GRADO,PAD,A1(U),K1,CAN,V,EEV,NV,EE)
      GO TO 65
  290 CONTINUE
C
C PAR DE SEPARACION TIPO 1
C
  295 IF(PTBJ2(V).GE.U.AND.(PTBJ1(V).NE.1.OR.PAD(U).NE.1.OR.V.GT.3)) 210
     &,265
  210 JJ=JJ+1
  215 IF(GG.EQ.0) GO TO 255
      DO 250 I=GG,1,-2
      XX=ESTA(I-1)
      YY=ESTA(I)
```

```
      IF((V.LE.XX.AND.XX.LT.V+ND(V)).OR.(V.LE.YY.AND.YY.LT.V+ND(V))) 220
     &,250
C
C  ** LLAMADA A LA SUBRUTINA QUE SUPRIME DOS ELEMENTOS DEL CONJUNTO ESTA
C     **
C
  220 CALL SU2(I,GG,ESTA,EE)
C
C  ** LLAMADA A LA SUBRUTINA QUE ANADE DOS ELEMENTOS AL CONJUNTO COMPO
C     **
C
      CALL AD2(EEE,XX,YY,COMPO,HH)
      GRADO(XX)=GRADO(XX)-1
      GRADO(YY)=GRADO(YY)-1
      GO TO 215
  250 CONTINUE
C
C  ** LLAMADA A LA SUBRUTINA QUE ANADE DOS ELEMENTOS AL CONJUNTO COMPO
C     **
C
  255 IF(U.NE.1.OR.PTBJ1(V).NE.1) CALL AD2(EEE,U,PTBJ1(V),COMPO,HH)
      HH=HH+1
      COMPO(HH)=-JJ
      IF(A1(U).EQ.V) A1(U)=PTBJ1(V)
C
C  ** LLAMADA A LA SUBRUTINA EMPLEADA PARA ELIMINAR ARISTAS MULTIPLES EN
C     LAS COMPONENTES TRICONEXAS **
C
      CALL MUL(EE,EEE,NV,GG,U,PTBJ1(V),HH,JJ,ESTA,GRADO,COMPO,1)
      IF(PTBJ1(V).NE.PAD(U)) GO TO 260
      JJ=JJ+1
C
C  ** LLAMADAS A LA SUBRUTINA QUE ANADE DOS ELEMENTOS AL CONJUNTO COMPO
C     **
C
      CALL AD2(EEE,U,PTBJ1(V),COMPO,HH)
      CALL AD2(EEE,U,PTBJ1(V),COMPO,HH)
      CALL AD2(EEE,PTBJ1(V),U,COMPO,HH)
      HH=HH+1
      COMPO(HH)=-JJ
      GO TO 265
C
C  ** LLAMADA A LA SUBRUTINA QUE ANADE DOS ELEMENTOS AL CONJUNTO ESTA **
C
  260 CALL AD2(EE,U,PTBJ1(V),ESTA,GG)
      GRADO(U)=GRADO(U)+1
      GRADO(PTBJ1(V))=GRADO(PTBJ1(V))+1
  265 DO 266 I=1,K1-1
      IF(CAM(I).GE.0) GO TO 266
      IF(CAM(I+1).EQ.U.AND.CAM(I+2).EQ.V) GO TO 267
  266 CONTINUE
      GO TO 270
C
C LA ARISTA (U,V) ES LA PRIMERA DE UN CAMINO GENERADO
C
  267 IF(FF.EQ.0) GO TO 279
      IF(INDT(FF).NE.0) GO TO 269
      DO 268 I=FF,1,-3
C
C  ** LLAMADA A LA SUBRUTINA QUE SUPRIME TRES ELEMENTOS AL CONJUNTO TSTA
```

```
C      **
C
       CALL SU3(I,FF,TSTA,NA)
       GO TO 267
   268 CONTINUE
   269 INDT(FF)=0
   270 IF(FF.EQ.0) GO TO 279
       IF(INDT(FF).NE.0) GO TO 279
       DO 275 I=FF,1,-3
       N=TSTA(I-2)
       A=TSTA(I-1)
       B=TSTA(I)
       IF(ALTO(U).LE.N) GO TO 275
C
C  ** LLAMADA A LA SUBRUTINA QUE SUPRIME TRES ELEMENTOS AL CONJUNTO TSTA
C      **
C
       CALL SU3(I,FF,TSTA,NA)
       GO TO 270
   275 CONTINUE
   279 PILA(KONT)=0
       KONT=KONT-1
       V=PILA(KONT)
       U=PILA(KONT-1)
       GO TO 40
   280 W=IAA(V,I)
       IAA(V,I)=-IAA(V,I)
       IF(V.GT.W) GO TO 510
       DO 400 I=1,K1-1
       IF(CAM(I).GE.0) GO TO 400
       IF(CAM(I+1).EQ.V.AND.CAM(I+2).EQ.W) GO TO 410
   400 CONTINUE
       GO TO 505
C
C LA ARISTA (V,W) ES LA PRIMERA DE UN CAMINO GENERADO
C
   410 Y=0
       FFF=FF
   420 IF(FF.EQ.0) GO TO 455
       IF(INDT(FF).NE.0) GO TO 455
       DO 450 I=FF,1,-3
       N=TSTA(I-2)
       A=TSTA(I-1)
       B=TSTA(I)
       IF(A.LE.PTBJ1(W)) GO TO 450
       Y=MAX0(Y,N)
C
C  ** LLAMADA A LA SUBRUTINA QUE SUPRIME TRES ELEMENTOS DEL CONJUNTO
C      TSTA **
C
       CALL SU3(I,FF,TSTA,NA)
       BB=B
       GO TO 420
   450 CONTINUE
   455 IF(FFF.EQ.FF) GO TO 460
       NN=MAX0(Y,W+ND(W)-1)
C
C  ** LLAMADAS A LA SUBRUTINA QUE ANADE TRES ELEMENTOS AL CONJUNTO TSTA
C      **
C
       CALL AD3(FF,NN,PTBJ1(W),BB,TSTA,NA)
```

```
      GO TO 505
  460 CALL AD3(FF,W+ND(W)-1,PTBJ1(W),V,TSTA,NA)
  505 INDT(FF)=1
      KONT=KONT+1
      PILA(KONT)=W
      U=V
      V=W
      GO TO 40
  510 DO 550 I=1,K1-1
      IF(CAM(I).GE.0) GO TO 550
      IF(CAM(I+1).EQ.V.AND.CAM(I+2).EQ.W) GO TO 560
  550 CONTINUE
      GO TO 620
C
C LA ARISTA (V,W) ES LA PRIMERA (Y ULTIMA) DE UN CAMINO GENERADO
C
  560 Y=0
      FFF=FF
  570 IF(FF.EQ.0) GO TO 605
      IF(INDT(FF).NE.0) GO TO 605
      DO 600 I=FF,1,-3
      H=TSTA(I-2)
      A=TSTA(I-1)
      B=TSTA(I)
      IF(A.LE.W) GO TO 600
      Y=MAX0(Y,H)
C
C ** LLAMADA A LA SUBRUTINA QUE SUPRIME TRES ELEMENTOS DEL CONJUNTO TSTA
C    **
C
      CALL SU3(I,FF,TSTA,NA)
      BB=B
      GO TO 570
  600 CONTINUE
  605 IF(FFF.EQ.FF) GO TO 610
C
C  ** LLAMADAS A LA SUBRUTINA QUE ANADE TRES ELEMENTOS AL CONJUNTO TSTA
C     **
C
      CALL AD3(FF,Y,W,BB,TSTA,NA)
      GO TO 620
  610 CALL AD3(FF,V,W,V,TSTA,NA)
  620 JJJ=JJ
C
C  ** LLAMADA A LA SUBRUTINA EMPLEADA PARA ELIMINAR ARISTAS MULTIPLES EN
C     LAS COMPONENTES TRICONEXAS **
C
      CALL MUL(EE,EEE,NV,GG,V,W,NN,JJ,ESTA,GRADO,COMPO,2)
      IF(JJJ.NE.JJ.OR.W.EQ.PAD(V)) GO TO 625
C
C  ** LLAMADA A LA SUBRUTINA QUE ANADE DOS ELEMENTOS AL CONJUNTO ESTA **
C
      CALL AD2(EE,V,W,ESTA,GG)
      GO TO 40
  625 IF(W.NE.PAD(V)) GO TO 40
  630 JJ=JJ+1
C
C  ** LLAMADAS A LA SUBRUTINA QUE ANADE DOS ELEMENTOS AL CONJUNTO COMPO
C     **
C
```

```
      CALL AD2(EEE,V,W,COMPO,HH)
      CALL AD2(EEE,V,W,COMPO,HH)
      CALL AD2(EEE,W,V,COMPO,HH)
      HH=HH+1
      COMPO(HH)=-JJ
      GRADO(V)=GRADO(V-1)
      GRADO(W)=GRADO(W-1)
      GO TO 40
  700 RETURN
      END
$
```

```
FTN4
$EMA(IJK,3)
C
C
C     **************
      PROGRAM PL3(5)
C     **************
C
C
C SEGMENTO 3 .................................. OBTENCION DEL GRAFO DUAL
C
C
      COMMON IPAR(4),NAME(3)
      COMMON IENT,ISAL,IEAUX,ISAUX,NV,NA,KKTT
      COMMON NOM(40),JAUX(2200)
      COMMON /IJK/ IB(150,50),IA(100,50),X(300),Y(300),IAA(50,10)
      IOP=0
C
C  ** LLAMADA A LA SUBRUTINA QUE APLICA EL TEST DE RECTANGULARIDAD **
C
   60 CALL REC(K1,NV,NA,JAUX(1),JAUX(101),IA)
C
C SI NO LO SUPERA SE HAN DE SUSTITUIR LAS ARISTAS AFECTADAS POR OTRAS
C
      IF(K1.EQ.0) GO TO 110
      WRITE(ISAL,3001)
      DO 100 I=1,K1
      II=4*(I-1)+1
      IT=4*(I-1)+4
      WRITE(ISAL,3002) (JAUX(J),J=II,IT)
  100 CONTINUE
      WRITE(ISAL,3003)
C
C  ** LLAMADA A LA SUBRUTINA QUE LEE LAS ARISTAS QUE SUSTITUYEN A OTRAS
C     **
C
      CALL LE1(IEAUX,ISAL,ISAUX,1,IA,IB)
C
C  ** LLAMADA A LA SUBRUTINA QUE OBTIENE UN CIRCUITO DEL GRAFO **
C
      CALL CIR(NV,NA,NV+1,JAUX(3*NA+3*NV+1),JAUX(3*NA+4*NV+2),JAUX(9*NA+
     &6*NV+3),JAUX(9*NA+7*NV+3),KK,LONT,IA)
C
C  ** LLAMADA A LA SUBRUTINA QUE OBTIENE LAS COORDENADAS DE LOS VERTICES
C     APLICANDO EL ALGORITMO DE TUTTE **
C
      CALL OBC(NV,NA,KK,LONT,JAUX(3*NA+3*NV+1),JAUX(9*NA+7*NV+3),X(1),
     &Y(1),X(NV+1),Y(NV+1),JAUX(9*NA+5*NV+3),IA,IB,KKTT,ISAL,ISAUX)
C
C SE VUELVE A COMPROBAR LA RECTANGULARIDAD
C
      GO TO 60
  110 IF(IOP.NE.0) GO TO 120
      NAA=NA
  111 WRITE(ISAL,3004)
C
C  ** LLAMADA A LA SUBRUTINA QUE LEE LAS NUEVAS ARISTAS INTRODUCIDAS **
C
```

```
      CALL LE2(IEAUX,ISAL,ISAUX,1,NA,NV,IA,IB)
      NEE=3*NV-7
      IF(NA.EQ.NEE) GO TO 118
      WRITE(ISAL,3005) NA,NEE
      IF(NA.LT.NEE) GO TO 111
      DO 116 I=NEE+1,NA
      J1=0
      J2=0
      DO 114 J=1,NV
      IF(IB(I,J).EQ.0) GO TO 115
      IF(J1.NE.0) GO TO 115
      J1=J
      IB(I,J1)=0
  114 CONTINUE
  115 J2=J
      IB(I,J2)=0
      IA(J1,J2)=0
      IA(J2,J1)=0
  116 CONTINUE
      NA=NEE
      WRITE(ISAL,3006)
      CALL LE1(IEAUX,ISAL,ISAUX,1,IA,IB)
  118 WRITE(ISAL,3007)
C
C  ** LLAMADA A LA SUBRUTINA QUE DIBUJA EL GRAFO TRIANGULADO **
C
  120 IF(ISSW(3).LT.0) GO TO 130
      R=5.
      CALL TUT(NA,NV,R,JAUX(3*NAA+3*NV+1),X,Y,IB,KKTT,ISAL)
C
C  ** LLAMADA A LA SUBRUTINA QUE TRASFORMA LAS COORDENADAS DE LOS VERTI-
C     CES INTERIORES **
C
  130 CALL DIC(ISAL,NV,NA,KONT,JONT,XMAX,YMAX,5*NV,JAUX(1),JAUX(9),
     &JAUX(9+5*NV),X(NV+1),Y(NV+1),IA,IB)
C
C  ** LLAMADA A LA SUBRUTINA QUE OBTIENE EL GRAFO DUAL **
C
      CALL DUA(ISAL,ISAUX,JONT,KONT,JAUX(9),JAUX(9+5*NV),X(NV+1),Y(NV+1)
     &,XMAX,YMAX)
C
C ELECCION DEL PARAMETRO OPCIONAL INTERACTIVO
C
      WRITE(ISAL,3008)
      READ(IEAUX,*) IOP
      IF(IOP.EQ.0) GO TO 220
C
C  ** LLAMADA A LA SUBRUTINA QUE LEE LAS ARISTAS QUE SUSTITUYEN A OTRAS
C
      CALL LE1(IEAUX,ISAL,ISAUX,1,IA,IB)
      GO TO 60
  220 STOP
C
C FORMATOS
C
 3001 FORMAT(//1X"MENSAJE !!SR!! EL GRAFO NO SUPERA EL TEST DE RECTANGUL
     &ARIDAD"//1X"LOS GRUPOS DE VERTICES AFECTADOS SON:"/2X"V1   V2   V3
     &V4"/2X"--   --   --   --")
 3002 FORMAT(4I4)
 3003 FORMAT(/1X"MENSAJE !!SR!! HAS DE SUSTITUIR UNA ARISTA DE CADA GRUP
     &O POR OTRA DE FORMA QUE"/16X"SE SUPERE EL TEST.")
```

```
 3004 FORMAT(//1X"MENSAJE ::SR:: A LA VISTA DEL GRAFO TRICONEXO DIBUJADO
     &, INTRODUCE LAS ARISTAS"/16X"NECESARIAS PARA TRIANGULAR EL GRAFO."
     &)
 3005 FORMAT(//1X"MENSAJE ::SR:: EL NUMERO DE ARISTAS ES"I4"Y SIN EMBARG
     &O TENDRIA QUE SER"I4)
 3006 FORMAT(//1X"MENSAJE ::SR:: SE HAN SUPRIMIDO LAS ULTIMAS ARISTAS "/
     &/16X"SI DESEAS PUEDES SUSTITUIR ALGUNAS ARISTAS")
 3007 FORMAT(//1X"MENSAJE ::SR:: GRAFO TRIANGULADO")
 3008 FORMAT(//1X"MENSAJE ::SR:: SI DESEAS TERMINAR INTRODUCE: 0"/16X"EN
     & CASO CONTRARIO INTRODUCE: 1")
      END
C
C
      SUBROUTINE REC(K1,NV,NA,JAUX,IAUX,IA)
C     ************************************
C
C SUBRUTINA QUE APLICA EL TEST DE RECTANGULARIDAD
C
      EMA IA(100,50)
      DIMENSION IAUX(15),JAUX(100)
      DO 10 I=1,15
      IAUX(I)=0
   10 CONTINUE
      DO 20 I=1,100
      JAUX(I)=0
   20 CONTINUE
      K1=0
      DO 300 I=1,NV
      K=0
      DO 50 J=1,NV
      IF(IA(I,J).EQ.0) GO TO 50
      K=K+1
      IAUX(K)=J
   50 CONTINUE
      IF(K.LT.3) GO TO 210
      DO 200 J1=1,K-2
      DO 150 J2=J1+1,K-1
      IF(IA(IAUX(J1),IAUX(J2)).LE.0) GO TO 150
      DO 100 J3=J2+1,K
      IF(IA(IAUX(J1),IAUX(J3)).LE.0) GO TO 100
      IF(IA(IAUX(J2),IAUX(J3)).LE.0) GO TO 100
      K1=K1+1
      JAUX(4*(K1-1)+1)=I
      JAUX(4*(K1-1)+2)=IAUX(J1)
      JAUX(4*(K1-1)+3)=IAUX(J2)
      JAUX(4*(K1-1)+4)=IAUX(J3)
      IA(IAUX(J1),I)=-1
      IA(IAUX(J2),I)=-1
      IA(IAUX(J3),I)=-1
      IA(I,IAUX(J1))=-1
      IA(I,IAUX(J2))=-1
      IA(I,IAUX(J3))=-1
  100 CONTINUE
  150 CONTINUE
  200 CONTINUE
  210 DO 250 J=1,K
  250 IAUX(J)=0
  300 CONTINUE
      DO 350 I=1,NV
      DO 350 J=1,NV
```

```
      IF(IA(I,J).LT.0) IA(I,J)=1
  350 CONTINUE
      RETURN
      END
C
C
      SUBROUTINE DIC(ISAL,NV,NA,KONT,JONT,XMAX,YMAX,V5,IVER,LCM,LP,XM,
     &YM,IA,IB)
C     ****************************************************************
C
C SUBRUTINA QUE TRANSFORMA LAS COORDENADAS DE LOS VERTICES INTERIORES DE
C FORMA QUE TODAS LAS ARISTAS SEAN ORTOGONALES
C
      INTEGER V5
      ENA IB(150,50),IA(100,50),XM(V5),YM(V5)
      DIMENSION IVER(8),LCM(NV),LP(V5)
C
C     PUESTA A CERO
C
      DO 10 I=1,8
      IVER(I)=0
   10 CONTINUE
      DO 20 I=1,NV
      LCM(I)=0
   20 CONTINUE
      DO 30 I=1,V5
      XM(I)=0
      YM(I)=0
      LP(I)=0
   30 CONTINUE
C
C OBTENCION DE LOS VERTICES INTERIORES DOBLEMENTE ORIENTADOS Y ORDENA-
C CION DE LOS MISMOS
C
      KT=0
      DO 100 I=5,NV
      KT1=0
      DO 50 J=1,NV
      IF(IA(I,J).EQ.0) GO TO 50
      IF(J.LE.4) KT1=KT1+1
   50 CONTINUE
      IF(KT1.NE.2) GO TO 100
      KT=KT+1
      IVER(KT)=I
      IF(KT.EQ.4) GO TO 110
  100 CONTINUE
  110 DO 150 I=1,4
      II=IVER(I)
      IF(IA(II,1).NE.0.AND.IA(II,4).NE.0) IVER(5)=II
      IF(IA(II,4).NE.0.AND.IA(II,3).NE.0) IVER(6)=II
      IF(IA(II,3).NE.0.AND.IA(II,2).NE.0) IVER(7)=II
      IF(IA(II,2).NE.0.AND.IA(II,1).NE.0) IVER(8)=II
  150 CONTINUE
      DO 200 I=1,4
      IVER(I)=IVER(4+I)
  200 CONTINUE
C
C OBTENCION DEL CIRCUITO INTERIOR MAS CERCANO AL EXTERIOR QUE ESTARA
C FORMADO POR TODOS LOS VERTICES QUE SON ADYACENTES A LOS EXTERIORES
C
      LCM(1)=IVER(1)
```

```
      KONT=1
      KT=0
      LCA=0
  210 KT=KT+1
  220 LCC=LCM(KONT)
      DO 250 I=5,NV
      IF(I.EQ.LCC.OR.I.EQ.LCA) GO TO 250
      IF(IA(I,LCC).NE.0.AND.IA(I,KT).NE.0) 230,250
  230 KONT=KONT+1
      LCM(KONT)=I
      IF(LCM(1).EQ.I) GO TO 560
      LCA=LCC
      IF(KT.EQ.4) GO TO 220
      IF(IA(I,KT+1).NE.0) 210,220
  250 CONTINUE
C
C SALIDA DE ERROR SI NO EXISTE ESTE CIRCUITO
C
      IERR=NER(ISAL,4)
      STOP 0004
  560 DO 600 I=1,KONT
      IF(LCM(I).EQ.IVER(2)) LCM(KONT+1)=I
      IF(LCM(I).EQ.IVER(3)) LCM(KONT+2)=I
      IF(LCM(I).EQ.IVER(4)) LCM(KONT+3)=I
  600 CONTINUE
      IF(LCM(KONT+1).GT.LCM(KONT+2)) 601,609
  601 J=KONT/2
      DO 605 I=1,J
      LL=LCM(I)
      LCM(I)=LCM(KONT+1-I)
      LCM(KONT+1-I)=LL
  605 CONTINUE
      DO 608 I=1,KONT
      IF(LCM(I).EQ.IVER(2)) LCM(KONT+1)=I
      IF(LCM(I).EQ.IVER(3)) LCM(KONT+2)=I
      IF(LCM(I).EQ.IVER(4)) LCM(KONT+3)=I
  608 CONTINUE
C
C SUBDIVISION DEL CIRCUITO EN CUATRO LADOS CADA UNO CORRESPONDIENTE A
C CADA UNA DE LAS ORIENTACIONES.
C DETERMINACION DEL LADO (PORCION DE CIRCUITO) QUE CONTENGA MAYOR NUMERO
C DE VERTICES INTERIORES
C ESTE LADO SERA EL INICIAL EN LA TRASFORMACION DE COORDENADAS DE LOS
C VERTICES INTERIORES DEL GRAFO.
C
  609 IMAX=LCM(KONT+1)-1
      JJ=18
      IF(LCM(KONT+2)-LCM(KONT+1).LT.IMAX) GO TO 610
      IMAX=LCM(KONT+2)-LCM(KONT+1)
      JJ=16
  610 IF(LCM(KONT+3)-LCM(KONT+2).LT.IMAX) GO TO 620
      IMAX=LCM(KONT+3)-LCM(KONT+2)
      JJ=17
  620 IF(KONT-LCM(KONT+3).GT.IMAX) JJ=15
C
      IF(JJ.NE.16) GO TO 630
      INI=LCM(KONT+2)
      INF=LCM(KONT+1)
      INO=KONT
      IFO=LCM(KONT+3)
```

```
      INC=-1
      IY=1
      IX=0
  630 IF(JJ.NE.15) GO TO 640
      INI=LCM(KONT+3)
      INF=KONT
      INO=LCM(KONT+1)
      IFO=LCM(KONT+2)
      INC=1
      IY=-1
      IX=0
  640 IF(JJ.NE.18) GO TO 660
      INI=LCM(KONT+1)
      INF=1
      INO=LCM(KONT+3)
      IFO=LCM(KONT+2)
      INC=-1
      IY=0
      IX=-1
  660 IF(JJ.NE.17) GO TO 670
      INI=LCM(KONT+2)
      INF=LCM(KONT+3)
      INO=1
      IFO=LCM(KONT+1)
      INC=1
      IY=0
      IX=1
  670 JT=JJ
C
C
C  ** LLAMADA A LA SUBRUTINA QUE OBTIENE LAS NUEVAS COORDENADAS DE LOS
C      VERTICES INTERIORES **
C
C LA LLAMADA VARIA SEGUN EL LADO INICIAL SEA NORTE O SUR (LA PRIMERA) O
C ESTE U OESTE (LA SEGUNDA)
C
  890 IF(IX.EQ.0) CALL COR(NA,V5,NV,INI,INF,INC,IY,JONT,XN,YN,LCM,IVER,
     &LP,IB,KONT,INO,IFO,IA)
      IF(IX.NE.0) CALL COR(NA,V5,NV,INI,INF,INC,IX,JONT,YN,XN,LCM,IVER,
     &LP,IB,KONT,INO,IFO,IA)
C
C  ** DIBUJO OPCIONAL DEL SUBGRAFO DE VERTICES Y ARISTAS INTERIORES
C TRASFORMADO CON LLAMADA A LA SUBRUTINA CORRESPONDIENTE **
C
      IF(ISSW(4).LT.0) CALL DIB(ISAL,JONT,XN,YN,LP,XMAX,YMAX)
      RETURN
      END
C
C
      SUBROUTINE COR(NA,V5,NV,INI,INF,INC,IXY,KONT,X,Y,LCM,IVER,LP,IB,
     &KTT,INO,IFO,IA)
C     ****************************************************************
C
C SUBRUTINA QUE OBTIENE LAS NUEVAS COORDENADAS DE LOS VERTICES INTERIO-
C RES. ESTA BASADA EN EL DESDOBLAMIENTO DE UNO DE LOS VERTICES DE CADA
C TRIANGULO OBTENIENDO SUCESIVOS RECTANGULOS
C
      INTEGER V5,VAUX(10)
      EMA IB(150,50),IA(100,50),X(V5),Y(V5)
      DIMENSION LCM(NV),IVER(8),LP(V5)
C
```

```
C INCREMENTOS EN X E Y DE LAS COORDENADAS: 2 CMS.
C
      KONT=0
      SUMX=-2.
      SUMY=0.
C
C COORDENADAS DE LOS VERTICES DEL LADO INICIAL
C
      DO 100 I=INI,INF,INC
      J=LCM(I)
      SUMX=SUMX+2.
      KONT=KONT+1
      LP(KONT)=J
      X(KONT)=SUMX
      Y(KONT)=SUMY
  100 CONTINUE
C
C COORDENADAS DE UN NUEVO LADO
C
      JONT=KONT
      LONT=0
      DO 800 IT=1,100
      SUMY=SUMY+2*IXY
      IONT=LONT+1
      LONT=KONT
      DO 325 JJ=IONT,LONT
      J=LP(JJ)
      JJJ=0
      K1=0
      DO 250 I=1,NA
      IF(IB(I,J).EQ.0) GO TO 250
      DO 200 K=1,NV
      IF(K.EQ.J.OR.IB(I,K).EQ.0) GO TO 200
      IF(K.LE.4) GO TO 250
      DO 150 KK=1,LONT
      IF(K.EQ.LP(KK)) GO TO 250
  150 CONTINUE
      K1=K1+1
      JJJ=K
      GO TO 250
  200 CONTINUE
  250 CONTINUE
      DO 300 IN=IND,IFD,INC
      IF(J.EQ.LCM(IN)) GO TO 310
  300 CONTINUE
      IF(K1-1) 325,305,310
  305 IF(JJJ.NE.LP(KONT)) JONT=JONT+1
      GO TO 320
  310 JJJ=J
  320 KONT=KONT+1
      LP(KONT)=JJJ
      X(KONT)=X(JJ)
      Y(KONT)=SUMY
  325 CONTINUE
  326 J1=0
      J2=0
      JJJ=0
      K1=1
C
C  ** LLAMADA A LA SUBRUTINA-TEST PARA TERMINAR LA TRASFORMACION **
```

```
C                                                                       COR-421
      CALL BAL(NV,VS,LONT,KONT,LP,IND,IFD,INC,LCN,KRR)                  COR-422
      IF(KRR.EQ.0) GO TO 367                                            COR-423
      DO 366 I=LONT+1,KONT-1                                            COR-424
      IF(LP(I).EQ.LP(I+1)) GO TO 327                                    COR-425
      IF(JJJ.EQ.0) GO TO 366                                            COR-426
      J2=I                                                              COR-427
      GO TO 328                                                         COR-428
  327 K1=K1+1                                                           COR-429
      IF(JJJ.EQ.0) J1=I                                                 COR-430
      JJJ=LP(I)                                                         COR-431
      IF(I.NE.KONT-1) GO TO 366                                         COR-432
      J2=KONT                                                           COR-433
  328 KKK=0                                                             COR-434
      DO 330 IN=IND,IFD,INC                                             COR-435
      IF(JJJ.EQ.LCN(IN)) GO TO 365                                      COR-436
  330 CONTINUE                                                          COR-437
      MT=0                                                              COR-438
      KT=KONT-LONT                                                      COR-439
      DO 332 IN=LONT+1,KONT                                             COR-440
      DO 331 JN=IONT,LONT                                               COR-441
      IF(K(JN).NE.K(IN)) GO TO 331                                      COR-442
      IF(LP(IN).EQ.LP(JN)) MT=MT+1                                      COR-443
      GO TO 332                                                         COR-444
  331 CONTINUE                                                          COR-445
  332 CONTINUE                                                          COR-446
      IF(KT.NE.MT) GO TO 365                                            COR-447
      DO 340 K=5,NV                                                     COR-448
      DO 335 L=1,KONT                                                   COR-449
      IF(K.EQ.LP(L)) GO TO 340                                          COR-450
  335 CONTINUE                                                          COR-451
      IF(IA(K,JJJ).EQ.0) GO TO 340                                      COR-452
      KKK=KKK+1                                                         COR-453
      VAUX(KKK)=K                                                       COR-454
  340 CONTINUE                                                          COR-455
      IF(KKK.EQ.0) GO TO 365                                            COR-456
      IF(KKK.GT.K1) GO TO 353                                           COR-457
      IF(J1.NE.LONT+1.AND.J2.NE.KONT) GO TO 354                         COR-458
      NN=0                                                              COR-459
      IF(J1.NE.LONT+1) GO TO 352                                        COR-460
  342 DO 350 K=1,KKK                                                    COR-461
      KK=VAUX(K)                                                        COR-462
      DO 345 L=1,4                                                      COR-463
      IF(IA(L,JJJ).NE.0.AND.IA(L,KK).NE.0) GO TO 351                    COR-464
  345 CONTINUE                                                          COR-465
  350 CONTINUE                                                          COR-466
  351 IF(NN.NE.0) GO TO 353                                             COR-467
      LP(J1)=KK                                                         COR-468
      JONT=JONT+1                                                       COR-469
      J1=J1+1                                                           COR-470
  352 NN=NN+1                                                           COR-471
      IF(J2.NE.KONT) 354,342                                            COR-472
  353 LP(J2)=KK                                                         COR-473
      JONT=JONT+1                                                       COR-474
      J2=J2-1                                                           COR-475
  354 JJ=LP(J1-1)                                                       COR-476
      NN=0                                                              COR-477
      I1=0                                                              COR-478
      I2=0                                                              COR-479
  355 DO 356 K=1,KKK                                                    COR-480
      KK=VAUX(K)                                                        COR-481
```

```
      IF(IA(KK,JJ).NE.0) GO TO 357
  356 CONTINUE
      IF(NN.EQ.0) 358,360
  357 IF(NN.NE.0) GO TO 359
      I1=KK
  358 NN=NN+1
      IF(J1.EQ.J2) GO TO 360
      JJ=LP(J2+1)
      GO TO 355
  359 I2=KK
  360 IF(I1.EQ.0.AND.I2.EQ.0) GO TO 363
      IF(I1.EQ.0) GO TO 361
      LP(J1)=I1
      JONT=JONT+1
      J1=J1+1
  361 IF(I2.EQ.0) GO TO 362
      LP(J2)=I2
      JONT=JONT+1
      J2=J2-1
  362 IF(J1.LE.J2) GO TO 354
  363 DO 364 K=1,KKK
      VAJK(K)=0
  364 CONTINUE
  365 JJJ=0
      J1=0
      J2=0
      K1=1
  366 CONTINUE
C
C OPERACION QUE SE REALIZA SI DOS VERTICES DEL NUEVO LADO QUE DEBIERAN
C SER ADYACENTES NO LO SON.
C
  367 IS=LONT+1
  368 MT=0
      NT=0
      DO 370 JJ=IS+1,KONT-1
      LT=LP(JJ)
      IF(MT.EQ.0.AND.LP(JJ-1).NE.LP(JJ)) MT=LP(JJ-1)
      IF(NT.EQ.0.AND.LP(JJ+1).NE.LP(JJ)) NT=LP(JJ+1)
      IF(MT.NE.0.AND.NT.NE.0) GO TO 375
  370 CONTINUE
      GO TO 396
  375 DO 380 I=1,NA
      IF(IB(I,MT).EQ.0.OR.IB(I,NT).EQ.0) 380,385
  380 CONTINUE
      IS=JJ
      GO TO 368
  385 KT=0
      DO 390 JJ=LONT+1,KONT
      IF(LP(JJ).EQ.LT) KT=KT+1
      IF(KT.EQ.0) GO TO 390
      IF(LP(JJ).NE.LT) GO TO 386
      LP(JJ)=0
      X(JJ)=0.
      Y(JJ)=0.
      GO TO 390
  386 LP(JJ-KT)=LP(JJ)
      X(JJ-KT)=X(JJ)
      Y(JJ-KT)=Y(JJ)
  390 CONTINUE
```

```
      DO 393 JJ=KONT-KT+1,KONT
      X(JJ)=0.
      Y(JJ)=0.
  393 LP(JJ)=0
      KONT=KONT-KT
      GO TO 367
C
C  ** LLAMADA A LA SUBRUTINA-TEST PARA TERMINAR LA TRANSFORMACION **
C
  396 CALL SAL(NV,VS,LONT,KONT,LP,IND,IFO,INC,LCN,KRR)
      IF(KRR.EQ.0) GO TO 810
      NT=0
      KT=KONT-LONT
      DO 405 I=LONT+1,KONT
      DO 400 J=IONT,LONT
      IF(X(J).NE.X(I)) GO TO 400
      IF(LP(I).EQ.LP(J)) NT=NT+1
      GO TO 405
  400 CONTINUE
  405 CONTINUE
      IF(KT.NE.NT) GO TO 800
C
C OPERACION QUE SE REALIZA SI SE OBTIENE UN NUEVO LADO IDENTICO A AQUEL
C DE QUIEN PROCEDE.
C
      I=LONT+1
      GO TO 410
  407 I=KONT
      GO TO 410
  408 DO 600 I=LONT+2,KONT-1
  410 NT=0
      DO 415 J=LONT+1,KONT
      IF(LP(I).EQ.LP(J)) NT=NT+1
  415 CONTINUE
      LPP=LP(I)
      DO 420 IN=IND,IFO,INC
      LCC=LCN(IN)
      IF(LCC.EQ.LPP.AND.I.EQ.LONT+1) GO TO 407
      IF(LCC.EQ.LPP.AND.I.EQ.KONT) GO TO 408
      IF(LCC.EQ.LPP) GO TO 600
  420 CONTINUE
      MT=0
      DO 450 K=5,NV
      DO 430 J=1,KONT
      IF(K.EQ.LP(J)) GO TO 450
  430 CONTINUE
      IF(IA(K,LPP).NE.0) MT=MT+1
  450 CONTINUE
      IF(NT.LE.MT) GO TO 326
      JI=1
      JF=KONT
      JT=0
  460 DO 500 J=JI,JF
      IF(X(I+JT).NE.X(J)) GO TO 500
      JT=JT+1
      DO 470 K=JF+1,J+1,-1
      LP(K)=LP(K-1)
      X(K)=X(K-1)
      Y(K)=Y(K-1)
  470 CONTINUE
      DO 480 K=J+1,JF+1
```

```
      IF(Y(K).NE.Y(J)) GO TO 510
      X(K)=X(K)+2.
  480 CONTINUE
      GO TO 520
  500 CONTINUE
      GO TO 520
  510 JI=K
      JF=JF+1
      GO TO 460
  520 LONT=LONT+JT-1
      KONT=KONT+JT
      IONT=IONT+JT-2
      GO TO 410
  600 CONTINUE
  800 CONTINUE
C
C CORRECCION DE LAS COORDENADAS DE TODOS LOS VERTICES INTERIORES PARA
C QUE TODAS LAS COORDENADAS X E Y SEAN MAYORES O IGUALES A CERO.
C
  810 RINCX=0.
      RINCY=0.
      DO 850 I=1,KONT
      IF(X(I).LT.RINCX) RINCX=X(I)
      IF(Y(I).LT.RINCY) RINCY=Y(I)
  850 CONTINUE
      DO 900 I=1,KONT
      X(I)=X(I)-RINCX
      Y(I)=Y(I)-RINCY
  900 CONTINUE
      RETURN
      END
C
C
      SUBROUTINE SAL(NV,V5,LONT,KONT,LP,INO,IFO,INC,LCN,KRR)
C     *****************************************************
C
C SUBRUTINA DE TEST PARA TERMINAR LA TRANSFORMACION
C
      INTEGER V5
      DIMENSION LCN(NV),LP(V5)
      KRR=0
      DO 100 I=LONT+1,KONT
      DO 50 J=INO,IFO,INC
      IF(LP(I).EQ.LCN(J)) GO TO 100
   50 CONTINUE
      KRR=1
      GO TO 110
  100 CONTINUE
  110 RETURN
      END
C
C
      SUBROUTINE DIB(ISAL,KONT,X,Y,LP,XMAX,YMAX)
C     ******************************************
C
C DIBUJA EL NUEVO SUBGRAFO DE VERTICES INTERIORES Y ARISTAS INTERIORES
C DE FORMA QUE TODAS LAS ARISTAS SEAN ORTOGONALES ENTRE SI.
C
      ENA X(KONT),Y(KONT)
      DIMENSION LP(KONT)
```

```
      CALL PLTLU(10)
C
C DETERMINACION DE LA ABCISA Y ORDENADA MAXIMA Y NUMERACION DE LOS VER-
C TICES INTERIORES.
C
      XMAX=0.
      YMAX=0.
      DO 50 I=1,KONT
      IF(X(I).GT.XMAX) XMAX=X(I)
      IF(Y(I).GT.YMAX) YMAX=Y(I)
   50 CONTINUE
      WRITE(ISAL,3001) XMAX,YMAX
      PAUSE 0001
      CALL PLOT(0.,0.,-3)
      DO 55 I=1,KONT
      RI=LP(I)
      CALL NUMB(X(I)-0.3,Y(I)-0.2,0.15,RI,0.,-1)
   55 CONTINUE
      XX=0.
      YY=0.
      K1=0
C
C TRAZADO DE LAS ARISTAS: PRIMERO EN UNA DIRECCION Y LUEGO EN LA ORTOGO-
C NAL.
C
   60 DO 100 I=1,KONT
      IF(XX.NE.X(I).OR.YY.NE.Y(I)) 100,110
  100 CONTINUE
      GO TO 115
  110 IF(K1.EQ.0) CALL PLOT(XX,YY,3)
      IF(K1.NE.0) CALL PLOT(XX,YY,2)
      K1=K1+1
  115 IF(XX.EQ.XMAX) GO TO 120
      XX=XX+2.
      GO TO 60
  120 IF(YY.EQ.YMAX) GO TO 125
      YY=YY+2.
      XX=0.
      K1=0
      GO TO 60
  125 XX=0.
      YY=0.
      K1=0
  130 DO 150 I=1,KONT
      IF(XX.NE.X(I).OR.YY.NE.Y(I)) 150,160
  150 CONTINUE
      GO TO 165
  160 IF(K1.EQ.0) CALL PLOT(XX,YY,3)
      IF(K1.NE.0) CALL PLOT(XX,YY,2)
      K1=K1+1
  165 IF(YY.EQ.YMAX) GO TO 170
      YY=YY+2.
      GO TO 130
  170 IF(XX.EQ.XMAX) GO TO 180
      XX=XX+2.
      YY=0.
      K1=0
      GO TO 130
  180 CALL PLOT(0.,0.,3)
      RETURN
C
```

```
C FORMATOS                                                              DIB-724
C                                                                       DIB-725
 3001 FORMAT(//1X"MENSAJE ::SR:: PON EL ORIGEN DE LA PLUMA DONDE DESEES" DIB-726
     &//16X"LA DIMENSION DEL CONTORNO ES"F7.2" X"F7.2)                  DIB-727
      END                                                               DIB-728
C                                                                       DUA-729
C                                                                       DUA-730
      SUBROUTINE DUA(ISAL,ISAUX,JONT,KONT,LCN,LP,X,Y,XMAX,YMAX)         DUA-731
C     **********************************************************        DUA-732
C                                                                       DUA-733
C OBTIENE EL GRAFO DUAL DEL INICIAL QUE SERA LA REPRESENTACION EN PLANTA DUA-734
C BUSCADA                                                               DUA-735
C                                                                       DUA-736
      ENA X(JONT),Y(JONT)                                               DUA-737
      DIMENSION LCN(KONT),LP(JONT)                                      DUA-738
      CALL PLTLU(10)                                                    DUA-739
      XMAX=0.                                                           DUA-740
      YMAX=0.                                                           DUA-741
      DO 5 I=1,JONT                                                     DUA-742
      IF(X(I).GT.XMAX) XMAX=X(I)                                        DUA-743
      IF(Y(I).GT.YMAX) YMAX=Y(I)                                        DUA-744
    5 CONTINUE                                                          DUA-745
      XX=XMAX+2.                                                        DUA-746
      YY=YMAX+2.                                                        DUA-747
      WRITE(ISAL,3001) XX,YY                                            DUA-748
      PAUSE 0002                                                        DUA-749
      CALL PLOT(1.,1.,-3)                                               DUA-750
      IF(ISSW(5).LT.0) 10,20                                            DUA-751
   10 WRITE(ISAL,5001)                                                  DUA-752
      WRITE(ISAL,5002)                                                  DUA-753
C                                                                       DUA-754
C OBTENCION DE LOS CUATRO PARAMETROS CORRESPONDIENTES A CADA VERTICE.   DUA-755
C                                                                       DUA-756
   20 DO 150 I=1,JONT                                                   DUA-757
      II=LP(I)                                                          DUA-758
      Z1=0.                                                             DUA-759
      Z2=0.                                                             DUA-760
      Z3=0.                                                             DUA-761
      Z4=0.                                                             DUA-762
      DO 55 JJ=1,KONT                                                   DUA-763
      J=LCN(JJ)                                                         DUA-764
      IF(II.NE.J) GO TO 55                                              DUA-765
      IF(X(I).EQ.0.) Z2=-1.                                             DUA-766
      IF(X(I).EQ.XMAX) Z4=XMAX+1.                                       DUA-767
      IF(Y(I).EQ.YMAX) Z3=YMAX+1.                                       DUA-768
      IF(Y(I).EQ.0.) Z1=-1.                                             DUA-769
      GO TO 60                                                          DUA-770
   55 CONTINUE                                                          DUA-771
   60 D1=100.                                                           DUA-772
      D2=100.                                                           DUA-773
      D3=100.                                                           DUA-774
      D4=100.                                                           DUA-775
      J1=0                                                              DUA-776
      J2=0                                                              DUA-777
      J3=0                                                              DUA-778
      J4=0                                                              DUA-779
      DO 100 J=1,JONT                                                   DUA-780
      JJ=LP(J)                                                          DUA-781
      IF(I.EQ.J) GO TO 100                                              DUA-782
      IF(X(J).NE.X(I)) GO TO 80                                         DUA-783
```

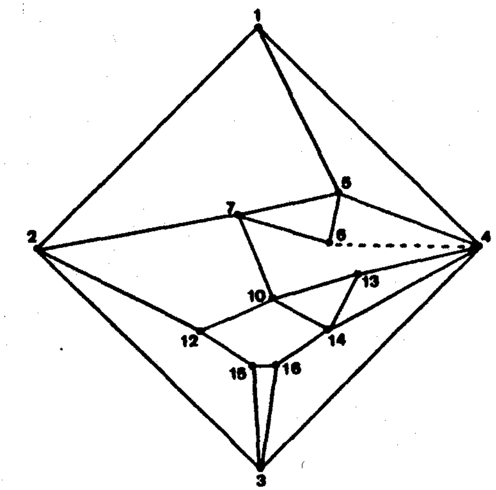
```
      IF(Y(J).GT.Y(I)) GO TO 70
      IF(Y(I)-Y(J).GT.D1) GO TO 100
      J1=J
      D1=Y(I)-Y(J)
      GO TO 100
   70 IF(Y(J)-Y(I).GT.D3) GO TO 100
      J3=J
      D3=Y(J)-Y(I)
      GO TO 100
   80 IF(Y(J).NE.Y(I)) GO TO 100
      IF(X(J).GT.X(I)) GO TO 90
      IF(X(I)-X(J).GT.D2) GO TO 100
      J2=J
      D2=X(I)-X(J)
      GO TO 100
   90 IF(X(J)-X(I).GT.D4) GO TO 100
      J4=J
      D4=X(J)-X(I)
  100 CONTINUE
      IF(J1.NE.0) Z1=(Y(I)+Y(J1))/2.
      IF(J3.NE.0) Z3=(Y(I)+Y(J3))/2.
      IF(J2.NE.0) Z2=(X(I)+X(J2))/2.
      IF(J4.NE.0) Z4=(X(I)+X(J4))/2.
      IF(Z1.EQ.0) Z1=Y(I)-1.
      IF(Z3.EQ.0) Z3=Y(I)+1.
      IF(Z2.EQ.0) Z2=X(I)-1.
      IF(Z4.EQ.0) Z4=X(I)+1.
      IF(ISSW(5).LT.0) WRITE(ISAL,5003) II,Z1,Z2,Z3,Z4
C
C NUMERACION Y TRAZADO DEL GRAFO DUAL
C
      RI=II
      X1=(Z2+Z4)/2.
      Y1=(Z1+Z3)/2.
      CALL NUMB(X1,Y1,0.15,RI,0.,-1)
      CALL PLOT(Z4,Z1,3)
      IF(LP(J1).NE.II) CALL PLOT(Z2,Z1,2)
      IF(LP(J1).EQ.II) CALL PLOT(Z2,Z1,3)
      IF(LP(J2).NE.II) CALL PLOT(Z2,Z3,2)
      IF(LP(J2).EQ.II) CALL PLOT(Z2,Z3,3)
      IF(LP(J3).NE.II) CALL PLOT(Z4,Z3,2)
      IF(LP(J3).EQ.II) CALL PLOT(Z4,Z3,3)
      IF(LP(J4).NE.II) CALL PLOT(Z4,Z1,2)
      IF(LP(J4).EQ.II) CALL PLOT(Z4,Z1,3)
  150 CONTINUE
      RETURN
C
C FORMATOS
C
 3001 FORMAT(//1X"MENSAJE ::3R:: PON EL ORIGEN DE LA PLUMA DONDE DESEES"
     &//16X"LA DIMENSION DEL CONTORNO ES"F7.2" X"F7.2)
 5001 FORMAT(//1X"PARAMETROS CORRESPONDIENTES A CADA VERTICE DEL GRAFO D
     &UAL:"/)
 5002 FORMAT(/2X,"V     Z1     Z2     Z3     Z4"/2X"-     --     --     --")
 5003 FORMAT(I3,4F5.2)
      END
$
```

A.II.1.4. Ejemplos

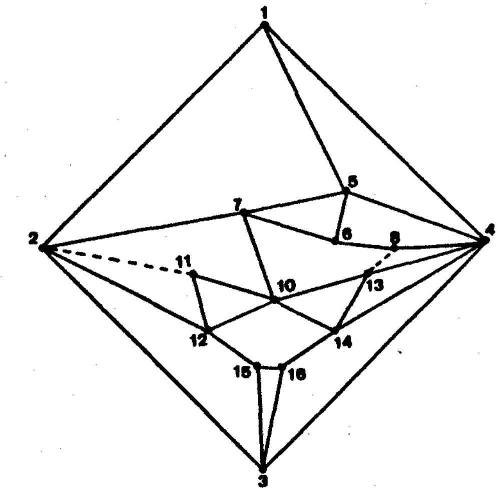
```
MENSAJE ::ED:: TITULA EL PROBLEMA

      EJEMPLO 1 (ADIMENSIONAL)
********************************************************************************


MENSAJE ::ED:: ASIGNE UN NOMBRE A CADA UNO DE LOS ESPACIOS

               AL FINAL DE LA LISTA INTRODUCE:0

ESP  NOMBRE
---  ------
  5  VESTIBULO
  6  COCINA
  7  ESTAR-COMEDOR
  8  LAVADERO
  9  DESPENSA
 10  PASILLO
 11  CUARTO DE BANO
 12  DORMITORIO DOBLE
 13  DORMITORIO SENCILLO
 14  DORMITORIO DOBLE
 15  ARMARIO
 16  ARMARIO

MENSAJE ::ED:: MATRIZ DE INCIDENCIA

               LA NUMERACION DE LOS VERTICES  Y ARISTAS SE HARA A PARTIR DEL 5.
               LOS 4 PRIMEROS VERTICES Y ARISTAS CORRESPONDEN A LOS PUNTOS
               CARDINALES Y SUS RELACIONES. ESTAN INTRODUCIDOS AUTOMATICAMENTE.
               AL FINAL DE LA LISTA INTRODUCE: 0


  A  V  V
  -  -  -
  5  5  1
  6  5  4
  7  5  6
  8  5  7
  9  6  8
 10  6  9
 11  7  2
 12  7 10
 13  8  4
 14 10 11
 15 10 12
 16 10 13
 17 10 14
 18 11 12
 19 12  2
 20 12 15
 21 13  4
 22 13 14
 23 14  4
 24 14 16
 25 15 16
 26 15  3
```

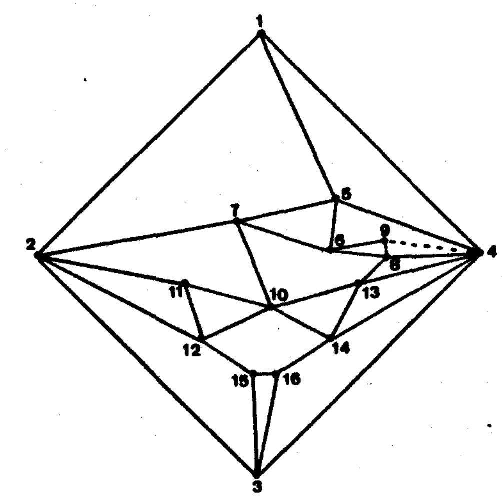
```
27 16  3
28  6  7

MENSAJE !!SR!! EL GRAFO ES CONEXO

MENSAJE !!SR!! EL GRAFO NO ES BICONEXO

LAS COMPONENTES BICONEXAS SON:

COMPONENTE:  1
--------------
ARISTAS:  10

COMPONENTE:  2
--------------
ARISTAS:   7   4   6   3  12  15  21  22  16  17  14  18   1   2  11  19  20
          25  26  27  24  23  13   9  28   8   5

LOS VERTICES DE SEPARACION DE ESAS COMPONENTES SON:

COMPONENTES: 1- 2 .... VERT. DE SEPAR:  6

                                                        COMPONENTE BICONEXA:  1
-------------------------------------------------------------------------------

ARISTAS:  10

MENSAJE !!SR!! NO SE DIBUJA POR SER ELEMENTAL

                                                        COMPONENTE BICONEXA:  2
-------------------------------------------------------------------------------

ARISTAS:   7   4   6   3  12  15  21  22  16  17  14  18   1   2  11  19  20
          25  26  27  24  23  13   9  28   8   5

MENSAJE !!SR!! LA COMPONENTE BICONEXA ES PLANAR

MENSAJE !!SR!! LA COMPONENTE BICONEXA NO ES TRICONEXA

LAS COMPONENTES TRICONEXAS Y TRIANGULOS (DONDE LAS ARISTAS VIENEN EXPRESADAS
POR SUS VERTICES EXTREMOS) SON:

COMPONENTE: 1
ARISTAS:   1- 2   2- 3   3- 4   4- 1   4- 5   5- 6   6- 4   6- 7   7- 5   7- 2
           7-10  10-12  12-15  15- 3  15-16  16-14  14-10  14- 4  14-13  13-10
          13- 4  16- 3  12- 2   5- 1

MENSAJE !!SR!! ARISTA VIRTUAL EN LA COMPONENTE 1:   6- 4

MENSAJE !!SR!! LAS ARISTAS VIRTUALES CITADAS HAN PASADO A SER REALES.
```

```
                  POSTERIORMENTE,SI SE DESEA,PUEDEN SER CAMBIADAS POR OTRAS.

MENSAJE !!SR!! ADEMAS DE LAS ARISTAS PERTENECIENTES A ALGUNA COMPONENTE
                  HAY OTRAS QUE NO PERTENECEN A NINGUNA DE ELLAS.

LAS ARISTAS QUE RELACIONAN UNAS COMPONENTES CON OTRAS Y LAS QUE NO ENLAZAN A
DOS COMPONENTES CUALESQUIERA SON:

ARISTA:  10-11  (NO LIGA A DOS COMPONENTES)
ARISTA:  11-12  (NO LIGA A DOS COMPONENTES)
ARISTA:   6- 8  (NO LIGA A DOS COMPONENTES)
ARISTA:   8- 4  (NO LIGA A DOS COMPONENTES)

------------------------------
COMPONENTE: 1 (TRICONEXA)
------------------------------

ARISTAS:   1- 2   2- 3   3- 4   4- 1   4- 5   5- 6   6- 4   6- 7   7- 5   7- 2
           7-10  10-12  12-15  15- 3  15-16  16-14  14-10  14- 4  14-13  13-10
          13- 4  16- 3  12- 2   5- 1

MENSAJE !!SR!! SE DIBUJA EN EL PLOTTER

MENSAJE !!SR!! SE HAN DE INTRODUCIR NUEVAS ARISTAS PARA CONVERTIR EN TRICONEXA
                  LA COMPONENTE BICONEXA.

                  LA ADICION SE HARA A LA VISTA DEL SUBGRAFO TRAZADO.
                  SE PUEDE REALIZAR ESCALONADAMENTE,INTRODUCIENDO SOLO PARTE DE
                  ELLAS, ANADIENDO EL RESTO A PARTIR DE UN NUEVO TRAZADO.

MENSAJE !!SR!! EL NUMERO DE ARISTAS ACTUALES ES:  29

MENSAJE !!ED!! INTRODUCE LAS NUEVAS ARISTAS

                  AL FINAL DE LA LISTA INTRODUCE: 0

  A  V  V
  -  -  -
 30  2 11

MENSAJE !!SR!! SI SE DESEA SE PUEDE SUSTITUIR ARISTAS

                  LOS DATOS SE DARAN EN EL SIGUIENTE ORDEN:
                  - NUMERO DE LA ARISTA
                  - VERTICE PRIMITIVO
                  - VERTICE PRIMITIVO
                  - VERTICE ACTUAL
                  - VERTICE ACTUAL
                  AL FINAL DE LA LISTA INTRODUCE: 0

  A  VP  VP  VA  VA
  -  --  --  --  --
 29   6   4   8  13
```

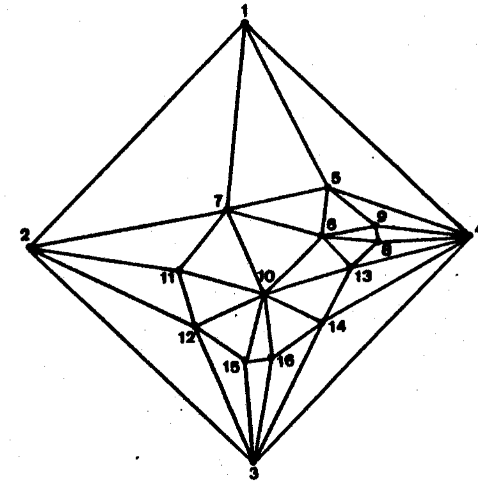


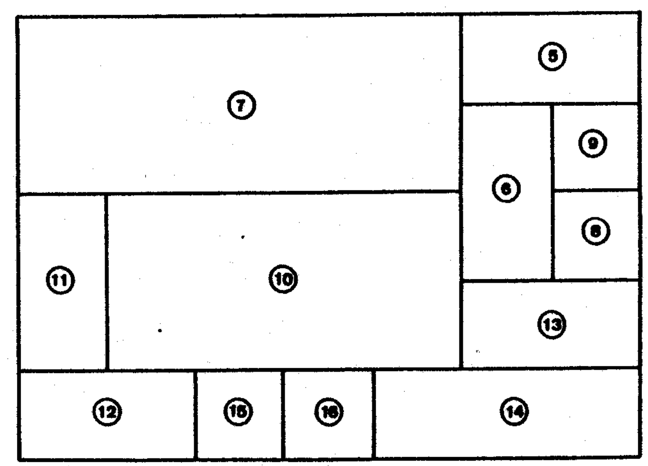
```
MENSAJE ::SR:: LA COMPONENTE BICONEXA ES PLANAR

MENSAJE ::SR:: LA COMPONENTE BICONEXA ES ADEMAS TRICONEXA

MENSAJE ::SR:: SE DIBUJA EN EL PLOTTER

MENSAJE ::SR:: SE HAN DE ANADIR NUEVAS ARISTAS PARA QUE EL GRAFO SEA TRICONEXO

               ESTAS ARISTAS ENLAZARAN A LAS DISTINTAS COMPONENTES BICONEXAS
               ENTRE SI.

MENSAJE ::SR:: EL NUMERO DE ARISTAS ACTUALES ES:  30

MENSAJE ::ED:: INTRODUCE LAS NUEVAS ARISTAS

               AL FINAL DE LA LISTA INTRODUCE: 0

  A  V  V
  -  -  -
 31  9  4
 32  9  8

MENSAJE ::SR:: SI SE DESEA SE PUEDE SUSTITUIR ARISTAS

               LOS DATOS SE DARAN EN EL SIGUIENTE ORDEN:
               - NUMERO DE LA ARISTA
               - VERTICE PRIMITIVO
               - VERTICE PRIMITIVO
               - VERTICE ACTUAL
               - VERTICE ACTUAL
               AL FINAL DE LA LISTA INTRODUCE: 0

  A  VP  VP  VA  VA
  -  --  --  --  --

MENSAJE ::SR:: EL GRAFO ES BICONEXO

MENSAJE ::SR:: EL GRAFO ES PLANAR

MENSAJE ::SR:: EL GRAFO ES TRICONEXO

MENSAJE ::SR:: SE DIBUJA EN EL PLOTTER

MENSAJE ::SR:: A LA VISTA DEL GRAFO TRICONEXO DIBUJADO, INTRODUCE LAS ARISTAS
               NECESARIAS PARA TRIANGULAR EL GRAFO.

MENSAJE ::SR:: EL NUMERO DE ARISTAS ACTUALES ES:  32

MENSAJE ::ED:: INTRODUCE LAS NUEVAS ARISTAS
```





```
                  AL FINAL DE LA LISTA INTRODUCE: 0

  A   V   V
  -   -   -
 33   5   9
 34   6  10
 35   6  13
 36   7   1
 37   7  11
 38  10  15
 39  10  16
 40  12   3
 41  14   3

MENSAJE !!SR!! GRAFO TRIANGULADO

MENSAJE !!SR!! SE DIBUJA EN EL PLOTTER

MENSAJE !!SR!! POR EL ORIGEN DE LA PLUMA DONDE DESEES

               LA DIMENSION DEL CONTORNO ES  14.00 X  10.00

MENSAJE !!SR!! SI DESEAS TERMINAR INTRODUCE: 0
               EN CASO CONTRARIO INTRODUCE: 1
```

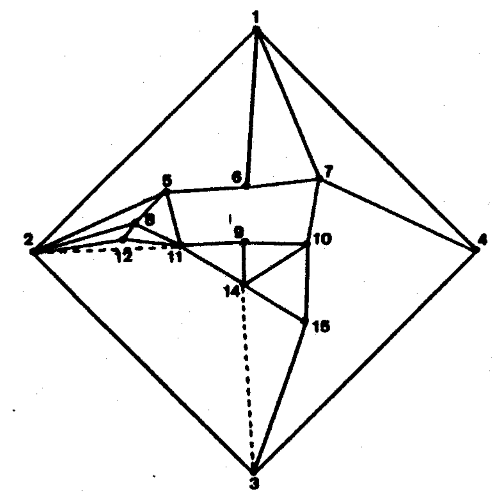




```
MENSAJE ::ED:: TITULA EL PROBLEMA

EJEMPLO 2 (ADIMENSIONAL)
********************************************************************************


MENSAJE ::ED:: ASIGNE UN NOMBRE A CADA UNO DE LOS ESPACIOS

               AL FINAL DE LA LISTA INTRODUCE:0

ESP  NOMBRE
---  ------
  5  DORMITORIO PRINCIPAL
  6  ESTUDIO
  7  COMEDOR
  8  ARMARIO
  9  CUARTO DE BAÑO
 10  COCINA
 11  PASILLO
 12  DORMITORIO SENCILLO
 13  ARMARIO
 14  VESTIBULO
 15  ESTAR
 16  DORMITORIO DOBLE
 17  ENTRADA

MENSAJE ::ED:: MATRIZ DE INCIDENCIA

               LA NUMERACION DE LOS VERTICES  Y ARISTAS SE HARA A PARTIR DEL 5.
               LOS 4 PRIMEROS VERTICES Y ARISTAS CORRESPONDEN A LOS PUNTOS
               CARDINALES Y SUS RELACIONES. ESTAN INTRODUCIDOS AUTOMATICAMENTE.
               AL FINAL DE LA LISTA INTRODUCE: 0

  A  V  V
  -  -  -
  5  2  5
  6  2 12
  7  2 16
  8  3 17
  9  3 15
 10  4  7
 11  1  7
 12  5  6
 13  6  7
 14  5 11
 15  7 10
 16  5  8
 17 11 12
 18  9 11
 19  9 10
 20 11 16
 21 11 14
 22 10 14
 23 13 16
 24 14 15
 25 10 15
```

```
26 14 17
27  1  6
28  9 14
29  8 12
30  2  8

MENSAJE ::SR:: EL GRAFO ES CONEXO


MENSAJE ::SR:: EL GRAFO NO ES BICONEXO

LAS COMPONENTES BICONEXAS SON:

COMPONENTE:  1
--------------
ARISTAS:  23

COMPONENTE:  2
--------------
ARISTAS:  11  22   1   4  10   3  26   8   9   2   5  12  14  16   6  17  29
          30   7  20  21  18  19  28  24  25  15  13  27


LOS VERTICES DE SEPARACION DE ESAS COMPONENTES SON:

COMPONENTES: 1- 2 .... VERT. DE SEPAR: 16

                                                     COMPONENTE BICONEXA:  1
-----------------------------------------------------------------------------

ARISTAS:  23

MENSAJE ::SR:: NO SE DIBUJA POR SER ELEMENTAL

                                                     COMPONENTE BICONEXA:  2
-----------------------------------------------------------------------------

ARISTAS:  11  22   1   4  10   3  26   8   9   2   5  12  14  16   6  17  29
          30   7  20  21  18  19  28  24  25  15  13  27


MENSAJE ::SR:: LA COMPONENTE BICONEXA ES PLANAR


MENSAJE ::SR:: LA COMPONENTE BICONEXA NO ES TRICONEXA

LAS COMPONENTES TRICONEXAS Y TRIANGULOS (DONDE LAS ARISTAS VIENEN EXPRESADAS
POR SUS VERTICES EXTREMOS) SON:


COMPONENTE: 1
ARISTAS:    1- 2   2- 3   3- 4   4- 1   4- 7   7- 1   7- 6   6- 5   5- 2   5- 8
            8- 2   8-12  12- 2  12-11  11- 5  11- 9   9-10  10- 7  10-14  14- 9
           14-11  14- 3  14-15  15-10  15- 3  11- 2   6- 1


MENSAJE ::SR:: ARISTA VIRTUAL EN LA COMPONENTE 1:  14- 3
```

```
MENSAJE !!SR!! ARISTA VIRTUAL EN LA COMPONENTE 1:  11- 2

MENSAJE !!SR!! LAS ARISTAS VIRTUALES CITADAS HAN PASADO A SER REALES.
               POSTERIORMENTE,SI SE DESEA,PUEDEN SER CAMBIADAS POR OTRAS.

MENSAJE !!SR!! ADEMAS DE LAS ARISTAS PERTENECIENTES A ALGUNA COMPONENTE
               HAY OTRAS QUE NO PERTENECEN A NINGUNA DE ELLAS.

LAS ARISTAS QUE RELACIONAN UNAS COMPONENTES CON OTRAS Y LAS QUE NO ENLAZAN A
DOS COMPONENTES CUALESQUIERA SON:

ARISTA:  11-16  (NO LIGA A DOS COMPONENTES)
ARISTA:  16- 2  (NO LIGA A DOS COMPONENTES)
ARISTA:  14-17  (NO LIGA A DOS COMPONENTES)
ARISTA:  17- 3  (NO LIGA A DOS COMPONENTES)

--------------------------
COMPONENTE: 1 (TRICONEXA)
--------------------------

ARISTAS:   1- 2   2- 3   3- 4   4- 1   4- 7   7- 1   7- 6   6- 5   5- 2   5- 8
           8- 2   8-12  12- 2  12-11  11- 5  11- 9   9-10  10- 7  10-14  14- 9
          14-11  14- 3  14-15  15-10  15- 3  11- 2   6- 1

MENSAJE !!SR!! SE DIBUJA EN EL PLOTTER

MENSAJE !!SR!! SE HAN DE INTRODUCIR NUEVAS ARISTAS PARA CONVERTIR EN TRICONEXA
               LA COMPONENTE BICONEXA.

               LA ADICION SE HARA A LA VISTA DEL SUBGRAFO TRAZADO.
               SE PUEDE REALIZAR ESCALONADAMENTE,INTRODUCIENDO SOLO PARTE DE
               ELLAS, ANADIENDO EL RESTO A PARTIR DE UN NUEVO TRAZADO.

MENSAJE !!SR!! EL NUMERO DE ARISTAS ACTUALES ES:  32

MENSAJE !!ED!! INTRODUCE LAS NUEVAS ARISTAS

               AL FINAL DE LA LISTA INTRODUCE: 0

  A  V  V
  -  -  -

MENSAJE !!SR!! SI SE DESEA SE PUEDE SUSTITUIR ARISTAS

               LOS DATOS SE DARAN EN EL SIGUIENTE ORDEN:
               - NUMERO DE LA ARISTA
               - VERTICE PRIMITIVO
               - VERTICE PRIMITIVO
               - VERTICE ACTUAL
               - VERTICE ACTUAL
               AL FINAL DE LA LISTA INTRODUCE: 0
```



```
 A  VP  VP  VA  VA
 -  --  --  --  --
31  14   3  16  17
32  11   2  14  16
```

MENSAJE !!SR!! LA COMPONENTE BICONEXA ES PLANAR

MENSAJE !!SR!! LA COMPONENTE BICONEXA ES ADEMAS TRICONEXA

MENSAJE !!SR!! SE DIBUJA EN EL PLOTTER

MENSAJE !!SR!! SE HAN DE AÑADIR NUEVAS ARISTAS PARA QUE EL GRAFO SEA TRICONEXO

ESTAS ARISTAS ENLAZARAN A LAS DISTINTAS COMPONENTES BICONEXAS ENTRE SI.

MENSAJE !!SR!! EL NUMERO DE ARISTAS ACTUALES ES: 32

MENSAJE !!ED!! INTRODUCE LAS NUEVAS ARISTAS

AL FINAL DE LA LISTA INTRODUCE: 0

```
 A  V  V
 -  -  -
33  2 12
34 12 13
```

MENSAJE !!SR!! SI SE DESEA SE PUEDE SUSTITUIR ARISTAS

LOS DATOS SE DARAN EN EL SIGUIENTE ORDEN:
- NUMERO DE LA ARISTA
- VERTICE PRIMITIVO
- VERTICE PRIMITIVO
- VERTICE ACTUAL
- VERTICE ACTUAL
AL FINAL DE LA LISTA INTRODUCE: 0

```
A  VP  VP  VA  VA
-  --  --  --  --
```

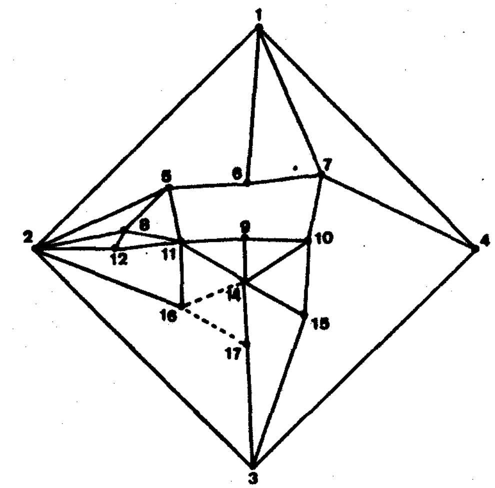
MENSAJE !!SR!! EL GRAFO ES BICONEXO

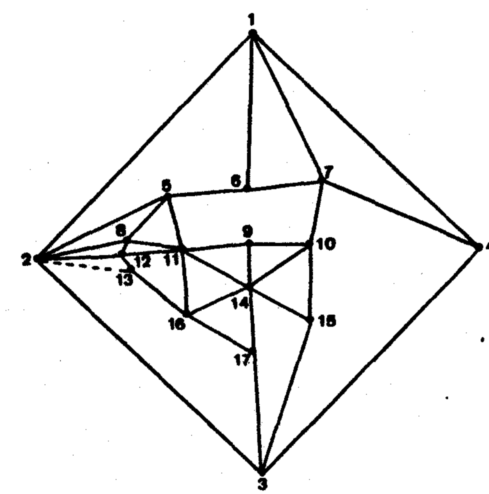
MENSAJE !!SR!! EL GRAFO ES PLANAR

MENSAJE !!SR!! EL GRAFO ES TRICONEXO

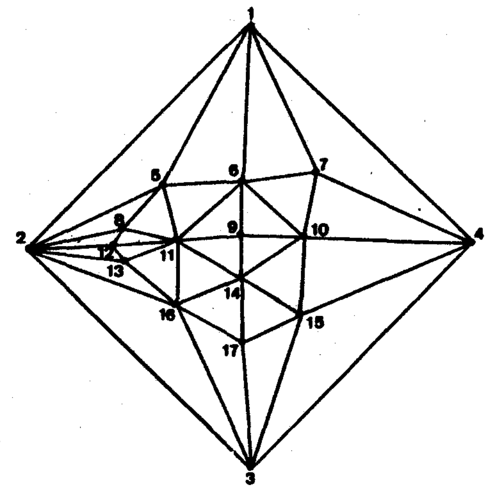
MENSAJE !!SR!! SE DIBUJA EN EL PLOTTER

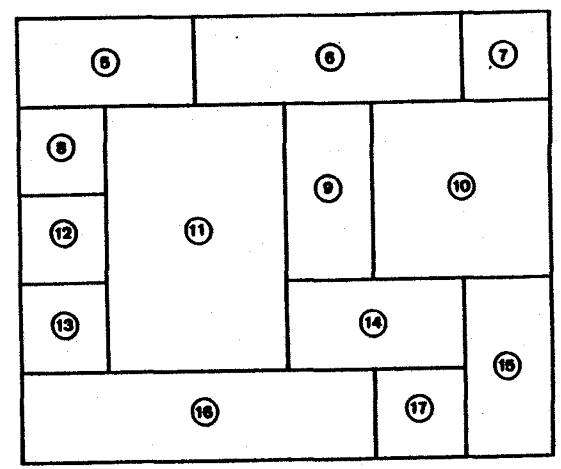
MENSAJE !!SR!! A LA VISTA DEL GRAFO TRICONEXO DIBUJADO, INTRODUCE LAS ARISTAS NECESARIAS PARA TRIANGULAR EL GRAFO.





```
MENSAJE ::SR:: EL NUMERO DE ARISTAS ACTUALES ES:  34

MENSAJE ::ED:: INTRODUCE LAS NUEVAS ARISTAS

               AL FINAL DE LA LISTA INTRODUCE: 0

  A  V  V
  -  -  -
 35  1  5
 36  3 16
 37  4 15
 38  6 11
 39  6  9
 40  6 10
 41  8 11
 42 11 13
 43 15 17
 44  4 10

MENSAJE ::SR:: GRAFO TRIANGULADO

MENSAJE ::SR:: SE DIBUJA EN EL PLOTTER

MENSAJE ::SR::, PON EL ORIGEN DE LA PLUMA DONDE DESEES

               LA DIMENSION DEL CONTORNO ES  12.00 X  10.00

MENSAJE ::SR:: SI DESEAS TERMINAR INTRODUCE: 0
               EN CASO CONTRARIO INTRODUCE: 1
```

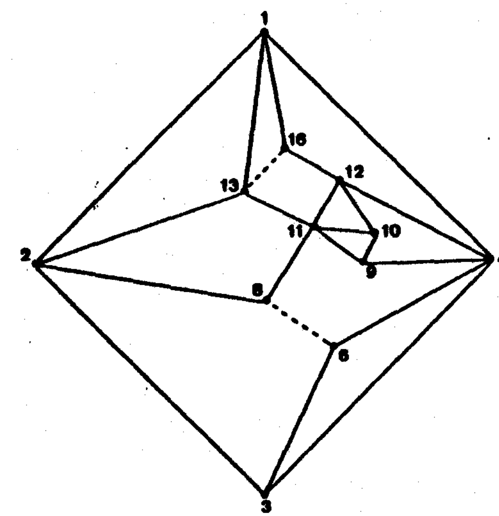




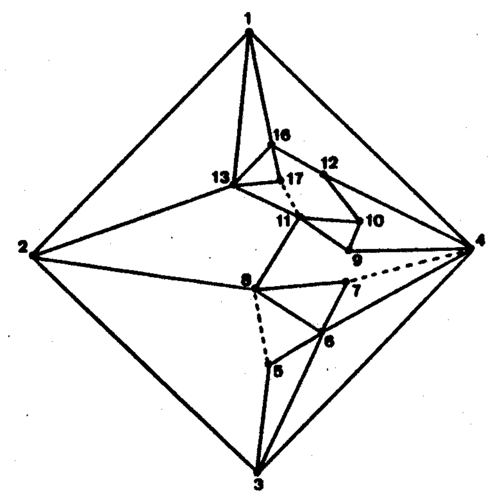
```
MENSAJE ::ED:: TITULA EL PROBLEMA

EJEMPLO 3 (ADIMENSIONAL)
****************************************************************************



MENSAJE ::ED:: ASIGNE UN NOMBRE A CADA UNO DE LOS ESPACIOS

               AL FINAL DE LA LISTA INTRODUCE:0

ESP  NOMBRE
---  ------
  5  VESTIBULO
  6  COCINA
  7  COMEDOR
  8  SALA DE ESTAR
  9  DORMITORIO DOBLE
 10  CUARTO DE BANO
 11  DISTRIBUIDOR
 12  DORMITORIO DOBLE
 13  DORMITORIO PRINCIPAL
 14  GUARDARROPA
 15  ASEO
 16  ARMARIO
 17  ARMARIO

MENSAJE ::ED:: MATRIZ DE INCIDENCIA

               LA NUMERACION DE LOS VERTICES  Y ARISTAS SE HARA A PARTIR DEL 5.
               LOS 4 PRIMEROS VERTICES Y ARISTAS CORRESPONDEN A LOS PUNTOS
               CARDINALES Y SUS RELACIONES. ESTAN INTRODUCIDOS AUTOMATICAMENTE.
               AL FINAL DE LA LISTA INTRODUCE: 0


  A  V  V
  -  -  -
  5  5  3
  6  5  6
  7  5 14
  8 14 15
  9  6  7
 10  7  8
 11  8  2
 12 13  2
 13 13  1
 14 13 17
 15 16 17
 16 16 12
 17 11 13
 18 11 12
 19 11  9
 20  6  3
 21  6  4
 22 12  4
 23  9  4
 24 11 10
 25  8 11
```

```
26 10 12
27  9 10
28  1 16

MENSAJE !!SR!! EL GRAFO ES CONEXO

MENSAJE !!SR!! EL GRAFO NO ES BICONEXO

LAS COMPONENTES BICONEXAS SON:

COMPONENTE:  1
--------------
ARISTAS:   8

COMPONENTE:  2
--------------
ARISTAS:   7

COMPONENTE:  3
--------------
ARISTAS:   4   1  13  15  14  18  19  24   9  10  11  25  17  12   2   3   6
           5  20  21  22  23  27  26  16  28

LOS VERTICES DE SEPARACION DE ESAS COMPONENTES SON:

COMPONENTES: 1- 2 .... VERT. DE SEPAR: 14
COMPONENTES: 2- 3 .... VERT. DE SEPAR:  5

                                                   COMPONENTE BICONEXA:  1
--------------------------------------------------------------------------

ARISTAS:   8

MENSAJE !!SR!! NO SE DIBUJA POR SER ELEMENTAL

                                                   COMPONENTE BICONEXA:  2
--------------------------------------------------------------------------

ARISTAS:   7

MENSAJE !!SR!! NO SE DIBUJA POR SER ELEMENTAL

                                                   COMPONENTE BICONEXA:  3
--------------------------------------------------------------------------

ARISTAS:   4   1  13  15  14  18  19  24   9  10  11  25  17  12   2   3   6
           5  20  21  22  23  27  26  16  28

MENSAJE !!SR!! LA COMPONENTE BICONEXA ES PLANAR

MENSAJE !!SR!! LA COMPONENTE BICONEXA NO ES TRICONEXA
```

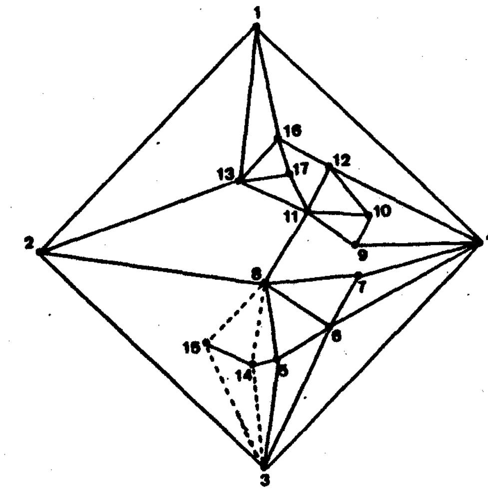
```
LAS COMPONENTES TRICONEXAS Y TRIANGULOS (DONDE LAS ARISTAS VIENEN EXPRESADAS
POR SUS VERTICES EXTREMOS) SON:

COMPONENTE: 1
ARISTAS:    1- 2   2- 3   3- 4   4- 1   4- 6   6- 3   6- 8   8- 2   8-11  11- 9
            9- 4   9-10  10-11  10-12  12-11  12- 4  12-16  16- 1  16-13  13-11
           13- 2  13- 1

MENSAJE ::SR:: ARISTA VIRTUAL EN LA COMPONENTE 1:   6- 8

MENSAJE ::SR:: ARISTA VIRTUAL EN LA COMPONENTE 1:  16-13

MENSAJE ::SR:: LAS ARISTAS VIRTUALES CITADAS HAN PASADO A SER REALES.
               POSTERIORMENTE,SI SE DESEA,PUEDEN SER CAMBIADAS POR OTRAS.

MENSAJE ::SR:: ADEMAS DE LAS ARISTAS PERTENECIENTES A ALGUNA COMPONENTE
               HAY OTRAS QUE NO PERTENECEN A NINGUNA DE ELLAS.

LAS ARISTAS QUE RELACIONAN UNAS COMPONENTES CON OTRAS Y LAS QUE NO ENLAZAN A
DOS COMPONENTES CUALESQUIERA SON:

ARISTA:  16-17  (NO LIGA A DOS COMPONENTES)
ARISTA:  17-12  (NO LIGA A DOS COMPONENTES)
ARISTA:   6- 7  (NO LIGA A DOS COMPONENTES)
ARISTA:   7- 8  (NO LIGA A DOS COMPONENTES)
ARISTA:   6- 5  (NO LIGA A DOS COMPONENTES)
ARISTA:   5- 3  (NO LIGA A DOS COMPONENTES)

--------------------------
COMPONENTE: 1 (TRICONEXA)
--------------------------

ARISTAS:    1- 2   2- 3   3- 4   4- 1   4- 6   6- 3   6- 8   8- 2   8-11  11- 9
            9- 4   9-10  10-11  10-12  12-11  12- 4  12-16  16- 1  16-13  13-11
           13- 2  13- 1

MENSAJE ::SR:: SE DIBUJA EN EL PLOTTER

MENSAJE ::SR:: SE HAN DE INTRODUCIR NUEVAS ARISTAS PARA CONVERTIR EN TRICONEXA
               LA COMPONENTE BICONEXA.

               LA ADICION SE HARA A LA VISTA DEL SUBGRAFO TRAZADO.
               SE PUEDE REALIZAR ESCALONADAMENTE,INTRODUCIENDO SOLO PARTE DE
               ELLAS, AÑADIENDO EL RESTO A PARTIR DE UN NUEVO TRAZADO.

MENSAJE ::SR:: EL NUMERO DE ARISTAS ACTUALES ES:  20

MENSAJE ::ED:: INTRODUCE LAS NUEVAS ARISTAS

               AL FINAL DE LA LISTA INTRODUCE: 0

  A  V  V
```

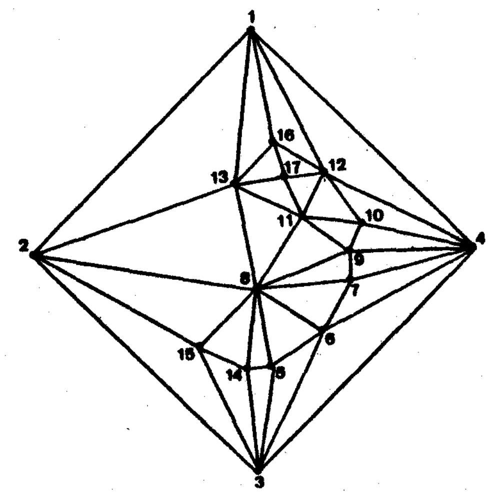


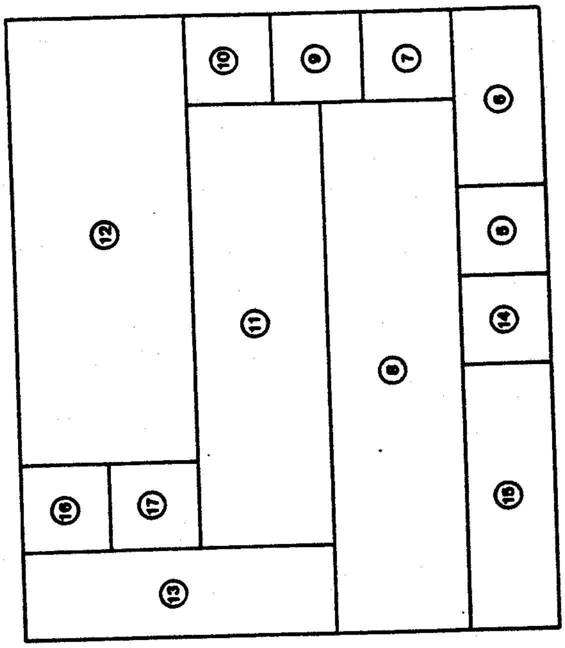
```
 -  -  -
31 11 17
32 10  4
33  7  4
34  5  8

MENSAJE ::SR:: SI SE DESEA SE PUEDE SUSTITUIR ARISTAS

               LOS DATOS SE DARAN EN EL SIGUIENTE ORDEN:
               - NUMERO DE LA ARISTA
               - VERTICE PRIMITIVO
               - VERTICE PRIMITIVO
               - VERTICE ACTUAL
               - VERTICE ACTUAL
               AL FINAL DE LA LISTA INTRODUCE: 0

  A  VP  VP  VA  VA
  -  --  --  --  --

MENSAJE ::SR:: LA COMPONENTE BICONEXA ES PLANAR

MENSAJE ::SR:: LA COMPONENTE BICONEXA ES ADEMAS TRICONEXA

MENSAJE ::SR:: SE DIBUJA EN EL PLOTTER

MENSAJE ::SR:: SE HAN DE ANADIR NUEVAS ARISTAS PARA QUE EL GRAFO SEA TRICONEXO

               ESTAS ARISTAS ENLAZARAN A LAS DISTINTAS COMPONENTES BICONEXAS
               ENTRE SI.

MENSAJE ::SR:: EL NUMERO DE ARISTAS ACTUALES ES:  34

MENSAJE ::ED:: INTRODUCE LAS NUEVAS ARISTAS

               AL FINAL DE LA LISTA INTRODUCE: 0

  A  V  V
  -  -  -
 35 14  8
 36 14  3
 37 15  8
 38 15  3

MENSAJE ::SR:: SI SE DESEA SE PUEDE SUSTITUIR ARISTAS

               LOS DATOS SE DARAN EN EL SIGUIENTE ORDEN:
               - NUMERO DE LA ARISTA
               - VERTICE PRIMITIVO
               - VERTICE PRIMITIVO
               - VERTICE ACTUAL
               - VERTICE ACTUAL
               AL FINAL DE LA LISTA INTRODUCE: 0
```



```
A  VP  VP  VA  VA
-  --  --  --  --

MENSAJE !!SR!! EL GRAFO ES BICONEXO

MENSAJE !!SR!! EL GRAFO ES PLANAR

MENSAJE !!SR!! EL GRAFO ES TRICONEXO
MENSAJE !!SR!! SE DIBUJA EN EL PLOTTER

MENSAJE !!SR!! A LA VISTA DEL GRAFO TRICONEXO DIBUJADO, INTRODUCE LAS ARISTAS
               NECESARIAS PARA TRIANGULAR EL GRAFO.

MENSAJE !!SR!! EL NUMERO DE ARISTAS ACTUALES ES:  38

MENSAJE !!ED!! INTRODUCE LAS NUEVAS ARISTAS

               AL FINAL DE LA LISTA INTRODUCE: 0

 A   V   V
 -   -   -
39  15   2
40  12   1
41   8   9
42  12  17
43  13   8
44   9   7

MENSAJE !!SR!! GRAFO TRIANGULADO
MENSAJE !!SR!! SE DIBUJA EN EL PLOTTER

MENSAJE !!SR!! PON EL ORIGEN DE LA PLUMA DONDE DESEES

               LA DIMENSION DEL CONTORNO ES  14.00 X  12.00

MENSAJE !!SR!! SI DESEAS TERMINAR INTRODUCE: 0
               EN CASO CONTRARIO INTRODUCE: 1
```






10
9
7
6
5
14
15
8
11
12
16
17
13

A.II.2. DIMENSIONAMIENTO

A.II.2.1. Manual del usuario

Ha de tenerse en cuenta los criterios generales señalados en A.II.1.1.

La secuencia de tarjetas o registros para la entrada de datos del programa DIM es casi toda previa a la ejecución del programa:

1. Titulación del problema

   FORMATO : (Un máximo de 80 caracteres alfanuméricos)

2. Número de vértices de los grafos dirigidos

   FORMATO: NVV, NVH

   NVV : nº de vértices del grafo vertical.
   NVH : nº de vértices del grafo horizontal.

3.a Coordenadas de los centros geométricos de los distintos espacios en el esquema adimensional (a = 1,2, ... n; n = nº de espacios)

4.a Condiciones de accesibilidad entre locales (a = 1,2, ... r; r = nº de condiciones)

   FORMATO : L, L1, L2, L3, ... Ln

   L : Local accesible a L_1, L_2, ... L_n

Si un local L_i es accesible a otro L_j, la comunicación es mútua. No será necesario, por tanto, señalar que L_j es accesible a L_i.

5. Tipo de problema

FORMATO : TIPO

TIPO = 1 .... LINEAL
TIPO = 2 .... NO LINEAL

6. Tipo de funci6n objetivo

FORMATO : OBJ

OBJ = 0 y TIPO = 1 .... MINIMIZACION DE LA LONGITUD Y ANCHURA

OBJ = 1 y TIPO = 1 .... MINIMIZACION DE UNA EXPRESION LINEAL

OBJ = 0 y TIPO = 2 .... MINIMIZACION DE LA SUPERFICIE

OBJ = 1 y TIPO = 2 .... MINIMIZACION DE UNA EXPRESION NO LINEAL.

7. Costos de la funci6n objetivo

FORMATO : C_5, C_6, C_7, ... C_{n+5} (n = n° de espacios)

C_i : Costo correspondiente al espacio i.

8. Superficie del contorno. (S6lo cuando TIPO = 2 y OBJ = 1)

FORMATO : SC

SC : Superficie del contorno

9.a Matriz de incidencia vertical (a = 1,2,.n+1; n = n° de espacios)

Si TIPO = 1 (problema lineal) FORMATO: LA,LO,LF,(CI),(CS)

LA: n° de la arista que representa al espacio o contorno. (Numerada a partir del 5. La arista que representa al contorno se numerará en último lugar).

LO: n° del vértice origen.

LF: n° del vértice extremo.

CI: valor de la cota inferior de la anchura.

CS: valor de la cota superior de la anchura.

Si se quiere dar un valor determinado a la anchura de una variable se da un valor a CI igual a CS.

Si TIPO = 2 (problema no lineal) FORMATO: LA,LO,LF,(CI),(SI)

LA: n° de la arista que representa al espacio o contorno. (Numerada a partir del 5)

LO: n° del vértice origen.

LF: n° del vértice extremo

CI: valor de la cota inferior de la anchura

SI: valor de la cota inferior de la superficie.

Los valores entre paréntesis en uno y otro caso indican que son datos opcionales. Si no se reflejan como datos los valores correspondientes se toman iguales a cero.

10.a Matriz de incidencia horizontal (a = 1,2, ... n+1; n = n° de espacios)

Si TIPO = 1 (Problema lineal) FORMATO: LA,LO,LF,(CI),(CS).

Igual significado que en la matriz de incidencia vertical pero referidos los valores de las cotas a la longitud en vez de a la anchura.

Si TIPO = 2 (Problema no lineal) FORMATO: LA,LO,LF,(CI)

Igual significado que en la matriz de incidencia vertical siendo el valor de la cota inferior de la longitud el que se dará ahora, y no figurando como entrada de datos la superficie mínima.

La numeración de las aristas en uno y otro grafo dirigido ha de ser de tal forma que un determinado espacio venga representado por un mismo número.

Una vez resuelto y trazado el esquema dimensionado el ordenador envía un mensaje por si se desea terminar o se quiere ajustar los resultados. En el primer caso se introduce un: "0", en el segundo un: "1". En este último caso hay que efectuar una nueva entrada de datos:

1. Valoración relativa para los distintos espacios

FORMATO : V_5, V_6, V_7, V_8 ... V_{n+5} (n: n° de espacios)

V_i : Valoración relativa para el espacio i.

Hay una sola opción prevista para salida de resultados intermedios: la impresión de la matriz de restricciones, los términos independientes y los códigos de las restricciones. Si se desea se pulsará el SWITCH 1.

Los mensajes de errores que el ordenador puede enviar a lo largo de la ejecución del programa son los siguientes:

ERROR 1 : El número de vértices o aristas es mayor que el permitido.

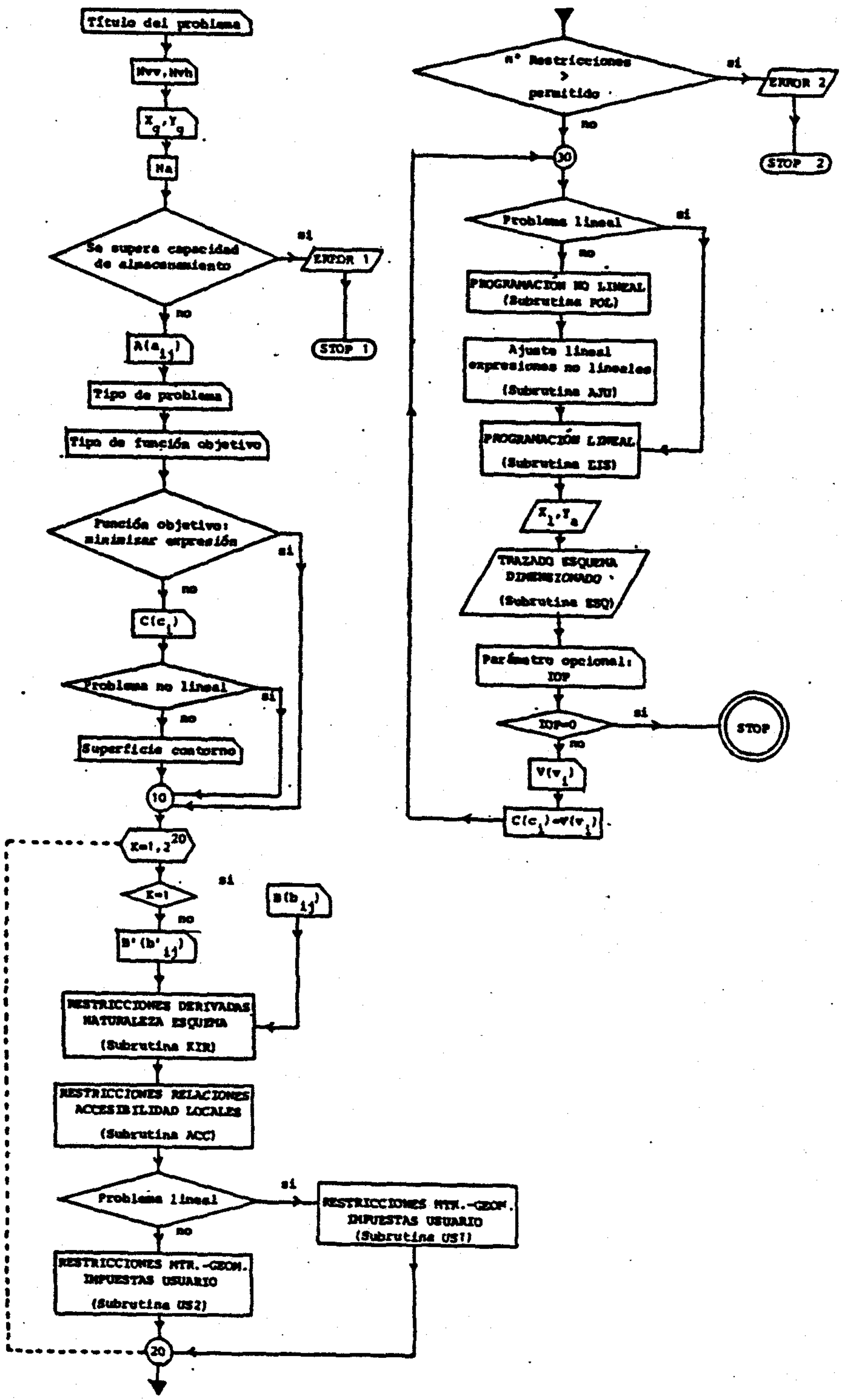
ERROR 2 : El número de restricciones supera la capacidad dispuesta.

ERROR 3 : El problema no está acotado.

ERROR 4 : El problema no tiene solución admisible.

El programa que se presenta está elaborado para un máximo de 60 aristas, 40 vértices y 240 expresiones para restricciones.

A.II.2.2. Organigrama general



Nvv: n° de vértices verticales
Nvh: n° de vertices horizontales

X_g, Y_g: coordenadas de los centros geométricos de cada espacio.

Na: n°de aristas

$A(a_{ij})$: matriz de accesibilidad entre locales.

X_l: longitud de los espacios y contorno.
Y_a: anchura de los espacios y contorno.

$C(c_i)$: costes de la función objetivo.

$V(v_i)$: valoración relativa para cada espacio.

$B(b_{ij})$: matriz de incidencia vertical

$B'(b'_{ij})$: matriz de incidencia horizontal.

A.II.2.3. Listado del Programa

```
FTN4
$EMA(IJK,3)
C
C
C      ***********
       PROGRAM DIM
C      ***********
C
C-------------------------------------------------------------------
C   PROGRAMA PARA DIMENSIONAMIENTO DE ESQUEMAS DISTRIBUTIVOS EN PLANTA
C-------------------------------------------------------------------
C
C
C TABLAS UTILIZADAS
C
C BB(60,40) ... MATRIZ DE INCIDENCIA HORIZONTAL
C BA(60,40) ... MATRIZ DE INCIDENCIA VERTICAL
C IR(240.120) . MATRIZ DE RESTRICCIONES
C IA(60,60) ... MATRIZ DE ACCESIBILIDAD
C
C
C LISTADO DE ERRORES
C
C ERROR 1 ..... NUMERO DE VERTICES O ARISTAS MAYOR QUE EL PERMITIDO
C ERROR 2 ..... EL NUMERO DE RESTRICCIONES SUPERA LA CAPACIDAD DISPUESTA
```

```
C ERROR 3 ..... PROBLEMA NO ACOTADO
C ERROR 4 ..... EL PROBLEMA NO TIENE SOLUCION ADMISIBLE
C
C
C ESCRITURAS OPCIONALES
C
C SWITCH 1 .... MATRIZ DE RESTRICCIONES, TERMINOS INDEPENDIENTES Y CODI-
C               GOS DE CADA UNA DE ELLAS.
C
C
      INTEGER CODE(240)
      DIMENSION IPAR(4),NOM(40),JAUX(150),X(60),Y(60),B(240,3),C(60)
      DIMENSION XB(120),ZETA(120)
      COMMON /IJK/ BA(60,40),BB(60,40),IR(240,120),IA(60,60)
C
C PARAMETROS DE ENTRADAS Y SALIDAS
C
      CALL RMPAR(IPAR)
      IENT=IPAR(1)
      ISAL=IPAR(2)
      IEAUX=IPAR(3)
      ISAUX=IPAR(4)
      IF(IENT.EQ.0) IENT=1
      IF(ISAL.EQ.0) ISAL=6
      IF(IEAUX.EQ.0) IEAUX=1
      IF(ISAUX.EQ.0) ISAUX=6
C
C LECTURA PREVIA
C
      WRITE(ISAL,1001)
      WRITE(ISAL,3001)
      READ(IENT,2001) NOM
      WRITE(ISAUX,5001) NOM
      WRITE(ISAL,3002)
      READ(IENT,*) NVV,NVH
      WRITE(ISAUX,5002) NVV
      WRITE(ISAUX,5003) NVH
      NV=NVV
      IF(NVH.GT.NVV) NV=NVH
C
C LECTURA DE LAS COORDENADAS DEL CENTRO GEOMETRICO DE CADA UNA DE LAS
C ESTANCIAS EN EL ESQUEMA ADIMENSIONAL.
C
      WRITE(ISAL,3004)
      DO 10 J=1,70
      READ(IENT,*) I,XX,YY
      IF(I.EQ.0) GO TO 15
      WRITE(ISAUX,5005) I,XX,YY
      II=I-4
      X(II)=XX
      Y(II)=YY
   10 CONTINUE
   15 NA=J
      NVI=NA-1
C
C CONTROL SI SE SUPERA LA CAPACIDAD DE ALMACENAMIENTO
C
      IF(NV.LE.40.AND.NA.LE.60) GO TO 20
      IERR=NER(ISAL,1)
      STOP 0001
C
```

```
C PUESTA A CERO
C
   20 DO 30 I=1,5*NA
      DO 30 J=1,3
      B(I,J)=0
   30 CONTINUE
      DO 35 I=1,5*NA
      CODE(I)=0
   35 CONTINUE
      DO 40 I=1,2*NA
      ZETA(I)=0
   40 CONTINUE
      DO 50 I=1,5*NA
      DO 50 J=1,2*NA
      IR(I,J)=0
   50 CONTINUE
      DO 60 I=1,NA
      DO 60 J=1,NV
      BB(I,J)=0
      BA(I,J)=0
   60 CONTINUE
      DO 70 I=1,NA
      DO 70 J=1,NA
      IA(I,J)=0
   70 CONTINUE
      DO 80 I=1,NA
      JAUX(I)=0
      C(I)=0
   80 CONTINUE
C
C LECTURA DE LA MATRIZ DE ACCESIBILIDAD
C
      WRITE(ISAL,3005)
      DO 150 K=1,100
      READ(IENT,*) I,(JAUX(J),J=1,NVI)
      IF(I.EQ.0) GO TO 160
      DO 100 J=1,NVI
      IF(JAUX(J).EQ.0) GO TO 110
      II=I-4
      JJ=JAUX(J)-4
      IA(II,JJ)=1
      IA(JJ,II)=1
  100 CONTINUE
  110 WRITE(ISAUX,5006) I,(JAUX(JJ),JJ=1,J-1)
      DO 120 J=1,NVI
      IF(JAUX(J).EQ.0) GO TO 150
      JAUX(J)=0
  120 CONTINUE
  150 CONTINUE
C
C LECTURA DEL TIPO DE PROBLEMA
C IOP=1. PROBLEMA LINEAL  IOP=2. PROBLEMA NO LINEAL.
C
  160 WRITE(ISAL,3006)
      READ(IENT,*) IOP
      IF(IOP.EQ.1) WRITE(ISAUX,5007)
      IF(IOP.EQ.2) WRITE(ISAUX,5008)
      LL=0
      MM=0
      JOP=0
```

```
C
C LECTURA DEL PARAMETRO OPCIONAL:
C PROBLEMA LINEAL:
C ITIP=0 MINIMIZAR LA LONGITUD Y ANCHURA.  ITIP=1 MINIMIZAR UNA FUNCION
C PROBLEMA NO LINEAL:
C ITIP=0 MINIMIZACION SUPERFICIE.  ITIP=1 MINIMIZACION FUNCION OBJETIVO
C
      WRITE(ISAL,3007)
      READ(IENT,*) ITIP
      IF(ITIP.EQ.0) GO TO 220
      WRITE(ISAUX,5009)
C
C LECTURA DE LOS COSTES DE LA FUNCION OBJETIVO SI SE HA ELEGIDO TIPO 1
C
      WRITE(ISAUX,3008)
      READ(IENT,*) (C(I),I=1,NVI)
      WRITE(ISAUX,5010) (C(I),I=1,NVI)
      DO 205 I=1,NVI
  205 C(I)=-C(I)
      IF(IOP.EQ.1) GO TO 290
C
C LECTURA EN EL PROBLEMA NO LINEAL DEL AREA TOTAL SI ESTA NO ES DATO
C
      WRITE(ISAL,3009)
      READ(IENT,*) SM
      LL=LL+1
      B(LL,1)=SM
      CODE(LL)=5
      DO 210 I=1,NVI
      IR(LL,I)=1
      IR(LL,I+NA)=1
  210 CONTINUE
      WRITE(ISAUX,5011) SM
      GO TO 290
  220 C(NA)=1.
      IF(IOP.EQ.1) WRITE(ISAUX,5012)
      IF(IOP.EQ.2) WRITE(ISAUX,5013)
C
C SERIE DE OPERACIONES A REALIZAR EN DOS CICLOS.
C PRIMER CICLO: GRAFO DIRIGIDO VERTICAL
C SEGUNDO CICLO: GRAFO DIRIGIDO HORIZONTAL
C
C LOS CODIGOS DE LAS RESTRICCIONES SON:
C PROBLEMA LINEAL: 0 MENOR O IGUAL
C                  1 MAYOR O IGUAL
C                  2 IGUAL
C PROBLEMA NO LINEAL: 1 RELACION LINEAL DE ADYACENCIA (MAYOR O IGUAL)
C                     2 RELACION LINEAL DE USUARIO    (MAYOR O IGUAL)
C                     3 RELACION NO LINEAL DE USUARIO (MAYOR O IGUAL)
C                     4 RELACION LINEAL KIRCHHOFF     (IGUAL)
C                     5 RELACION NO LINEAL USUARIO    (IGUAL)
C
  290 DO 500 K=1,2
C
C LECTURA DE LA MATRIZ DE INCIDENCIA VERTICAL O HORIZONTAL
C SE REFLEJAN ADEMAS LAS RESTRICCIONES IMPUESTAS POR EL USUARIO
C
      IF(K.EQ.1) WRITE(ISAL,3010)
      IF(K.EQ.2) WRITE(ISAL,3011)
      WRITE(ISAL,3012)
      IF(K.EQ.1.AND.IOP.EQ.1) WRITE(ISAL,3013)
```

```
      IF(K.EQ.2.AND.IOP.EQ.1) WRITE(ISAL,3014)
      IF(K.EQ.1.AND.IOP.EQ.2) WRITE(ISAL,3015)
      IF(K.EQ.2.AND.IOP.EQ.2) WRITE(ISAL,3016)
      DO 300 I=1,70
      READ(IENT,*) LA,LO,LF,RO,RF
      IF(LA.EQ.0) GO TO 310
      IF(IOP.EQ.1) WRITE(ISAUX,5014) LA,LO,LF,RO,RF
      IF(IOP.EQ.2.AND.K.EQ.1) WRITE(ISAUX,5014) LA,LO,LF,RO,RF
      IF(IOP.EQ.2.AND.K.EQ.2) WRITE(ISAUX,5015) LA,LO,LF,RO
      IF(RO.EQ.0) RO=1.01
      IF(RF.EQ.0) RF=1.01
      LA=LA-4
      IF(K.EQ.1) BA(LA,LO)=RO
      IF(K.EQ.2) BB(LA,LO)=RO
      IF(K.EQ.1) BA(LA,LF)=-RF
      IF(K.EQ.2) BB(LA,LF)=-RF
      RF=0.
      RO=0.
  300 CONTINUE
C
C  ** LLAMADA A LA SUBRUTINA KIR QUE OBTIENE RESTRICCIONES POR LA PRIME-
C     RA LEY DE KIRCHHOFF **
C
  310 IF(K.EQ.1.AND.IOP.EQ.1) CALL KIR(NA,NVV,LL,1,IOP,B,CODE,BA,IR)
      IF(K.EQ.1.AND.IOP.EQ.2) CALL KIR(NA,NVV,LL,K,IOP,B,CODE,BA,IR)
      IF(K.EQ.2.AND.IOP.EQ.1) CALL KIR(NA,NVH,LL,1,IOP,B,CODE,BB,IR)
      IF(K.EQ.2.AND.IOP.EQ.2) CALL KIR(NA,NVH,LL,K,IOP,B,CODE,BB,IR)
C
C  ** LLAMADA A LA SUBRUTINA ACC QUE OBTIENE LAS RESTRICCIONES DE LAS
C     RELACIONES DE ACCESIBILIDAD **
C
      IF(K.EQ.1.AND.IOP.EQ.1) CALL ACC(NVV,NVI,NA,LL,1,CODE,B,Y,X,BA,
     &IR,IA)
      IF(K.EQ.1.AND.IOP.EQ.2) CALL ACC(NVV,NVI,NA,LL,K,CODE,B,Y,X,BA,
     &IR,IA)
      IF(K.EQ.2.AND.IOP.EQ.1) CALL ACC(NVH,NVI,NA,LL,1,CODE,B,X,Y,BB,
     &IR,IA)
      IF(K.EQ.2.AND.IOP.EQ.2) CALL ACC(NVH,NVI,NA,LL,K,CODE,B,X,Y,BB,
     &IR,IA)
C
C  ** LLAMADA A LA SUBRUTINA US1 O US2 QUE OBTIENEN LAS RESTRICCIONES
C     DIMENSIONALES  DEL USUARIO SEGUN SEA EL TIPO DE PROBLEMA **
C
      IF(K.EQ.1.AND.IOP.EQ.1) CALL US1(NA,NVV,LL,B,CODE,BA,IR)
      IF(K.EQ.2.AND.IOP.EQ.1) CALL US1(NA,NVH,LL,B,CODE,BB,IR)
      IF(K.EQ.1.AND.IOP.EQ.2) CALL US2(NA,NVV,LL,K,B,CODE,BA,IR)
      IF(K.EQ.2.AND.IOP.EQ.2) CALL US2(NA,NVH,LL,K,B,CODE,BB,IR)
      IF(K.EQ.1) MM=LL
  500 CONTINUE
C
C CONTROL POR SI EL NUMERO DE RESTRICCIONES ES MAYOR QUE EL PERMITIDO
C
      IF(LL.LE.240.AND.LL.LE.5*NA) GO TO 501
      IERR=NER(ISAL,2)
      STOP 0002
C
C ESCRITURA OPCIONAL DE LA MATRIZ DE RESTRICCIONES, TERMINOS INDEPEN -
C DIENTES Y CODIGOS.
C
  501 IF(ISSW(1).LT.0) 350,555
```

```
 350 WRITE(ISAUX,5016) K                                               DIN-267
     DO 400 I=1,K                                                      DIN-268
     IF(IOP.EQ.2) WRITE(ISAUX,5016) (IR(I,J),J=1,2*NA)                 DIN-269
     IF(IOP.EQ.1) WRITE(ISAUX,5016) (IR(I,J),J=1,NA)                   DIN-270
 400 CONTINUE                                                          DIN-271
     WRITE(ISAUX,5016) K                                               DIN-272
     DO 450 I=MM+1,LL                                                  DIN-273
     WRITE(ISAUX,5010) B(I,1),CODE(I)                                  DIN-274
 450 CONTINUE                                                          DIN-275
 505 IF(IOP.EQ.2) GO TO 610                                            DIN-276
 510 NET=0                                                             DIN-277
     NLET=0                                                            DIN-278
     NGET=0                                                            DIN-279
     DO 550 I=1,MM                                                     DIN-280
     IF(CODE(I).EQ.0) NLET=NLET+1                                      DIN-281
     IF(CODE(I).EQ.1) NGET=NGET+1                                      DIN-282
     IF(CODE(I).EQ.2) NET=NET+1                                        DIN-283
 550 CONTINUE                                                          DIN-284
     INDS=0                                                            DIN-285
C                                                                      DIN-286
C  ** LLAMADA A LA SUBRUTINA QUE SOLUCIONA EL PROBLEMA DE PROGRAMACION DIN-287
C     LINEAL UTILIZANDO EL ALGORITMO SIMPLEX **                        DIN-288
C                                                                      DIN-289
     CALL LIS(ISAL,NA,MM,1,NET,NGET,NLET,ITIP,CODE,B,NOM,C,XB,IR(1,1), DIN-290
    &IR(1,61),ZETA,INDS)                                               DIN-291
     INDS=1                                                            DIN-292
     NET=0                                                             DIN-293
     NLET=0                                                            DIN-294
     NGET=0                                                            DIN-295
     DO 600 I=MM+1,LL                                                  DIN-296
     IF(CODE(I).EQ.0) NLET=NLET+1                                      DIN-297
     IF(CODE(I).EQ.1) NGET=NGET+1                                      DIN-298
     IF(CODE(I).EQ.2) NET=NET+1                                        DIN-299
 600 CONTINUE                                                          DIN-300
     CALL LIS(ISAL,NA,LL,MM+1,NET,NGET,NLET,ITIP,CODE,B,NOM,C,XB,IR(1,1 DIN-301
    &),IR(1,61),ZETA,INDS)                                             DIN-302
 601 WRITE(ISAL,4001)                                                  DIN-303
     DO 605 J=1,NA                                                     DIN-304
     JJ=J+4
     WRITE(ISAL,4002) JJ,ZETA(J+NA),JJ,ZETA(J)                         DIN-306
 605 CONTINUE                                                          DIN-307
     WRITE(ISAL,4003)                                                  DIN-308
C                                                                      DIN-309
C  ** LLAMADA A LA SUBRUTINA QUE DIBUJA EL ESQUEMA DE PLANTA YA DIMEN- DIN-310
C     SIONADO **                                                       DIN-311
C                                                                      DIN-312
     CALL ESQ(ISAL,NA,AVN,AVV,ZETA(NA+1),ZETA(1),JAUX(2*NA+1),JAUX(1), DIN-313
    &JAUX(NA+1),BA,BB,K,Y)                                             DIN-314
     IF(JOP.EQ.0) 620,350                                              DIN-315
C                                                                      DIN-316
C  ** LLAMADA A LA SUBRUTINA QUE RESUELVE EL PROBLEMA DE PROGRAMACION NO DIN-317
C     LINEAL UTILIZANDO EL ALGORITMO DE POLAK-RIVIERE **               DIN-318
C                                                                      DIN-319
 610 IF(IOP.EQ.2) CALL POL(ISAL,NA,LL,C,CODE,B,ZETA,IR)                DIN-320
C                                                                      DIN-321
C  ** LLAMADA A LA SUBRUTINA QUE AJUSTA LOS VALORES OBTENIDOS PARA LOS DIN-322
C     ESPACIOS SI SE HA EMPLEADO LA PROGRAMACION NO LINEAL.            DIN-323
C                                                                      DIN-324
     CALL AJU(NA,LL,NET,NGET,NLET,ITIP,MM,CODE,B,NOM,C,XB,ZETA,INDS,   DIN-325
    &IR,JAUX)                                                          DIN-326
     MM=MM-1                                                           DIN-327
```

```
      GO TO 601
  620 WRITE(ISAL,3017)
C
C LECTURA DE UN PARAMETRO OPCIONAL POR SI SE QUIERE AJUSTAR LOS VALORES
C HALLADOS
C
      READ(IEAUX,*) JOP
      IF(JOP.EQ.0) GO TO 850
      WRITE(ISAL,3018)
      READ(IEAUX,*) (C(I),I=1,NVI)
      WRITE(ISAUX,5010) (C(I),I=1,NVI)
      DO 650 I=1,NVI
      C(I)=-C(I)
  650 CONTINUE
      LL=LL+1
      IR(LL,NA)=1
      B(LL,1)=ZETA(2*NA)
      CODE(LL)=2
      DO 750 I=LL,MM+1,-1
      B(I+1,1)=B(I,1)
      CODE(I+1)=CODE(I)
      DO 700 J=1,NA
      IR(I+1,J)=IR(I,J)
  700 CONTINUE
  750 CONTINUE
      DO 800 J=1,NA
      IR(MM+1,J)=0
  800 CONTINUE
      MM=MM+1
      IR(MM,NA)=1
      B(MM,1)=ZETA(NA)
      CODE(MM)=2
      LL=LL+1
      GO TO 510
  850 STOP
C
C
C FORMATOS
C
 1001 FORMAT(1H1)
 2001 FORMAT(40A2)
 3001 FORMAT(/1X"MENSAJE ::ED:: TITULA EL PROBLEMA"/)
 5001 FORMAT(1X,40A2,/1X,79"*",//)
 3002 FORMAT(1X"MENSAJE ::ED:: NUMERO DE VERTICES VERTICALES Y HORIZONTA
     &LES"/)
 5002 FORMAT(1X"NUMERO DE VERTICES VERTICALES ...."I3)
 5003 FORMAT(1X"NUMERO DE VERTICES HORIZONTALES .."I3,//)
 3004 FORMAT(1X"MENSAJE ::ED:: COORDENADAS DEL CENTRO GEOMETRICO DE LOS
     &ESPACIOS."//16X"SE HARA A PARTIR DEL ESQUEMA ADIMENSIONAL,DANDO UN
     &AS DIMENSIONES"/16X"ARBITRARIAS A CADA ESPACIO. ESTOS SE NUMERARAN
     &A PARTIR DEL 5."/16X"AL FINAL DE LA LISTA INTRODUCE: 0"//1X"ESP
     &   XG      YG"/1X"---      --      --")
 5005 FORMAT(I4,2F7.2)
 3005 FORMAT(//1X"MENSAJE ::ED:: CONDICIONES DE ACCESIBILIDAD ENTRE LOCA
     &LES"//16X"SE FIJA UN LOCAL Y SE INDICA CUALES DEBEN SER ACCESIBLES
     & A EL."/16X"AL FINAL DE LA LISTA INTRODUCE: 0"//1X" L   L1   L2   L3
     & L4   L5"/1X" -   --   --   --   --   --")
 5006 FORMAT(I3,30I4,//)
 3006 FORMAT(//1X"MENSAJE ::ED:: TIPO DE PROBLEMA:  1 .. LINEAL / 2 .. N
     &O LINEAL"/)
```

```
5007 FORMAT(//1X"PROBLEMA LINEAL"//)
5008 FORMAT(//1X"PROBLEMA NO LINEAL"//)
3007 FORMAT(1X"MENSAJE !!ED!! TIPO DE FUNCION OBJETIVO:"//16X"PROBLEMA
    &LINEAL ...... 0: MINIMIZACION DE LA LONGITUD Y ANCHURA"/39X"1: MIN
    &IMIZACION DE UNA EXPRESION"/16X"PROBLEMA NO LINEAL ... 0: MINIMIZA
    &CION DE LA SUPERFICIE"/39X"1: MINIMIZACION DE UNA EXPRESION"//)
5009 FORMAT(1X,"FUNCION OBJETIVO:  MINIMIZACION DE UNA EXPRESION"//)
3008 FORMAT(1X"MENSAJE !!ED!! COSTOS DE LA FUNCION OBJETIVO"//1X" C5  C
    &6  C7  C8  C9 C10 C11 C12 C13 C14 C15 C16"/1X" --  --  --  --  --
    &--- --- --- --- --- --- ---")
5019 FORMAT(30F4.1)
3009 FORMAT(//1X,"MENSAJE !!ED!! SUPERFICIE DE PLANTA"/)
5011 FORMAT(1X"SUPERFICIE DE PLANTA"F6.2,//)
5012 FORMAT(1X,"FUNCION OBJETIVO: MINIMIZACION DE LA LONGITUD Y ANCHURA
    &")
5013 FORMAT(1X,"FUNCION OBJETIVO: MINIMIZACION DE LA SUPERFICIE"//)
3010 FORMAT(//1X"MENSAJE !!ED!! MATRIZ DE INCIDENCIA VERTICAL"//16X"LA
    &NUMERACION DE LOS ESPACIOS SE HARA A PARTIR DEL 5."/16X"LA ARISTA
    &QUE REPRESENTA AL CONTORNO SE NUMERARA EN ULTIMO LUGAR.")
3011 FORMAT(//1X"MENSAJE !!ED!! MATRIZ DE INCIDENCIA HORIZONTAL"//16X"I
    &GUAL CRITERIO DE NUMERACION QUE EN LA VERTICAL.")
3012 FORMAT(/16X"SE RECOGEN AQUI LOS REQUISITOS METRICO-GEOMETRICOS."/1
    &6X"LOS DATOS SE DAN EN EL SIGUIENTE ORDEN:"/16X"- ARISTA"/16X"- VE
    &RTICE ORIGEN"/16X"- VERTICE EXTREMO")
3013 FORMAT(16X"- COTA INFERIOR DE LA ANCHURA",/16X"- COTA SUPERIOR DE
    &LA ANCHURA",//1X"  A  VO  VE   CI   CS"/1X"  -  --  --   --   --")
3014 FORMAT(16X"- COTA INFERIOR DE LA LONGITUD"/16X"- COTA SUPERIOR DE
    &LA LONGITUD"//1X"  A  VO  VE   CI   CS"/1X"  -  --  --   --   --")
3015 FORMAT(16X"- COTA INFERIOR DE LA ANCHURA",/16X"- COTA INFERIOR DE
    &LA SUPERFICIE"//1X"  A  VO  VE   CI   CI"/1X"  -  --  --   --   --
    &")
3016 FORMAT(15X"- COTA INFERIOR DE LA LONGITUD"//1X"  A  VO  VE   CI"/1
    &X"  -  --  --   --")
5014 FORMAT(3I4,2F5.1)
5015 FORMAT(3I4,F5.1)
4001 FORMAT(//1X,8"-"," LOS RESULTADOS OBTENIDOS PARA CADA ESPACIO Y EL
    & CONTORNO SON: ",8"-"/)
4002 FORMAT(1X"X("I2")="F6.2,3X,"Y("I2")="F6.2)
4003 FORMAT(/1X,79"-"///)
3017 FORMAT(//1X"MENSAJE !!ED!! SI SE DESEA AJUSTAR LOS RESULTADOS INTR
    &ODUCE: 1"/16X"SI QUIERES TERMINAR ............... INTRODUCE: 0"//)
3018 FORMAT(1X"MENSAJE !!ED!! VALORACION RELATIVA PARA CADA ESPACIO."//
    &16X"LOS VALORES ASIGNADOS SERAN PROPORCIONALES A LA IMPORTANCIA QU
    &E"/16X"SE ATRIBUYE A LA SUPERFICIE DE CADA ESPACIO."//1X" V5  V6
    &V7  V8  V9 V10 V11 V12 V13V14 V15 V16"/1X" --  --  --  --  --  ---
    & --- --- --- --- --- ---")
5016 FORMAT(40I3)
     END
C
C
$EMA(IJK,3)
     SUBROUTINE KIR(HA,HV,LL,K,IDP,B,CODE,BB,IR)
C    ******************************************
C
C SUBRUTINA QUE OBTIENE LAS RESTRICCIONES DERIVADAS DE LA PRIMERA LEY DE
C KIRCHHOFF.
C
     EMA BB(60,40),IR(240,120)
     INTEGER CODE(240)
     DIMENSION B(240,3)
     JJ=LL
```

```
C
C ELABORACION DE LA RESTRICCIONES POR KIRCHHOFF
C
      DO 100 I=1,NA
      DO 50 J=1,NV-1
      IF(BB(I,J).EQ.0) GO TO 50
      IR(JJ+J,I+(K-1)*NA)=BB(I,J)/ABS(BB(I,J))
   50 CONTINUE
  100 CONTINUE
      LL=JJ+NV-1
C
C VECTORES CODIGO Y TERMINOS INDEPENDIENTES
C
      DO 150 I=1,NV-1
      IF(IOP.EQ.2) CODE(I+JJ)=4
      IF(IOP.EQ.1) CODE(I+JJ)=2
      B(I+JJ,1)=0.
  150 CONTINUE
      RETURN
      END
C
C
$EMA(IJK,3)
      SUBROUTINE ACC(NV,NVI,NA,LL,K,CODE,B,X,Y,BB,IR,IA)
C     **************************************************
C
C SUBRUTINA QUE OBTIENE LAS RESTRICCIONES DERIVADAS DE LAS RELACIONES DE
C ACCESIBILIDAD IMPUESTAS POR EL USUARIO.
C
      EMA BB(60,40),IR(240,120),IA(60,60)
      INTEGER CODE(240)
      DIMENSION B(240,3),X(NA),Y(NA)
      NN=LL
C
C ELABORACION DE LAS RESTRICCIONES
C
      DO 250 I=1,NVI-1
      DO 200 J=I+1,NVI
      IF(IA(I,J).EQ.0) GO TO 200
      DO 50 JJ=1,NV
      IF(BB(I,JJ).EQ.0.OR.BB(J,JJ).EQ.0) GO TO 50
      IF(BB(I,JJ).GT.0.AND.BB(J,JJ).GT.0) GO TO 50
      IF(BB(I,JJ).LT.0.AND.BB(J,JJ).LT.0) GO TO 50
      GO TO 60
   50 CONTINUE
      GO TO 200
   60 LL=LL+1
      IR(LL,I+(K-1)*NA)=1
      IR(LL+1,J+(K-1)*NA)=1
      DO 150 II=1,NA-1
      IF(II.EQ.I.OR.II.EQ.J) GO TO 150
      IF(BB(II,JJ).EQ.0) GO TO 150
      IF(K.EQ.1) 70,80
   70 IF(Y(J).GT.Y(I)) 110,120
   80 IF(Y(J).GT.Y(I)) 120,110
  110 IF(BB(II,JJ).LT.0) 130,140
  120 IF(BB(II,JJ).GT.0) 130,140
  130 IF(X(II).LT.X(I)) GO TO 150
      IR(LL,II+(K-1)*NA)=1
      IR(LL+1,II+(K-1)*NA)=-1
```

```
      GO TO 150
  140 IF(K(II).LT.K(J)) GO TO 150
      IR(LL,II+(K-1)*NA)=-1
      IR(LL+1,II+(K-1)*NA)=1
  150 CONTINUE
      K1=0
      K2=0
C
C SI TODOS LOS TERMINOS DE LA FILA SON POSITIVOS ESTA SE ELIMINA
C
      DO 155 II=1,NA*K
      IF(IR(LL,II).LT.0) K1=K1+1
      IF(IR(LL+1,II).LT.0) K2=K2+1
  155 CONTINUE
      IF(K1.NE.0.OR.K2.EQ.0) GO TO 165
      DO 160 II=1,NA*K
      IR(LL,II)=IR(LL+1,II)
      IR(LL+1,II)=0.
  160 CONTINUE
      GO TO 200
  165 IF(K1.EQ.0.OR.K2.NE.0) GO TO 175
      DO 170 II=1,NA*K
      IR(LL+1,II)=0.
  170 CONTINUE
      GO TO 200
  175 IF(K1.NE.0.AND.K2.NE.0) GO TO 185
      DO 180 II=1,NA*K
      IR(LL,II)=0.
      IR(LL+1,II)=0.
  180 CONTINUE
      LL=LL-1
  185 IF(K1.NE.0.AND.K2.NE.0) LL=LL+1
  200 CONTINUE
  250 CONTINUE
      IF(LL.EQ.NN) GO TO 360
      DO 300 I=NN+1,LL
      CODE(I)=1
      B(I,1)=1.
  300 CONTINUE
  360 RETURN
      END
C
C
$EMA(IJK,3)
      SUBROUTINE US1(NA,NV,LL,B,CODE,BB,IR)
C     ************************************
C
C EXPRESA LAS RESTRICCIONES DIMENSIONALES IMPUESTAS POR EL USUARIO EN EL
C PROBLEMA LINEAL.
C
      EMA BB(60,40),IR(240,120)
      INTEGER CODE(240)
      DIMENSION B(240,3)
      NN=LL
C
C DETERMINACION DE LAS COTAS SUPERIOR E INFERIOR PARA UNA VARIABLE.
C SI LAS DOS SON IGUALES LA RESTRICCION ES DEL TIPO IGUAL.
C
      DO 100 I=1,NA
      J1=0
      J2=0
```

```
      DO 50 J=1,NV
      IF(BB(I,J).EQ.0) GO TO 50
      IF(BB(I,J).GT.0) J1=J
      IF(BB(I,J).LT.0) J2=J
   50 CONTINUE
      IF(BB(I,J1).NE.-BB(I,J2)) GO TO 60
      MM=MM+1
C
C SI UNA VARIABLE NO POSEE COTA INFERIOR, SE HACE ESTA IGUAL A CERO.
C
      IF(BB(I,J1).EQ.1.01) GO TO 70
      IF(BB(I,J1).NE.1.01) B(MM,1)=BB(I,J1)
      CODE(MM)=2
      IR(MM,I)=1
      GO TO 100
   60 MM=MM+1
   70 IF(BB(I,J1).EQ.1.01) 80,90
   80 B(MM)=0.
      GO TO 95
   90 B(MM,1)=BB(I,J1)
   95 CODE(MM)=1
      IR(MM,I)=1
      IF(BB(I,J2).EQ.-1.01) GO TO 100
      B(MM+1,1)=-BB(I,J2)
      CODE(MM+1)=0
      MM=MM+1
      IR(MM,I)=1
  100 CONTINUE
      LL=MM
      RETURN
      END
C
C
$EMA(IJK,3)
      SUBROUTINE US2(NA,NV,LL,K,B,CODE,BB,IR)
C     ****************************************
C
C SUBRUTINA QUE EXPRESA LAS RESTRICCIONES DIMENSIONALES DEL USUARIO EN
C EL PROBLEMA NO LINEAL
C
      ENA BB(60,40),IR(240,120)
      INTECER CODE(240)
      DIMENSI3N B(240,3)
      MM=LL
C
C EN LA MATRIZ DE INCIDENCIA EL TERMINO POSITIVO INDICA LA COTA INFERIOR
C DE LA VARIABLE Y EL TERMINO NEGATIVO LA COTA INFERIOR DE LA SUPERFICIE
C DE LA ESTANCIA.
C
      DO 100 I=1,NA
      J1=0
      J2=0
      DO 50 J=1,NV
      IF(BB(I,J).EQ.0) GO TO 50
      IF(BB(I,J).GT.0) J1=J
      IF(BB(I,J).LT.0) J2=J
   50 CONTINUE
      MM=MM+1
C
C SI UNA VARIABLE NO POSEE COTA INFERIOR SE HACE ESTA IGUAL A CERO.
```

```
C
      IF(BB(I,J1).EQ.1.01) B(MM,1)=0.
      IF(BB(I,J1).NE.1.01) B(MM,1)=BB(I,J1)
      CODE(MM)=2
      IR(MM,I+(K-1)*NA)=1
      IF(K.EQ.2) GO TO 100
C
C SI UNA ESTANCIA NO TIENE AREA MINIMA SE HACE ESTA IGUAL A CERO.
C
      IF(BB(I,J2).EQ.-1.01) GO TO 100
      B(MM+1,1)=-BB(I,J2)
      CODE(MM+1)=3
      MM=MM+1
      IR(MM,I)=1
      IR(MM,I+NA)=1
  100 CONTINUE
      LL=MM
      RETURN
      END
C
C
      FUNCTION NER(ISAL,IERR)
C     ***********************
C
C FUNCION QUE DETERMINA EL TIPO DE ERROR
C
      WRITE(ISAL,3001) IERR
      NER=IERR
      RETURN
C
C FORMATOS
C
 3001 FORMAT(//'ERROR ................ 'I3)
      END
C
C
      SUBROUTINE LIS(ISAL,K,M,IJ,NET,NCET,NLET,NTYPE,CODE,B,TITLE,
     &C,XB,IR,A,ZETA,INDS)
C     **********************************************************
C
C SUBRUTINA QUE SOLUCIONA EL PROBLEMA DE PROGRAMACION LINEAL, UTLIZANDO
C EL ALGORITMO SIMPLEX.
C
      INTEGER CODE(240),XB(120),C1(120),BASICS,OPTSOL
      REAL TITLE(20)
      DIMENSION B(240,3),C(60),ZETA(120),B1(120),C2(60)
      ENA IR(240,60),A(120,60)
      DO 10 I=1,120
      XB(I)=0
      C1(I)=0
      B1(I)=0
   10 CONTINUE
      DO 20 I=1,K
      C2(I)=C(I)
   20 CONTINUE
      DO 100 I=M,IJ,-1
      B1(I-IJ+1)=B(I,1)
      C1(I-IJ+1)=CODE(I)
      DO 50 J=K,1,-1
      A(I-IJ+1,J)=IR(I,J)
   50 CONTINUE
```

```
 100 CONTINUE
     M1=M-IJ+1
C
C  ** LLAMADA A LA SUBRUTINA QUE PROPORCIONA VARIABLES DE HOLGURA, DE -
C     FECTO Y ARTIFICIALES CUANDO SEAN NECESARIAS **
C
      CALL SSA(KP1,MP1,K,M1,NGET,NLET,NET,NP1,N,NC1,NC,INDEXG,INDEXL,
     &INDEXE,NTYPE,SUM,C2,XB,C1,B1,A)
      BASICS=0
      OPTSOL=0
C
C  ** LLAMADA A LA SUBRUTINA QUE EFECTUA EL ALGORITMO SIMPLEX **
C
      CALL SIM(ISAL,BASICS,M1,K,SUM,OPTSOL,N,NP1,MP1,NC1,C2,XB,A,ZETA,
     &INDS)
      IF(NTYPE.EQ.1) GO TO 220
      SUN=-SUM
  220 RETURN
      END
C
C
      SUBROUTINE SSA(KP1,MP1,K,M,NGET,NLET,NET,NP1,N,NC1,NC,INDEXG,
     &INDEXL,INDEXE,NTYPE,SUM,C,XB,CODE,B,A)
      ************************************************************
C
C
C SUBRUTINA QUE PROPORCIONA VARIABLES DE HOLGURA, DEFECTOS Y ARTIFICIA-
C LES CUANDO SEAN NECESARIAS.
C
      INTEGER CODE(120),XB(120)
      DIMENSION B(120),C(60),ARTV(120)
      ENA A(120,60)
C
C INICIALIZACION DE VARIABLES
C
      DO 1 I=1,120
      ARTV(I)=0
    1 CONTINUE
      IA=1
      KP1=K+1
      MP1=M+1
      N=K+2*NGET+NLET+NET
      NP1=N+1
      NC=K+NGET+1
      NC1=NC+NLET
      INDEXG=<+1
      INDEXL=K+NGET+1
      INDEXE=K+NGET+NLET+1
      DO 69 I=1,MP1
      DO 69 J=KP1,NP1
   69 A(I,J)=0.
      DO 5 I=1,M
    5 A(I,NP1)=B(I)
      DO 4 I=1,M
      IF(CODE(I).EQ.0) GO TO 6
      IF(CODE(I).EQ.1) GO TO 8
      ARTV(IA)=I
      IA=IA+1
      XB(I)=INDEXE
      A(I,INDEXE)=1.
      INDEXE=INDEXE+1
```

```
      GO TO 4
    8 XB(I)=INDEXE
      ARTV(IA)=I
      IA=IA+1
      INDEXE=INDEXE+1
      A(I,INDEXG)=-1.
      INDEXG=INDEXG+1
      GO TO 4
    6 XB(I)=INDEXL
      A(I,INDEXL)=1.
      INDEXL=INDEXL+1
    4 CONTINUE
C
C COMPROBAR MAXIMIZACION
C
      IF(MTYPE.EQ.0) GO TO 12
      DO 60 J=1,K
   60 A(MP1,J)=-C(J)
      GO TO 50
   12 DO 55 J=1,K
   55 A(MP1,J)=C(J)
   50 DO 61 J=KP1,NP1
      A(MP1,J)=0.
   61 C(J)=0.
      DO 62 J=1,K
   62 C(J)=-A(MP1,J)
      DO 63 J=NC1,N
   63 C(J)=-10.E2
      IF(NGET+NET.EQ.0) RETURN
      IA=IA-1
      KPGTE=K+NGET
      DO 64 J=1,KPGTE
      SUM=0.
      DO 65 I=1,IA
      QQQQ=ARTV(I)
   65 SUM=SUM+A(QQQQ,J)
   64 A(MP1,J)=A(MP1,J)-10.E2*SUM
      SUM=0.
      DO 66 I=1,IA
      QQQQ=ARTV(I)
   66 SUM=SUM+A(QQQQ,NP1)
      A(MP1,NP1)=A(MP1,NP1)-10.E2*SUM
      RETURN
      END
C
C
      SUBROUTINE SIM(ISAL,BASICS,M,K,SUM,OPTSOL,N,MP1,NP1,NC1,C,XB,A,
     &ZETA,KRS)
C     ***************************************************************
C
C SUBRUTINA QUE REALIZA EL ALGORITMO SIMPLEX.
C
      INTEGER XB(120),BASICS,OPTSOL
      DIMENSION C(60),ZETA(120)
      EMA A(120,60)
C
C PASO 1: OBTENER UNA SOLUCION BASICA ADMISIBLE.
C
  100 BASICS=BASICS+1
      GO TO 200
  105 SUM=0.
```

```
      DO 112 I=1,M
      IF(KB(I).LE.K) ZETA(KB(I)+KRS*K)=A(I,NP1)
      PPP=KB(I)
  112 SUM=SUM+C(PPP)*A(I,NP1)
      IF(OPTSOL.EQ.1)GO TO 920
C
C PASO 2,3: ESCOGER LA VARIABLE NO BASICA DE COSTO RELATIVO MAS NEGATIVO
C PARA INTRODUCIRLA EN LA BASE. SI NINGUNO ES NEGATIVO, SE HA ALCANZADO
C EL OPTIMO.
C IR AL PASO 9
C
  200 NEG=0
      GNEG=0.
      DO 210 J=1,N
      IF(A(MP1,J).GE.GNEG) GO TO 210
      GNEG=A(MP1,J)
      NEG=J
  210 CONTINUE
      IF(NEG.EQ.0) GO TO 900
C
C PASO 4: COMPROBAR SI LA FUNCION OBJETIVO ES NO ACOTADA.
C
  400 SPR=10.E10
      DO 410 I=1,M
      IF(A(I,NEG).LE.0.00001) GO TO 410
      IF(A(I,NP1)/A(I,NEG).GE.SPR) GO TO 410
      SPR=A(I,NP1)/A(I,NEG)
      NSPR=I
  410 CONTINUE
      IF(SPR.LE.10.E8) GO TO 510
      IERR=NER(ISAL,3)
      STOP 0003
C
C PASO 5: PIVOTEAR
C
  510 PELE=A(NSPR,NEG)
      DO 500 J=1,NP1
  500 A(NSPR,J)=A(NSPR,J)/PELE
      KB(NSPR)=NEG
C
C PASOS 6,7,8: REALIZAR TRASFORMACIONES ELEMENTALES E IR AL PASO 1 PARA
C IMPRIMIR LA NUEVA SOLUCION BASICA ADMISIBLE.
C
  600 DO 610 I=1,MP1
      IF(I.EQ.NSPR) GO TO 610
      HOLD=A(I,NEG)
      DO 620 J=1,NP1
  620 A(I,J)=A(I,J)-HOLD*A(NSPR,J)
  610 CONTINUE
      GO TO 100
C
C PASOS 9,10: SI UNA VARIABLE ARTIFICIAL ES POSITIVA NO EXISTE SOLUCION
C ADMISIBLE. EN CASO CONTRARIO SE HA OBTENIDO EL OPTIMO.
C
  900 OPTSOL=1.
      GO TO 105
  920 DO 930 I=1,M
      IF(KB(I).LT.NC1) GO TO 930
      IF(A(I,MP1).LE.0) GO TO 930
      IERR=NER(ISAL,4)
```

```
      STOP 0004
  930 CONTINUE
      KRS=1
      RETURN
C
C FORMATOS
C
      END
C
C
$EMA(IJK,3)
      SUBROUTINE POL(ISAL,N,K,C,CODE,B,ZETA,IR)
C     ************************************
C
C SUBRUTINA QUE RESUELVE EL PROBLEMA DE PROGRAMACION NO LINEAL UTILIZAND
C EL ALGORITMO DE POLAK-RIVIERE.
C ESTE ALGORITMO PENALIZA LA FUNCION OBJETIVO CON UN PENALTY EXTERIOR Y
C OTRO INTERIOR Y UTILIZA EL METODO DEL GRADIENTE CONJUGADO.
C
      INTEGER CODE(240)
      DIMENSION C(60),B(240,3),ZETA(120),GF(120),GP(120),GPP(120)
      DIMENSION GF1(120),G(120),H(120),Z(120)
      EMA IR(240,120)
      JJB=0
   10 NN=0
C
C VALORES DE LOS DISTINTOS FACTORES:
C CAMPO DE ACCION RECOMENDABLE:
C                                     ONE (0,2.5)
C                                     BETA(0.5,0.8)
C                                     EX1 0.05
C                                     EX2 0.0025
C                                     EXI (0.5,1)
C
      ONE=1.
      BETA=0.08
      EX1=0.05
      EX2=0.025
      EXI=0.7
      SN=0.
C
C PASO 0: SE TOMA COMO VALORES INICIALES LA COTA INFERIOR DE CADA UNA DE
C LAS VARIABLES. PARA LA LONGITUD Y ANCHURA DEL CONTORNO, SI NO SON DA-
C TOS, SE TOMA LA RAIZ CUADRADA DE LA SUMA DE LOS CUADRADOS DE LOS VALO-
C RES INICIALES DE LAS RESTANTES VARIABLES.
C
      DO 150 I=1,K
      IF(CODE(I).NE.2) GO TO 140
      DO 100 J=1,2*N
      IF(IR(I,J).NE.0) GO TO 110
  100 CONTINUE
  110 ZETA(J)=B(I,1)
  140 IF(CODE(I).NE.5) GO TO 150
      SN=B(I,1)
  150 CONTINUE
      IF(SN.NE.0) GO TO 210
      DO 200 I=1,N-1
      SN=SN+ZETA(I)*ZETA(I+N)
  200 CONTINUE
  210 ZETA(N)=SQRT(SN)
      ZETA(2*N)=SQRT(SN)
```

```
      KONT=0
C
C PASO 1: ITERACIONES PARA AJUSTAR LA SOLUCION OPTIMA.
C
      DO 850 L=0,10
      JONT=0
C
C PASO 2: DIVISION DE LAS RESTRICCIONES UNA VEZ SUSTITUIDAS LAS VARIA-
C BLES EN ELLAS POR LOS VALORES INICIALES (EN RESTRICCIONES MAYORES O
C IGUALES QUE CERO Y RESTRICCIONES MENORES QUE CERO). LAS PRIMERAS SE
C PENALIZAN EXTERIORMENTE Y LAS SEGUNDAS INTERIORMENTE.
C
      DO 300 I=1,K
      B(I,2)=B(I,1)
      IF(CODE(I).EQ.5) GO TO 270
      DO 250 J=1,2*N
      IF(IR(I,J).EQ.0) GO TO 250
      IF(CODE(I).NE.1.AND.CODE(I).NE.4) GO TO 260
      B(I,2)=B(I,2)-IR(I,J)*ZETA(J)
  250 CONTINUE
      GO TO 300
  260 IF(CODE(I).EQ.2) B(I,2)=B(I,2)-ZETA(J)
      IF(CODE(I).EQ.3) B(I,2)=B(I,2)-ZETA(J)*ZETA(J+N)
      IF(CODE(I).NE.5) GO TO 300
  270 DO 280 JJ=1,N-1
      B(I,2)=B(I,2)-ZETA(JJ)*ZETA(JJ+N)
  280 CONTINUE
  300 CONTINUE
C
C PASO 3:
C  ** LLAMADA A LA SUBRUTINA QUE OBTIENE LOS VALORES DE LA FUNCION OBJE-
C     TIVO Y FUNCIONES PENALTIS **
C
  310 CALL FUN(N,2*N,K,F,P,PP,CODE,C,ZETA,B,IR)
C
C PASO 4:
C  ** LLAMADA A LA SUBRUTINA QUE OBTIENE EL GRADIENTE DE LA FUNCION OB-
C     JETIVO **
C
      CALL GR1(N,2*N,GF,C,ZETA)
C
C PASO 5:
C  ** LLAMADA A LA SUBRUTINA QUE OBTIENE LOS GRADIENTES DE LAS FUNCIONES
C     PENALTIS **
C
      CALL GR2(N,2*N,K,CODE,GP,GPP,ZETA,B,IR)
C
C PASO 6: DETERMINACION DEL VALOR DE LA FUNCION OBJETIVO PENALIZADA.
C
      F1=F+P/EX1+EX2*PP
C
C PASO 7: DETERMINACION DEL GRADIENTE DE LA FUNCION PENALIZADA.
C
      DO 400 J=1,2*N
      GF1(J)=GF(J)+GP(J)/EX1+EX2*GPP(J)
  400 CONTINUE
      IF(JONT.EQ.1) GO TO 660
  401 DO 405 J=1,2*N
      G(J)=-GF1(J)
      H(J)=-GF1(J)
```

```
  405 CONTINUE
      HN=0
C
C PASO 8: DETERMINACION DE LA MINIMA LONGITUD DEL PASO.
C
  410 HN=0
      K=0.
  420 DO 450 J=1,2*N
      Z(J)=ZETA(J)+K*H(J)
  450 CONTINUE
C
C  ** LLAMADA A LA SUBRUTINA QUE OBTIENE EL VALOR DE LA FUNCION OBJETIVO
C     Y LAS FUNCIONES PENALTIS **
C
      CALL FUN(N,2*N,K,F,P,PP,CODE,C,Z,B,IR)
      TET=F+P/EK1+EK2*PP-F1
C
C  ** LLAMADA A LA SUBRUTINA QUE OBTIENE EL GRADIENTE DE LA FUNCION OBJE
C
      CALL GR1(N,2*N,GF,C,Z)
C
C  ** LLAMADA A LA SUBRUTINA QUE OBTIENE LOS GRADIENTES DE LAS FUNCIONES
C     PENALTIS **
C
      CALL GR2(N,2*N,K,CODE,GP,GPP,Z,B,IR)
      TETT=0
      DO 500 J=1,2*N
      GF(J)=GF(J)+GP(J)/EK1+EK2*GPP(J)
      TETT=TETT+GF(J)*H(J)
  500 CONTINUE
      IF(TETT.EQ.0.) GO TO 580
      TAND=1.
  510 DO 550 J=1,2*N
      Z(J)=ZETA(J)+(K-TAND*TETT)*H(J)
  550 CONTINUE
C
C  ** LLAMADA A LA SUBRUTINA QUE OBTIENE EL VALOR DE LA FUNCION OBJETIVO
C     Y DE LAS FUNCIONES PENALTIS **
C
      CALL FUN(N,2*N,K,F,P,PP,CODE,C,Z,B,IR)
      F2=F+P/EK1+EK2*PP
      TET1=F2-F1
      DELTA=TET1-TET+TAND*TETT**2/2.
      IF(DELTA.LE.0.001) GO TO 560
      TAND=BETA*TAND
      GO TO 510
  560 HN=HN+1
      IF(HN.GE.2) GO TO 570
      K=K-TAND*TETT
      GO TO 420
  570 UN=K-TAND*TETT
      GO TO 590
  580 UN=K
C
C PASO 9: DETERMINACION DE LOS NUEVOS VALORES DE LAS VARIABLES.
C
  590 DO 600 J=1,2*N
      Z(J)=ZETA(J)
      ZETA(J)=ZETA(J)+UN*H(J)
  600 CONTINUE
      KONT=KONT+1
```

```
      JONT=1
      GO TO 310
C
C PASO 10: DETERMINACION DE LA NORMA DEL NUEVO GRADIENTE DE LA FUNCION
C OBJETIVO PENALIZADA
C
C
  650 RGF=0
      DO 700 J=1,2*N
      RGF=RGF+GF1(J)**2
  700 CONTINUE
      RGF=SQRT(RGF)
      IF(RGF.GT.EXI) 710,860
  710 NN=NN+1
      IF(ISSW(0).LT.0) WRITE(ISAL,3001) JJS,NN
 3001 FORMAT(2I3)
C
C PASO 11: COMPROBACION SI SE HA REALIZADO TANTOS CICLOS COMO VARIABLES.
C
      IF(NN.LT.2*N) 719,715
C
C PASO 12: GRADIENTE CONJUGADO.
C
  715 JJS=JJS+1
      IF(JJS.EQ.2) 860,401
  719 RG=0.
      RGF=0.
      DO 720 J=1,2*N
      RG=RG+G(J)**2
      RGF=RGF+GF1(J)**2
  720 CONTINUE
      IF(RGF.LE.RG) GO TO 740
      DO 730 J=1,2*N
      ZETA(J)=ONE*ZETA(J)+(1-ONE)*Z(J)
  730 CONTINUE
  740 RG=0
      ESC=0
      DO 750 J=1,2*N
      ESC=ESC+(GF1(J)+G(J))*GF1(J)
      RG=RG+G(J)**2
      G(J)=-GF1(J)
  750 CONTINUE
      ESC=ESC/RG
      DO 800 J=1,2*N
      H(J)=G(J)+ESC*H(J)
  800 CONTINUE
      GO TO 410
  850 CONTINUE
  860 RETURN
      END
C
C
$EMA(IJK,3)
      SUBROUTINE FUN(N,NN,K,F,P,PP,CODE,C,Z,B,IR)
C     ********************************************
C
C SUBRUTINA QUE OBTIENE LOS VALORES DE LAS FUNCIONES OBJETIVO, PENALTY
C EXTERIOR Y PENALTY INTERIOR.
C
      INTEGER CODE(K)
```

```
      DIMENSION C(N),Z(NN),B(240,3)
      EMA IR(240,120)
      F=0
      DO 50 J=1,N
      F=F+C(J)*Z(J)*Z(J+N)
   50 CONTINUE
      P=0
      PP=0
      DO 200 I=1,K
      B(I,3)=B(I,1)
      IF(CODE(I).EQ.5) GO TO 120
      DO 100 J=1,NN
      IF(IR(I,J).EQ.0) GO TO 100
      IF(CODE(I).NE.1.AND.CODE(I).NE.4) GO TO 110
      B(I,3)=B(I,3)-IR(I,J)*Z(J)
  100 CONTINUE
      GO TO 200
  110 IF(CODE(I).EQ.2) B(I,3)=B(I,3)-Z(J)
      IF(CODE(I).EQ.3) B(I,3)=B(I,3)-Z(J)*Z(J+N)
      IF(CODE(I).NE.5) GO TO 200
  120 DO 150 JJ=1,N-1
      B(I,3)=B(I,3)-Z(JJ)*Z(JJ+N)
  150 CONTINUE
  200 CONTINUE
      DO 250 I=1,K
      IF(CODE(I).LE.3.AND.B(I,3).LT.0) GO TO 210
      IF(CODE(I).GT.3.OR.B(I,2).GE.0) P=P+B(I,3)*B(I,3)
  210 IF(CODE(I).LE.3.AND.B(I,2).LT.0) PP=PP-1./B(I,3)
  250 CONTINUE
      RETURN
      END
C
C
      SUBROUTINE GR1(N,NN,GF,C,Z)
C     **************************
C
C SUBRUTINA QUE OBTIENE EL GRADIENTE DE LA FUNCION OBJETIVO
C
      DIMENSION C(N),GF(NN),Z(NN)
      DO 50 J=1,NN
      GF(J)=0.
      IF(J.LE.N) GF(J)=C(J)*Z(J+N)
      IF(J.GT.N) GF(J)=C(J-N)*Z(J-N)
   50 CONTINUE
      RETURN
      END
C
C
$EMA(IJK,3)
      SUBROUTINE GR2(N,NN,K,CODE,GP,GPP,Z,B,IR)
C     *****************************************
C
C SUBRUTINA QUE OBTIENE LOS GRADIENTES DE LA FUNCION PENALTY EXTERIOR E
C INTERIOR
C
      INTEGER CODE(K)
      DIMENSION Z(NN),B(240,3),GP(NN),GPP(NN)
      EMA IR(240,120)
      DO 100 J=1,NN
      GP(J)=0
      GPP(J)=0
```

```
      DO 50 I=1,K
      IF(IR(I,J).EQ.0) GO TO 50
      IF(CODE(I).EQ.3.OR.CODE(I).EQ.5) GO TO 20
      IF(CODE(I).LE.3.AND.B(I,3).LT.0) GO TO 10
      IF(CODE(I).GT.3.OR.B(I,2).GE.0) GP(J)=GP(J)-B(I,3)*IR(I,J)
   10 IF(CODE(I).LE.3.AND.B(I,2).LT.0) GPP(J)=GPP(J)-IR(I,J)/B(I,3)**2
      GO TO 30
   20 IF(CODE(I).EQ.3.AND.B(I,3).LT.0) GO TO 30
      IF(CODE(I).GT.3.OR.B(I,2).GE.0.AND.J.LE.N) GP(J)=GP(J)-B(I,3)*Z(J+
     &N)
      IF(CODE(I).GT.3.OR.B(I,2).GE.0.AND.J.GT.N) GP(J)=GP(J)-B(I,3)*Z(J-
     &N)
   30 IF(CODE(I).LE.3.AND.B(I,2).LT.0.AND.J.LE.N) GPP(J)=GPP(J)-Z(J+N)/B
     &(I,3)**2
      IF(CODE(I).LE.3.AND.B(I,2).LT.0.AND.J.GT.N) GPP(J)=GPP(J)-Z(J-N)/B
     &(I,3)**2
   50 CONTINUE
      GP(J)=2.*GP(J)
  100 CONTINUE
      RETURN
      END
C
C
      SUBROUTINE AJU(N,K,NET,NGET,NLET,ITIP,LL,CODE,B,NOM,C,XB,Z,INDS,
     &IR,JAUX)
C     ***************************************************************
C
C SUBRUTINA PARA AJUSTAR LOS VALORES OBTENIDOS POR PROGRAMACION LINEAL
C APLICANDO EL ALGORITMO SIMPLEX A PARTIR DE LOS RESULTADOS OBTENIDOS
C
      EMA IR(240,120)
      INTEGER CODE(240),NOM(40),XB(120),JAUX(240)
      DIMENSION Z(120),C(60),B(240,3)
C
C VAMOS A ELIMINAR DE LA MATRIZ DE RESTRICCIONES LAS NO LINEALES
C
      DO 200 I=1,K
      IF(CODE(I).NE.3.AND.CODE(I).NE.5) GO TO 200
      DO 100 J=I,K
      DO 50 JJ=1,2*N
      IR(J,JJ)=IR(J+1,JJ)
   50 CONTINUE
      B(J,1)=B(J+1,1)
      CODE(J)=CODE(J+1)
  100 CONTINUE
      DO 150 J=1,2*N
      IR(K,J)=0
  150 CONTINUE
      B(K,1)=0
      CODE(K)=0
      K=K-1
  200 CONTINUE
C
C SE AGRUPA LA MATRIZ DE RESTRICCIONES
C
      LL=0
      DO 350 I=1,K
      KT=0
      DO 250 J=1,N
      IF(IR(I,J).EQ.0) GO TO 250
```

```
      KT=KT+1
  250 CONTINUE
      IF(KT.NE.0) GO TO 350
      IF(LL.EQ.0) LL=I
      DO 300 J=1,N
      IR(I,J)=IR(I,J+N)
      IR(I,J+N)=0
  300 CONTINUE
  350 CONTINUE
C
C SE ACOTAN LAS VARIABLES DE NUEVO DE ACUERDO A LOS RESULTADOS OBTENIDOS
C
      DO 500 I=1,K
      IF(CODE(I).NE.2) GO TO 500
      DO 450 J=1,N
      IF(IR(I,J).NE.0) GO TO 460
  450 CONTINUE
  460 IF(LL.LE.I) GO TO 470
      IF(Z(J).GT.B(I,1)) B(I,1)=Z(J)
      GO TO 500
  470 IF(Z(J+N).GT.B(I,1)) B(I,1)=Z(J+N)
  500 CONTINUE
      ITIP=0
      NET=0
      NLET=0
      NGET=0
      INDS=0
C
C TRASFORMACION DE LOS CODIGOS DE LAS RESTRICCIONES PARA A AJUSTARLOS
C A UNO DE LOS TIPOS DE LA PROGRAMACION LINEAL
C
      DO 550 I=1,K
      IF(CODE(I).EQ.2) CODE(I)=1
      IF(CODE(I).EQ.4) CODE(I)=2
  550 CONTINUE
      DO 600 I=1,LL-1
      IF(CODE(I).EQ.1) NGET=NGET+1
      IF(CODE(I).EQ.2) NET=NET+1
  600 CONTINUE
C
C  ** LLAMADA A LA SUBRUTINA QUE APLICA EL ALGORITMO DE PROGRAMACION LI-
C     NEAL **
C
      CALL LIS(ISAL,N,LL-1,1,NET,NGET,NLET,ITIP,CODE,B,NOM,C,XB,
     &IR(1,1),IR(1,61),Z,INDS)
      INDS=1
      ITIP=0
      NLET=0
      NET=0
      NGET=0
      DO 650 I=LL,K
      IF(CODE(I).EQ.1) NGET=NGET+1
      IF(CODE(I).EQ.2) NET=NET+1
  650 CONTINUE
C
C  ** LLAMADA A LA SUBRUTINA QUE APLICA EL ALGORITMO DE PROGRAMACION LI-
C     NEAL **
C
      CALL LIS(ISAL,N,K,LL,NET,NGET,NLET,ITIP,CODE,B,NOM,C,XB,IR(1,1),
     &IR(1,61),Z,INDS)
      RETURN
```

```
      END                                                               AJU-296
C                                                                       ESQ-297
C                                                                       ESQ-298
      SUBROUTINE ESQ(ISAL,NA,NVH,NVV,XL,YL,NUNE,IPIL,ISTA,BA,BB,X,Y)    ESQ-299
C     **************************************************************    ESQ-300
C                                                                       ESQ-301
C SUBRUTINA QUE TRAZA EL ESQUEMA YA DIMENSIONADO                        ESQ-302
C                                                                       ESQ-303
      ENA BB(60,40),BA(60,40)                                           ESQ-304
      INTEGER VI,VF,VE,VA,WI,WF,WE,WA                                   ESQ-305
      DIMENSION X(60),Y(60),XL(60),YL(60),NUNE(60),ISTA(60),IPIL(60)    ESQ-306
      DIMENSION XG(60),YG(60),IAUX(60),JAUX(60)                         ESQ-307
C                                                                       ESQ-308
C PUESTA A CERO                                                         ESQ-309
C                                                                       ESQ-310
      DO 10 I=1,NA                                                      ESQ-311
      NUNE(I)=0                                                         ESQ-312
      ISTA(I)=0                                                         ESQ-313
      IAUX(I)=0                                                         ESQ-314
      JAUX(I)=0                                                         ESQ-315
      XG(I)=0                                                           ESQ-316
      YG(I)=0                                                           ESQ-317
   10 CONTINUE                                                          ESQ-318
      DO 20 I=1,NVH                                                     ESQ-319
      IPIL(I)=0                                                         ESQ-320
   20 CONTINUE                                                          ESQ-321
C                                                                       ESQ-322
C HACEMOS TODOS LOS TERMINOS DE LA MATRIZ DE INCIDENCIA HORIZONTAL: 0,1, ESQ-323
C HACEMOS TODOS LOS TERMINOS DE LA MATRIZ DE INCIDENCIA VERTICAL: 0,1,-1 ESQ-324
C                                                                       ESQ-325
   45 DO 100 I=1,NA                                                     ESQ-326
      DO 50 J=1,NVH                                                     ESQ-327
      IF(BB(I,J).EQ.0) GO TO 50                                         ESQ-328
      BB(I,J)=BB(I,J)/ABS(BB(I,J))                                      ESQ-329
   50 CONTINUE                                                          ESQ-330
      DO 80 J=1,NVV                                                     ESQ-331
      IF(BA(I,J).EQ.0) GO TO 80                                         ESQ-332
      BA(I,J)=BA(I,J)/ABS(BA(I,J))                                      ESQ-333
   80 CONTINUE                                                          ESQ-334
  100 CONTINUE                                                          ESQ-335
C                                                                       ESQ-336
C NUMERACION DE LAS ARISTAS Y OBTENCION DE LAS NUEVAS COORDENADAS DEL C. ESQ-337
C                                                                       ESQ-338
      DO 120 J=1,NVV                                                    ESQ-339
      IF(BA(NA,J).EQ.-1) WI=J                                           ESQ-340
      IF(BA(NA,J).EQ.1) WF=J                                            ESQ-341
      IF(WI.NE.0.AND.WF.NE.0) GO TO 130                                 ESQ-342
  120 CONTINUE                                                          ESQ-343
  130 DO 150 J=1,NVH                                                    ESQ-344
      IF(BB(NA,J).EQ.1) VI=J                                            ESQ-345
      IF(BB(NA,J).EQ.-1) VF=J                                           ESQ-346
      IF(VI.NE.0.AND.VF.NE.0) GO TO 160                                 ESQ-347
  150 CONTINUE                                                          ESQ-348
  160 IONT=0                                                            ESQ-349
      KONT=1                                                            ESQ-350
      VE=VI                                                             ESQ-351
      IPIL(KONT)=VE                                                     ESQ-352
      JONT=0                                                            ESQ-353
      IA1=0                                                             ESQ-354
      IA2=0                                                             ESQ-355
```

```
260 II=0
    XMIN=100.
    DO 300 I=1,NA-1
    IF(BB(I,VE).NE.-1.OR.NUME(I).NE.0) GO TO 300
    IF(X(I).GT.XMIN) GO TO 300
    II=I
    XMIN=X(II)
300 CONTINUE
    IF(II.EQ.0) GO TO 440
    DO 320 J=1,NVH
    IF(BB(II,J).EQ.1) GO TO 325
320 CONTINUE
325 VA=J
    DO 330 J=1,NVV
    IF(BA(II,J).EQ.-1) WE=J
    IF(BA(II,J).EQ.1) WA=J
330 CONTINUE
    IF(VE.EQ.VI) IA1=II
    IF(VE.NE.VI) IA1=0
    IF(WA.EQ.WI) IA2=II
    IF(WA.NE.WI) IA2=0
    JONT=JONT+1
    NUME(II)=JONT
    IAUX(II)=IA1
    JAUX(II)=IA2
    IF(JONT.EQ.NA-1) GO TO 450
    IONT=IONT+1
    ISTA(IONT)=II
    VE=VA
    IF(VE.EQ.VF) GO TO 445
    KONT=KONT+1
    IPIL(KONT)=VE
    GO TO 260
440 IPIL(KONT)=0
    KONT=KONT-1
445 VE=IPIL(KONT)
    GO TO 260
450 CALL COP(NA,NVV,X,Y,NUME,IAUX,BA)
    CALL COP(NA,NVH,Y,X,NUME,JAUX,BB)
    DO 520 K=1,10
    DO 500 I=1,NA-1
    DO 490 J=1,NA-1
    IF(NUME(J).NE.I) GO TO 490
    II=IAUX(J)
    IF(II.NE.J) GO TO 460
    IF(YG(J).EQ.0) YG(J)=YL(J)/2.
    GO TO 470
460 IF(YG(II).EQ.0.OR.YG(J).NE.0) GO TO 470
    IF(Y(II).GT.Y(J)) YG(J)=YG(II)-YL(II)/2.-YL(J)/2.
    IF(Y(II).LT.Y(J)) YG(J)=YG(II)+YL(II)/2.+YL(J)/2.
470 JJ=JAUX(J)
    IF(JJ.NE.J) GO TO 480
    IF(XG(J).EQ.0) XG(J)=XL(J)/2.
    GO TO 500
480 IF(XG(JJ).EQ.0.OR.XG(J).NE.0) GO TO 500
    IF(X(JJ).GT.X(J)) XG(J)=XG(JJ)-XL(JJ)/2.-XL(J)/2.
    IF(X(JJ).LT.X(J)) XG(J)=XG(JJ)+XL(JJ)/2.+XL(J)/2.
490 CONTINUE
500 CONTINUE
    DO 510 I=1,NA-1
    IF(XG(I).EQ.0.OR.YG(I).EQ.0) GO TO 520
```

```
  510 CONTINUE
      GO TO 530
  520 CONTINUE
  530 WRITE(ISAL,3001)
      CALL PLTLU(10)
      CALL PLOT(0.,0.,-3)
      DO 600 I=1,NA-1
      RI=I+4
      CALL NUMB(XG(I),YG(I),0.15,RI,0.,-1)
      Z1=XG(I)+XL(I)/2.
      Z2=YG(I)+YL(I)/2.
      Z3=XG(I)-XL(I)/2.
      Z4=YG(I)-YL(I)/2.
      CALL PLOT(Z1,Z2,3)
      CALL PLOT(Z3,Z2,2)
      CALL PLOT(Z3,Z4,2)
      CALL PLOT(Z1,Z4,2)
      CALL PLOT(Z1,Z2,2)
  600 CONTINUE
      CALL PLOT (0.,0.,3)
      STOP
C
C FORMATOS
C
 3001 FORMAT(//1X"MENSAJE ::SR:: SE DIBUJA EN EL PLOTTER")
      END
C
C
      SUBROUTINE COP(NA,NV,X,Y,NUME,IAUX,BA)
C     **************************************
C
C     SUBRUTINA PARA COMPLETAR PARAMETROS PARA EL TRAZADO POSTERIOR
C
      DIMENSION BA(60,40)
      DIMENSION X(60),Y(60),IAUX(60),NUME(60)
      DO 250 K=1,10
      DO 230 L=1,NA-1
      DO 200 I=1,NA-1
      IF(NUME(I).NE.L) GO TO 200
      IF(IAUX(I).NE.0) GO TO 230
      DO 50 J=1,NV
      IF(BA(I,J).EQ.1) VA=J
      IF(BA(I,J).EQ.-1) VE=J
   50 CONTINUE
      KK=0
      DMIN=10000.
      DO 100 J=1,NA-1
      IF(J.EQ.I) GO TO 100
      IF(Y(J).GT.Y(I)) GO TO 100
      IF(BA(J,VA).NE.1.AND.BA(J,VE).NE.-1) GO TO 100
      IF((X(J)-X(I))**2+(Y(J)-Y(I))**2.GE.DMIN) GO TO 100
      KK=J
      DMIN=(X(J)-X(I))**2+(Y(J)-Y(I))**2
  100 CONTINUE
      IF(KK.NE.0) GO TO 180
      DO 150 J=1,NA-1
      IF(J.EQ.I) GO TO 150
      IF(BA(J,VA).NE.-1.AND.BA(J,VE).NE.1) GO TO 150
      IF((X(J)-X(I))**2+(Y(J)-Y(I))**2.GE.DMIN) GO TO 150
      IF(IAUX(J).EQ.0.OR.IAUX(J).EQ.I) GO TO 150
```

```
      IF(Y(IAUX(J)).GT.Y(I)) GO TO 150
      KK=J
      DMIN=(X(J)-X(I))**2+(Y(J)-Y(I))**2
  150 CONTINUE
      IAUX(I)=IAUX(KK)
      GO TO 230
  180 IAUX(I)=KK
      GO TO 230
  200 CONTINUE
  230 CONTINUE
      DO 240 I=1,NA-1
      IF(IAUX(I).EQ.0) GO TO 250
  240 CONTINUE
      GO TO 260
  250 CONTINUE
  260 RETURN
      END
$
```

A.II.2.4. Ejemplos

```
MENSAJE ::ED:: TITULA EL PROBLEMA

EJEMPLO 1 (DIMENSIONADO)
********************************************************************************

MENSAJE ::ED:: NUMERO DE VERTICES VERTICALES Y HORIZONTALES

NUMERO DE VERTICES VERTICALES ....  6
NUMERO DE VERTICES HORIZONTALES ..  8

MENSAJE ::ED:: COORDENADAS DEL CENTRO GEOMETRICO DE LOS ESPACIOS.

               SE HARA A PARTIR DEL ESQUEMA ADIMENSIONAL,DANDO UNAS DIMENSIONES
               ARBITRARIAS A CADA ESPACIO. ESTOS SE NUMERARAN A PARTIR DEL 5.
               AL FINAL DE LA LISTA INTRODUCE: 0

ESP      XG     YG
---      --     --
  5    8.00  11.00
  6    7.50   8.00
  7    3.00   9.00
  8    9.50   7.00
  9    9.50   9.00
 10    5.00   4.50
 11    2.00   5.00
 12    2.00   2.00
 13    8.00   5.00
 14    8.00   2.00
 15    4.50   1.50
 16    5.50   1.50

MENSAJE ::ED:: CONDICIONES DE ACCESIBILIDAD ENTRE LOCALES

               SE FIJA UN LOCAL Y SE INDICA CUALES DEBEN SER ACCESIBLES A EL.
               AL FINAL DE LA LISTA INTRODUCE: 0

 L  L1  L2  L3  L4  L5
 -  --  --  --  --  --
 6   9   8
10   7  11  12  14  13
 5   6   7

MENSAJE ::ED:: TIPO DE PROBLEMA:  1 .. LINEAL / 2 .. NO LINEAL


PROBLEMA NO LINEAL

MENSAJE ::ED:: TIPO DE FUNCION OBJETIVO:

               PROBLEMA LINEAL ...... 0: MINIMIZACION DE LA LONGITUD Y ANCHURA
                                      1: MINIMIZACION DE UNA EXPRESION
               PROBLEMA NO LINEAL ... 0: MINIMIZACION DE LA SUPERFICIE
                                      1: MINIMIZACION DE UNA EXPRESION
```

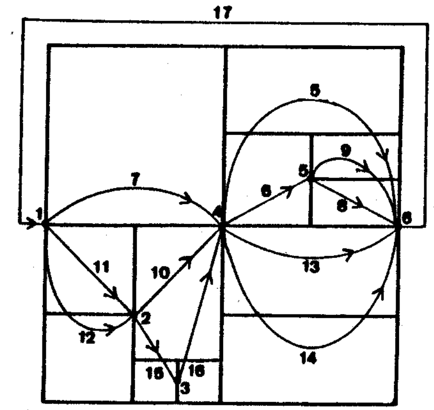
FUNCION OBJETIVO: MINIMIZACION DE LA SUPERFICIE

MENSAJE ::ED:: MATRIZ DE INCIDENCIA VERTICAL

LA NUMERACION DE LOS ESPACIOS SE HARA A PARTIR DEL 5.
LA ARISTA QUE REPRESENTA AL CONTORNO SE NUMERARA EN ULTIMO LUGAR.

SE RECOGEN AQUI LOS REQUISITOS METRICO-GEOMETRICOS.
LOS DATOS SE DAN EN EL SIGUIENTE ORDEN:
- ARISTA
- VERTICE ORIGEN
- VERTICE EXTREMO
- COTA INFERIOR DE LA ANCHURA
- COTA INFERIOR DE LA SUPERFICIE

A	VO	VE	CI	CI
-	--	--	--	--
5	4	6	1.1	1.5
6	4	5	2.0	6.0
7	1	4	3.0	12.0
8	5	6	1.0	0.0
9	5	6	1.0	0.0
10	2	4	1.0	0.0
11	1	2	1.5	0.0
12	1	2	3.0	10.0
13	4	6	2.0	6.0
14	4	6	3.0	10.0
15	2	3	.7	0.0
16	3	4	.7	0.0
17	6	1	0.0	0.0

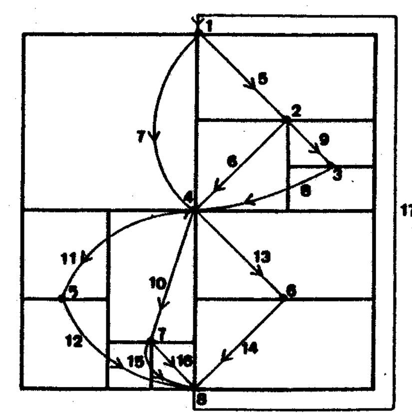


MENSAJE ::ED:: MATRIZ DE INCIDENCIA HORIZONTAL

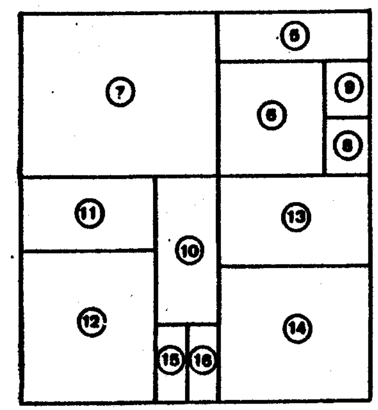
IGUAL CRITERIO DE NUMERACION QUE EN LA VERTICAL.

SE RECOGEN AQUI LOS REQUISITOS METRICO-GEOMETRICOS.
LOS DATOS SE DAN EN EL SIGUIENTE ORDEN:
- ARISTA
- VERTICE ORIGEN
- VERTICE EXTREMO
- COTA INFERIOR DE LA LONGITUD

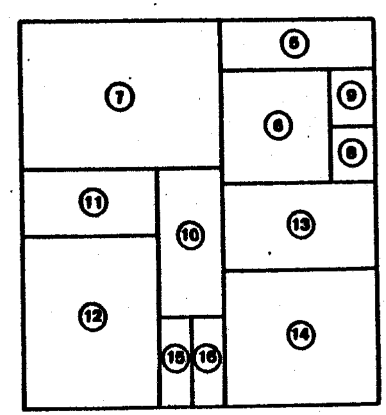
A	VO	VE	CI
-	--	--	--
5	1	2	1.1
6	2	4	2.0
7	1	4	3.0
8	3	4	1.0
9	2	3	1.0
10	4	7	1.0
11	4	5	1.5
12	5	8	3.0
13	4	6	2.0
14	6	8	3.0
15	7	8	.7



```
16   7   8   .7
17   0   1  0.0

-------- LOS RESULTADOS OBTENIDOS PARA CADA ESPACIO Y EL CONTORNO SON: --------

X( 5)=  3.38   Y( 5)=  1.10
X( 6)=  2.38   Y( 6)=  2.50
X( 7)=  4.35   Y( 7)=  3.60
X( 8)=  1.00   Y( 8)=  1.26
X( 9)=  1.00   Y( 9)=  1.24
X(10)=  1.40   Y(10)=  3.29
X(11)=  3.00   Y(11)=  1.63
X(12)=  3.00   Y(12)=  3.38
X(13)=  3.33   Y(13)=  2.00
X(14)=  3.33   Y(14)=  3.00
X(15)=   .70   Y(15)=  1.72
X(16)=   .70   Y(16)=  1.72
X(17)=  7.73   Y(17)=  8.60

------------------------------------------------------------------------------
```

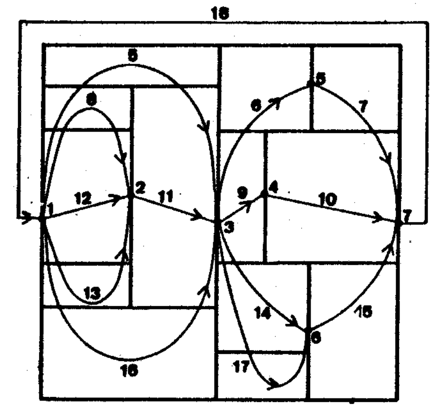


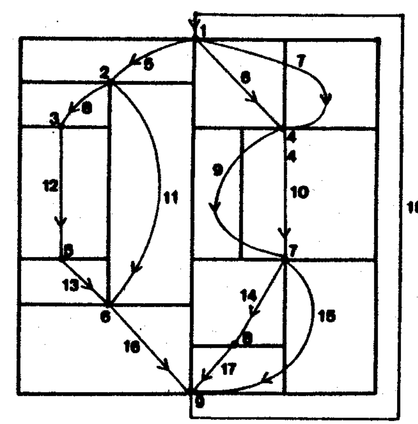
```
MENSAJE !!SR!! SE DIBUJA EN EL PLOTTER

MENSAJE !!ED!! SI SE DESEA AJUSTAR LOS RESULTADOS INTRODUCE: 1
               SI QUIERES TERMINAR .............. INTRODUCE: 0

MENSAJE !!ED!! VALORACION RELATIVA PARA CADA ESPACIO.

               LOS VALORES ASIGNADOS SERAN PROPORCIONALES A LA IMPORTANCIA QUE
               SE ATRIBUYE A LA SUPERFICIE DE CADA ESPACIO.

 V5  V6  V7  V8  V9 V10 V11 V12 V13 V14 V15 V16
 --  --  --  --  -- --- --- --- --- --- --- ---
1.0 5.0 8.0 1.0 1.0 1.0 5.0 9.0 9.0 9.0 5.0 5.0

-------- LOS RESULTADOS OBTENIDOS PARA CADA ESPACIO Y EL CONTORNO SON: --------

X( 5)=  3.38   Y( 5)=  1.10
X( 6)=  2.38   Y( 6)=  2.50
X( 7)=  4.35   Y( 7)=  3.28
X( 8)=  1.00   Y( 8)=  1.26
X( 9)=  1.00   Y( 9)=  1.24
X(10)=  1.40   Y(10)=  3.32
X(11)=  3.00   Y(11)=  1.50
X(12)=  3.00   Y(12)=  3.82
X(13)=  3.33   Y(13)=  2.00
X(14)=  3.33   Y(14)=  3.00
X(15)=   .70   Y(15)=  2.00
X(16)=   .70   Y(16)=  2.00
X(17)=  7.73   Y(17)=  8.60

------------------------------------------------------------------------------
```



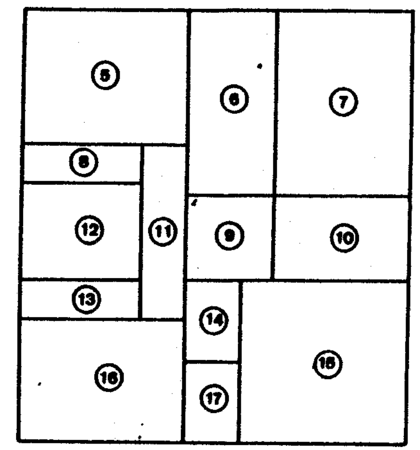
```
MENSAJE ::ED:: TITULA EL PROBLEMA

EJEMPLO 2 (DIMENSIONADO)
*****************************************************************

MENSAJE ::ED:: NUMERO DE VERTICES VERTICALES Y HORIZONTALES

NUMERO DE VERTICES VERTICALES ....  7
NUMERO DE VERTICES HORIZONTALES ..  9

MENSAJE ::ED:: COORDENADAS DEL CENTRO GEOMETRICO DE LOS ESPACIOS.

               SE HAYA A PARTIR DEL ESQUEMA ADIMENSIONAL,DANDO UNAS DIMENSIONES
               ARBITRARIAS A CADA ESPACIO. ESTOS SE NUMERARANA PARTIR DEL 5.
               AL FINAL DE LA LISTA INTRODUCE: 0

ESP       XG      YG
---       --      --
  5     2.00   12.00
  6     6.00   11.00
  7     9.00   11.00
  8     1.00    9.00
  9     5.00    7.00
 10     9.00    6.00
 11     3.00    6.00
 12     1.00    6.00
 13     1.00    3.00
 14     6.00    3.00
 15     9.00    2.00
 16     2.00    1.00
 17     5.00     .50

MENSAJE ::ED:: CONDICIONES DE ACCESIBILIDAD ENTRE LOCALES

               SE FIJA UN LOCAL Y SE INDICA CUALES DEBEN SER ACCESIBLES A EL.
               AL FINAL DE LA LISTA INTRODUCE: 0

 L  L1  L2  L3  L4  L5
 -  --  --  --  --  --
17  14
14  11  15
10   7  15
 7   6
11   5   6  12  16   9

MENSAJE ::ED:: TIPO DE PROBLEMA:  1 .. LINEAL / 2 .. NO LINEAL


PROBLEMA NO LINEAL


MENSAJE ::ED:: TIPO DE FUNCION OBJETIVO:

               PROBLEMA LINEAL ...... 0: MINIMIZACION DE LA LONGITUD Y ANCHURA
                                      1: MINIMIZACION DE UNA EXPRESION
```

```
PROBLEMA NO LINEAL ... 0: MINIMIZACION DE LA SUPERFICIE
                       1: MINIMIZACION DE UNA EXPRESION


FUNCION OBJETIVO: MINIMIZACION DE LA SUPERFICIE



MENSAJE ::ED:: MATRIZ DE INCIDENCIA VERTICAL

         LA NUMERACION DE LOS ESPACIOS SE HARA A PARTIR DEL 5.
         LA ARISTA QUE REPRESENTA AL CONTORNO SE NUMERARA EN ULTIMO LUGAR.

         SE RECOGEN AQUI LOS REQUISITOS METRICO-GEOMETRICOS.
         LOS DATOS SE DAN EN EL SIGUIENTE ORDEN:
         - ARISTA
         - VERTICE ORIGEN
         - VERTICE EXTREMO
         - COTA INFERIOR DE LA ANCHURA
         - COTA INFERIOR DE LA SUPERFICIE

  A  VO  VE   CI   CI
  -  --  --   --   --
  5   1   3  3.0 10.0
  6   3   5  2.0  6.0
  7   5   7  3.0 12.0
  8   1   2  1.0  0.0
  9   3   4  1.6  3.0
 10   4   7  1.6  6.0
 11   2   3  1.0  0.0
 12   1   2  2.0  6.0
 13   1   2  1.0  0.0
 14   3   6  1.1  1.5
 15   6   7  3.0 10.0
 16   1   3  3.0 10.0
 17   3   6  1.1  1.5
 18   7   1  9.0  0.0

MENSAJE ::ED:: MATRIZ DE INCIDENCIA HORIZONTAL

         IGUAL CRITERIO DE NUMERACION QUE EN LA VERTICAL.

         SE RECOGEN AQUI LOS REQUISITOS METRICO-GEOMETRICOS.
         LOS DATOS SE DAN EN EL SIGUIENTE ORDEN:
         - ARISTA
         - VERTICE ORIGEN
         - VERTICE EXTREMO
         - COTA INFERIOR DE LA LONGITUD

  A  VO  VE   CI
  -  --  --   --
  5   1   2  3.0
  6   1   4  2.0
  7   1   4  3.0
  8   2   3   .6
  9   4   7  1.5
 10   4   7  1.6
 11   2   6  1.0
```





```
12   3   5  2.0
13   5   6   .6
14   7   8  1.1
15   7   9  3.0
16   6   9  3.0
17   8   9  1.1
18   9   1  7.0

-------- LOS RESULTADOS OBTENIDOS PARA CADA ESPACIO Y EL CONTORNO SON: --------

X( 5)=  3.70   Y( 5)=  2.97
X( 6)=  2.00   Y( 6)=  4.10
X( 7)=  3.00   Y( 7)=  4.16
X( 8)=  2.70   Y( 8)=   .85
X( 9)=  1.88   Y( 9)=  1.88
X(10)=  3.12   Y(10)=  1.92
X(11)=  1.90   Y(11)=  3.95
X(12)=  2.70   Y(12)=  2.20
X(13)=  2.70   Y(13)=   .85
X(14)=  1.24   Y(14)=  1.87
X(15)=  3.76   Y(15)=  3.72
X(16)=  3.70   Y(16)=  2.79
X(17)=  1.24   Y(17)=  1.85
X(18)=  8.70   Y(18)=  9.46

------------------------------------------------------------------------------

MENSAJE ::BR:: SE DIBUJA EN EL PLOTTER

MENSAJE ::ED:: SI SE DESEA AJUSTAR LOS RESULTADOS INTRODUCE: 1
               SI QUIERES TERMINAR .............. INTRODUCE: 0
```



```
MENSAJE !!ED!! TITULA EL PROBLEMA

EJEMPLO 3 (DIMENSIONADO)
********************************************************************************

MENSAJE !!ED!! NUMERO DE VERTICES VERTICALES Y HORIZONTALES

NUMERO DE VERTICES VERTICALES ....  8
NUMERO DE VERTICES HORIZONTALES ..  8

MENSAJE !!ED!! COORDENADAS DEL CENTRO GEOMETRICO DE LOS ESPACIOS.

               SE HARA A PARTIR DEL ESQUEMA ADIMENSIONAL,DANDO UNAS DIMENSIONES
               ARBITRARIAS A CADA ESPACIO. ESTOS SE NUMERARANA PARTIR DEL 5.
               AL FINAL DE LA LISTA INTRODUCE! 0

ESP       XG      YG
---       --      --
  5     5.00    1.00
  6     8.00    2.00
  7     7.00    5.00
  8     3.00    4.00
  9     7.00    7.00
 10     9.00    9.00
 11     6.00    9.00
 12     8.00   12.00
 13     2.00   11.00
 14     3.00    1.00
 15     1.00    1.00
 16     5.00   13.00
 17     5.00   11.00

MENSAJE !!ED!! CONDICIONES DE ACCESIBILIDAD ENTRE LOCALES

               SE FIJA UN LOCAL Y SE INDICA CUALES DEBEN SER ACCESIBLES A EL.
               AL FINAL DE LA LISTA INTRODUCE! 0

 L  L1  L2  L3  L4  L5
 -  --  --  --  --  --
 5   6   8  14
 8   7  11
11   9  10  12  13
14  13

MENSAJE !!ED!! TIPO DE PROBLEMA!  1 .. LINEAL / 2 .. NO LINEAL


PROBLEMA NO LINEAL


MENSAJE !!ED!! TIPO DE FUNCION OBJETIVO!

               PROBLEMA LINEAL ...... 0! MINIMIZACION DE LA LONGITUD Y ANCHURA
                                      1! MINIMIZACION DE UNA EXPRESION
```

PROBLEMA NO LINEAL ... 0: MINIMIZACION DE LA SUPERFICIE
1: MINIMIZACION DE UNA EXPRESION

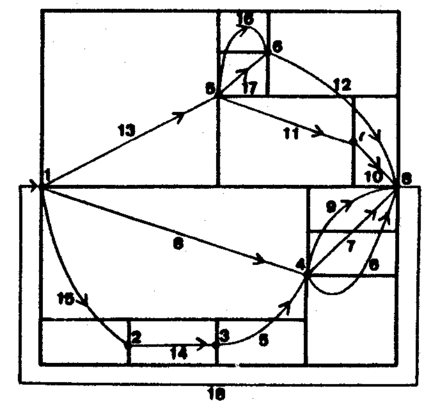
FUNCION OBJETIVO: MINIMIZACION DE LA SUPERFICIE

MENSAJE ::ED:: MATRIZ DE INCIDENCIA VERTICAL

LA NUMERACION DE LOS ESPACIOS SE HARA A PARTIR DEL 5.
LA ARISTA QUE REPRESENTA AL CONTORNO SE NUMERARA EN ULTIMO LUGAR.

SE RECOGEN AQUI LOS REQUISITOS METRICO-GEOMETRICOS.
LOS DATOS SE DAN EN EL SIGUIENTE ORDEN:
- ARISTA
- VERTICE ORIGEN
- VERTICE EXTREMO
- COTA INFERIOR DE LA ANCHURA
- COTA INFERIOR DE LA SUPERFICIE

A	VO	VE	CI	CI
5	3	4	1.1	1.5
6	4	8	2.0	6.0
7	4	8	3.0	10.0
8	1	4	3.0	12.0
9	4	8	2.0	6.0
10	7	8	1.5	0.0
11	5	7	1.1	0.0
12	6	8	3.0	10.0
13	1	5	3.0	10.0
14	2	3	1.5	0.0
15	1	2	1.5	0.0
16	5	6	1.0	0.0
17	5	6	1.0	0.0
18	8	1	0.0	0.0

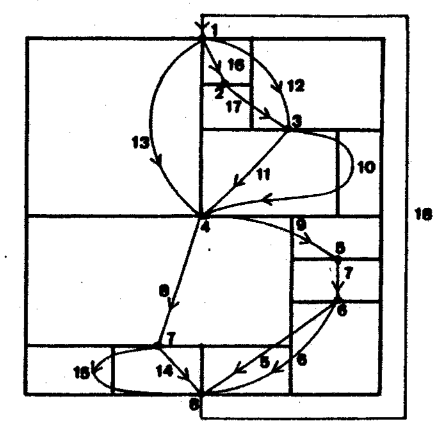


MENSAJE ::ED:: MATRIZ DE INCIDENCIA HORIZONTAL

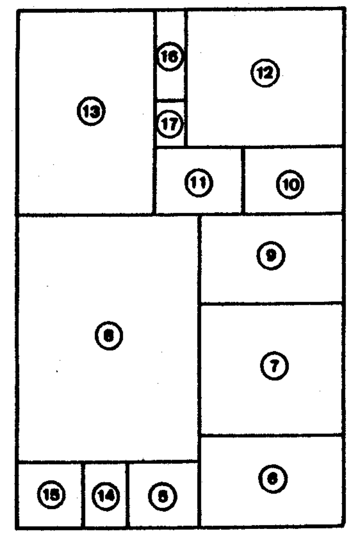
IGUAL CRITERIO DE NUMERACION QUE EN LA VERTICAL.

SE RECOGEN AQUI LOS REQUISITOS METRICO-GEOMETRICOS.
LOS DATOS SE DAN EN EL SIGUIENTE ORDEN:
- ARISTA
- VERTICE ORIGEN
- VERTICE EXTREMO
- COTA INFERIOR DE LA LONGITUD

A	VO	VE	CI
5	7	8	1.1
6	6	8	2.0
7	5	6	3.0
8	4	7	3.0
9	4	5	2.0
10	3	4	1.5
11	3	4	2.0
12	1	3	3.0



```
13   1   4  3.0
14   7   8  1.0
15   7   8  1.5
16   1   2   .6
17   2   3   .6
18   8   1  0.9

-------- LOS RESULTADOS OBTENIDOS PARA CADA ESPACIO Y EL CONTORNO SON: --------

X( 5)=  1.61   Y( 5)=  1.50
X( 6)=  3.24   Y( 6)=  2.06
X( 7)=  3.24   Y( 7)=  3.00
X( 8)=  4.11   Y( 8)=  5.56
X( 9)=  3.24   Y( 9)=  2.00
X(10)=  2.24   Y(10)=  1.50
X(11)=  2.00   Y(11)=  1.50
X(12)=  3.55   Y(12)=  3.10
X(13)=  3.11   Y(13)=  4.60
X(14)=  1.00   Y(14)=  1.50
X(15)=  1.50   Y(15)=  1.50
X(16)=   .69   Y(16)=  2.05
X(17)=   .69   Y(17)=  1.05
X(18)=  7.35   Y(18)= 11.66

------------------------------------------------------------------------------


MENSAJE !!SR!! SE DIBUJA EN EL PLOTTER

MENSAJE !!ED!! SI SE DESEA AJUSTAR LOS RESULTADOS INTRODUCE: 1
               SI QUIERES TERMINAR .............. INTRODUCE: 0
```



1. Orden alfabético de autores

Alexander, C. ENSAYO SOBRE LA SÍNTESIS DE LA FORMA. Ed. Infinito, Buenos Aires, 1976, (4ª ed.). Trad. de NOTE ON THE SYNTHESIS OF FORM. Harvard University Press, Cambridge (Mass.), 1966.
1966

Avondo-Bodino, G. "Graph Theory and operations research". En APPLICATIONS OF GRAPH THEORY, (R.J. Wilson y L.W. Beineke ed.), Academic Press, Londres, 1979, pp. 223-253.
1979

Baglivo, J.A. y Graver J.E. INCIDENCE AND SYMMETRY IN DESIGN AND ARCHITECTURE. Cambridge University Press, col. Cambridge Urban and Architectural Studies, vol. 7, Cambridge, 1983.
1983

Berge, C. GRAPHES, Bordas, 1983 (3ª ed.), 1ª ed. Dunod, 1970.
1970

Bloch, C.J. y Krishamurti. "The counting of rectangular dissections". En ENVIRONMENT AND PLANNING B, Vol. 5, 1978, pp. 207-214.
1978

Broadbent, G. DISEÑO ARQUITECTÓNICO, ARQUITECTURA Y CIENCIAS
1974 HUMANAS. Ed. Gustavo Gili, col. Arquitectura/ Perspectivas, Barcelona, 1976. Trad. de DESIGN IN ARCHITECTURE. ARCHITECTURE AND THE HUMAN SCIENCES. John Whiley & Sons Ltd., Londres 1974.

Brooks, R.L. ed. al. "The dissection of rectangles into squa
1940 res". En DUKE MATHEMATICAL JOURNAL, Vol. 7, 1940, pp. 312-341.

Bruno, J. ed. al. "A new planarity test based on 3-Connecti-
1970 vity". En IEEE. TRANSACTIONS ON CIRCUIT THEORY, Vol. 17, n°2, mayo, 1970, pp. 197-206.

Bryant, P.R. "Graph Theory and electrical networks". En
1979 APPLICATIONS OF GRAPH THEORY, (R.J. Wilson y L.W. Beineke ed.). Academic Press, Londres, 1979, pp. 81-119.

Cattermole, K.W. "Graph Theory and Communications Networks".
1979 En APPLICATIONS OF GRAPH THEORY, (R.J. Wilson y L.W. Beineke ed.). Academic Press, Londres, 1979, pp. 17-57.

Cea, J. OPTIMISATION THEORIE ET ALGORITHMES. Ed. Dunod, col.
1971 Méthodes Mathemâtiques de l'informatique, Vol. 2, París, 1971.

Charalambous, C. "Non linear least pth. optimization and non
1977 linear programming". En MATHEMATICAL PROGRAMMING, Vol. 12, 1977, pp. 195-225.

Cousin, J. "Topological organization of Architectural space".
1970 Architectural Design, Vol. 10, Octubre, 1970, pp. 491-493.

Earl, C.F. "A note on the generation of rectangular dissec-
1977 tions". En ENVIRONMENT AND PLANNING B, Vol. 4, 1977, pp. 241-246.

Earl, C.F. y March, L.J. "Architectural applications of graph
1979 theory". En APPLICATIONS OF GRAPH THEORY, (R.J. - Wilson y L.W. Beineke ed.), Academic Press, Londres, 1979, pp. 327-355.

Eastman, C.M. "Representations for space planning". En COMMU
1970 NICATIONS OF THE ACM, Vol. 13, n°4, Abril, 1970, pp. 242-250.

Eastman, C.M. "Preliminary Report on a system for general -
1972 space planning". En COMMUNICATIONS OF THE ACM, - Vol. 15, n°2, Febrero, 1972, pp. 76-87.

Eastman, C.M. "Automated Space Planning". En ARTIFICIAL INTE
1973 LLIGENCE, Vol. 4, 1973, pp. 41-64.

Fisher, G.J. "Computer recognition and extraction of planar
1966 graphs from the incidence matrix". En IEEE TRANSACTIONS ON CIRCUIT THEORY, Vol. 13, n°2, Junio, 1966, pp. 154-163.

Gero, J.S. "Note on 'Synthesis and optimization of small rec
1977 tangular floor plans' of Mitchell, Steadman and Liggett". En ENVIRONMENT AND PLANNING B, Vol. 4, 1977, pp. 81-88.

Gill, P.E. ed al. PRACTICAL OPTIMIZATION. Academic Press, -
1981 Londres, 1981.

Grooms, H.R. "Algorithm for matrix bandwidth reduction". En
1972 JOURNAL OF THE STRUCTURAL DIVISION, Enero, 1972, pp. 203-214.

Harary, F. ed al. "On enumerating certain design problems in
1978 terms of bicoloured graphs with no isolates". En ENVIRONMENT AND PLANNING B, Vol. 5, 1978, pp. 31-43.

Hock, W. y Schittokowski, K. TEST EXAMPLES FOR NON LINEAR
1981 PROGRAMMING CODES. Ed. Board, col. Lecture Notes in Economics and Mathematical Systems, Vol. 187, Nueva York, 1981.

Hopcroft, J. y Tarjan, R. "Efficient algoritms for graph ma-
1973,a nipulation". En COMMUNICATIONS OF THE ACM, Vol. 16, n°6, Junio 1973, pp. 372-378.

Hopcroft, J. y Tarjan, R. "Dividing a graph into triconnec-
1973,b ted components". En SIAM J. COMPUT., Vol. 2, 1973, pp. 135-158.

Hopcroft, J. y Tarjan, R. "Efficient Planarity Testing". En
1974 JOURNAL OF THE ASSOCIATION FOR COMPUTING MACHINERY, Vol. 21, n°4, Octubre, 1974, pp. 549-568.

Hutton, G. y Roston, M. COMPUTER PROGRAMS FOR THE BUILDING
1979 INDUSTRY. Architectural Press. Londres, 1979, - (2a ed.).

Kaufmann, A. y Cruon, R. LA PROGRAMACIÓN DINÁMICA. Cía. Ed.
1967 Continental, México, 1967, Trad, de LA PROGRAMMATION DYNAMIQUE. Ed. Dunod, París, 1964.

Kaufmann, A. PUNTOS Y FLECHAS. TEORÍA DE LOS GRAFOS. Marcom-
1976 bo S.A., Boixerau Ed.,Barcelona, 1976. Trad. de - DES POINTS ET DES FLECHES... LA THEORIE DES GRAPHES, Ed. Dunod, París.

Korf, R.E. "A shape independent theory of space allocation".
1977 En ENVIRONMENT AND PLANNING B, Vol. 4, 1977, pp. 37-50.

Krejcirik, M. "Computer-aided plant layout". En COMPUTER
1969 AIDED DESIGN, Otoño, 1969, pp. 7-19.

Krishnamurti, R. y O'N Roe, P.H. "Algorithmic aspects of -
1978 plan generation and enumeration". En ENVIRONMENT AND PLANNING B, Vol. 5, 1978, pp. 157-177.

Krishnamurti, R. "3-rectangulations: an algorithm to genera-
1979 te box packings". En ENVIRONMENT AND PLANNING B, Vol. 6, 1979, pp. 331-352.

Kuratowski, K. "Sur le problème des courbes gauches en topo-
1930 logie". En FUNDAMENTA MATH., Vol. 15, 1930, pp. 271-283.

Kuratowski, K. INTRODUCCIÓN A LA TEORÍA DE CONJUNTOS Y A LA
1961 TOPOLOGÍA. Ed. Vicens-Vives, Barcelona, 1966. Trad, de WSTEP DO TEORII MNOGOSCI I TOPOLOGII, Panstowowe Wyda Wnictow Naukowe. Varsovia, 1961.

Larrañeta, J. PROGRAMACIÓN LINEAL Y GRAFOS. Secretariado de
1977 Publicaciones de la Universidad de Sevilla, Sevilla, 1977.

Levin, P.H. "Use of graphs to decide the optimun layout of -
1964 buildings". En THE ARCHITECT'S JOURNAL INFORMATION, Vol. 7, Octubre 1964, pp. 809-815.

March, L. y Steadman, P. THE GEOMETRY OF ENVIRONMENT, RIBA -
1974 Publications Ltd, Londres, 1971. Trad. al italiano como LA GEOMETRIA DELL'AMBIENTE: UNA INTRODU-

ZIONE ALLA ORGANIZZAZIONE SPAZIALE NELLA PROGETTA ZIONE. Gabrielle Mazzota ed. Milán, 1974.

March, L. y Earl, C.F. "On counting architectural plans". En
1977 ENVIRONMENT AND PLANNING B, Vol. 4, 1977, pp. 57-80.

Mitchell, W.J. ed. al. "Synthesis and optimization of small
1976 rectangular floor plans". En ENVIRONMENT AND PLANNING B, Vol. 3, 1976, pp. 37-70.

Mitchell, W.J. COMPUTER-AIDED ARCHITECTURAL DESIGNS, Mason/
1977 Charter Publishers, Nueva York, 1977.

Moucka, J. "The use of propositional calculus in architectu-
1979 ral design". En ENVIRONMENT AND PLANNING B, Vol. 6, 1979, pp. 263-268.

Negroponte, N. LA MACHHINA PER L'ARCHITETTURA. Ed. Il Saggia
1970 tore, col. Struttura e forma urbana, Milán, 1974. Trad. de THE ARCHITECTURE MACHINE, The M.I.T. Press, Cambridge (Mass.), 1970.

Nielsen, F. "Flowgraphs". En APPLICATIONS OF GRAPH THEORY,
1979 (R.J. Wilson y L.W. Beineke ed.), Academic Press, Londres, 1979, pp. 59-80.

Polak, E. COMPUTATIONAL METHODS IN OPTIMIZATION: A UNIFIED
1971 APPROACH, Academic Press, col. Mathematics in science and engineering, Vol. 77, Londres, 1971.

Portlock, P.C. y Whitehead, B. "Three Dimensional Layout
1974 Planning". En BUILD SCIENCE, Vol. 9, 1974, pp. 45-53.

Rao, S.S. OPTIMIZATION THEORY AND APPLICATIONS, Wiley Eas-
1978 tern Limited, Nueva Delhi, 1978.

Read, R. "Algorithms in graph theory". En APPLICATIONS OF
1979 GRAPH THEORY, (R.J. Wilson y L.W. Beineke ed.), Academic Press, Londres, 1979, pp. 381-417.

Ritzman, L.P. "The efficiency of computer algirothms for
1972 plant layout". En MANAGEMENT SCIENCE, Vol. 18, n°5, Enero, Part. I, 1972, pp. 240-247.

Rodrigues, J.S. "Node numbering optimization in structural
1975 analysis". En JOURNAL OF THE STRUCTURAL DIVISION, Febrero, 1975, pp. 361-376.

Scarano, R. PROGETTAZIONE PER OTTIMIZZAZIONE, Liguori, Edito
1979 re, col. La società e la scienza, Vol. 6, Nápoles, 1979.

Seguí, J. "Investigación sobre procesos de diseño. 1- Plan-
1972,a teamiento general". En PUBLICACIÓN DEL SEMINARIO DE ARQUITECTURA Y AUTOMÁTICA EN LA E.T.S.A. DE MA DRID, Madrid, Junio, 1972.

Seguí, J. y Gutiérrez Guitián, M.V. "Ideas para el análisis
1972,b y simulación de procesos artísticos". En PUBLICA CIÓN DEL SEMINARIO DE ARQUITECTURA Y AUTOMÁTICA EN LA E.T.S.A. DE MADRID, Madrid, Junio, 1972.

Seguí, J. y Gutiérrez Guitián, M.V. "Investigación en proce-
1974 sos de diseño. Modelo operativo de formalización". EN BOLETÍN DEL CENTRO DE CÁLCULO DE LA UNIVERSIDAD DE MADRID, n°24, Madrid, Enero, 1974, pp. 1-38.

Sevilla, C. "Investigación sobre procesos de diseño. 2-Modelo de Organismo (I y II partes)". En PUBLICACIÓN DEL SEMINARIO DE ARQUITECTURA Y AUTOMÁTICA EN LA E.T.S.A. DE MADRID, Madrid, Junio, 1972.
1972

Sevilla, C. "Notas sobre la informatización y el diseño de Arquitectura". En INFORMES DE LA CONSTRUCCIÓN, n°337, 1981, pp. 5-20.
1981

Shaviv, E. y Gali, D. "A Model for space allocation in complex buildings: A computer graphic approach". En BUILD. INTERNATIONAL, Vol. 7, 1974, pp. 493-518.
1974

Turner, J.C. MATEMÁTICA MODERNA APLICADA, PROBABILIDADES, ESTADÍSTICA E INVESTIGACIÓN OPERATIVA, Alianza, Edit., Madrid, 1974, Trad, de MODERN APPLIED MATHEMATICS: PROBABILITY-STATISTICS-OPERATIONAL RESEARCH, The English Universities Press Ltd., Londres, 1970.
1970

Tutte, W.T. "A census of planar maps". En CANADIAN JOURNAL OF MATHEMATICS, Vol. 15, 1962, pp.249-271.
1962

Tutte, W.T. "How to draw a graph". En PROC. LONDON MATH. SOC., Vol. 3, n°13, 1963, pp. 743-768.
1963

Varios autores. ACTAS DEL CONGRESO DE LA U.I.A., Madrid, 1975.
1975

Wilson, R.J. y Beineke L.W. APPLICATIONS OF GRAPH THEORY, Academic, Press, Londres, 1979.
1979

Wright, R. ed. al. "Constraint processing in design". En JOURNAL OF THE STRUCTURAL DIVISION, Enero, 1971, pp. 481-494.
1971