

19



OFICINA ESPAÑOLA DE  
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 987 437**

51 Int. Cl.:

**G06F 9/44** (2008.01)

**G06F 9/50** (2006.01)

**G06Q 10/047** (2013.01)

**G06Q 10/0631** (2013.01)

**G06Q 10/0633** (2013.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **22.09.2014 PCT/FI2014/050723**

87 Fecha y número de publicación internacional: **26.03.2015 WO15040282**

96 Fecha de presentación y número de la solicitud europea: **22.09.2014 E 14802698 (2)**

97 Fecha y número de publicación de la concesión europea: **24.07.2024 EP 3049926**

54 Título: **Generación de soluciones paralelas**

30 Prioridad:

**23.09.2013 FI 20135946**

45 Fecha de publicación y mención en BOPI de la traducción de la patente:

**14.11.2024**

73 Titular/es:

**AI DIMENSION OY (100.0%)**

**Tietotie 2**

**90460 Oulu, FI**

72 Inventor/es:

**BRÄYSY, OLLI**

74 Agente/Representante:

**LINAGE GONZÁLEZ, Rafael**

ES 2 987 437 T3

Aviso: En el plazo de nueve meses a contar desde la fecha de publicación en el Boletín Europeo de Patentes, de la mención de concesión de la patente europea, cualquier persona podrá oponerse ante la Oficina Europea de Patentes a la patente concedida. La oposición deberá formularse por escrito y estar motivada; sólo se considerará como formulada una vez que se haya realizado el pago de la tasa de oposición (art. 99.1 del Convenio sobre Concesión de Patentes Europeas).

**DESCRIPCIÓN**

Generación de soluciones paralelas

5 **Campo**

La invención se refiere a un aparato de procesamiento de datos y a un código de programa informático, que se utilizan para generar una solución paralela.

10 **Antecedentes**

La computación paralela masiva se ha convertido en el factor central para aumentar la potencia y la velocidad de optimización. En este sentido, la utilización de tecnologías tanto de CPU (Unidad Central de Procesamiento) como de GPU (Unidad de Procesamiento de Gráficos) desempeña un papel importante. El desafío es que los procedimientos de optimización actuales, en particular en el caso de problemas de optimización discretos NP-hard (*Nondeterministic Polynomial-time hard [Tiempo polinómico no determinista difícil]*), son poco adecuados para la computación paralela masiva. Además, la optimización de problemas prácticos realistas es desafiante debido a heurísticas de mejora complejas y el mantenimiento de estos procedimientos es laborioso.

20 En base a lo indicado anteriormente, varios tipos de procedimientos que se basan en particiones, como la cobertura de conjuntos y la partición de conjuntos, son una opción lógica. En ellos, al optimizar las rutas, por ejemplo, primero se crea un conjunto de rutas o partes del mismo y luego las rutas creadas se combinan en una solución, ya sea de manera exacta o heurística.

25 Existen algunas investigaciones sobre el uso de GPU para la optimización. La GPU ha sido utilizada para distribuir partes del trabajo a realizar entre diferentes procesadores. En la práctica, la evaluación de movimientos se ha realizado con la ayuda de la GPU. Siempre es necesario realizar una evaluación con respecto a un punto de referencia, es decir, siempre que se encuentra un movimiento de mejora, se deben interrumpir las evaluaciones, actualizar el punto de referencia y comenzar desde el principio. Debido a que la actualización es una operación ardua, las GPU están en un estado de espera la mayor parte del tiempo. Por lo tanto, las ventajas de velocidad obtenidas son marginales. Se han realizado algunos intentos para hacer que las evaluaciones sean proporcionalmente más complejas y/o las operaciones de actualización proporcionalmente más ligeras, con lo que habría menos tiempo de espera para los procesadores, pero, por consiguiente, se pierde tiempo en computación ineficiente.

35 El problema fundamental de los procedimientos de cobertura de conjuntos es que generar rutas compatibles y al mismo tiempo eficientes y suficientemente diferentes es un desafío inconmensurablemente grande si las rutas se construyen de forma independiente o en paralelo. Esto hace que, para compilar una buena solución, la cantidad de rutas individuales construidas debe ser realmente alta, lo que se convierte en un desafío a pesar del aumento en la potencia de computación.

Los siguientes documentos abordan el problema del enrutamiento de vehículos:

- 45 - Chris Groer et al.: "A Parallel Algorithm for the Vehicle Routing Problem (Un algoritmo paralelo para el problema de enrutamiento de vehículos)", *INFORMS Journal on Computing*, vol. 23, núm. 2, 1 de mayo de 2011 (2011-05-01), páginas 315-330;
- Teodor Gabriel Crainic et al.: "Parallel meta-heuristics (Metaheurísticas paralelas)", en *Informes de CIRRELT*, Universidad de Montreal, Canadá. CIRRELT-2009-22, 1 de mayo de 2009 (2009-05-01), páginas 1-53, Universidad de Montreal;
- 50 - Nagata Y et al.: "A powerful route minimization heuristic for the vehicle routing problem with time windows (Una poderosa heurística de minimización de rutas para el problema de enrutamiento de vehículos con ventanas de tiempo)", *OPERATIONS RESEARCH LETTERS*, NORTH-HOLLAND, AMSTERDAM, NL, vol. 37, no. 5, 1 de septiembre de 2009 (2009-09-01), páginas 333-338;
- 55 - Olli Bräysy et al.: "A multi-start local search algorithm for the vehicle routing problem with time windows (Un algoritmo de búsqueda local de inicio múltiple para el problema de enrutamiento de vehículos con ventanas de tiempo)", *European Journal of Operational Research*, vol. 159, núm. 3, 8 de octubre de 2003, páginas 586-605; y
- 60 - Christian Schulz et al.: "GPU computing in discrete optimization. Part II: Survey focused on routing problems (Computación con GPU en optimización discreta. Parte II: Encuesta centrada en problemas de enrutamiento)", *EURO Journal on Transportation and Logistics*, vol. 2, no. 1-2, 24 de abril de 2013 (2013-04-24), páginas 159-186.

**Breve descripción**

65 La presente invención busca proporcionar un aparato de procesamiento de datos mejorado y un código de

programa informático mejorado.

De acuerdo con un aspecto de la presente invención, se proporciona un aparato de procesamiento de datos como se especifica en la reivindicación 1.

5

De acuerdo con otro aspecto de la presente invención, se proporciona un medio de distribución legible por ordenador como se especifica en la reivindicación 7.

10

Las ventajas de la invención incluyen, al menos, un excelente soporte para computación paralela masiva y diversos modelos, así como su universalidad y viabilidad. Debido a que la combinación se realiza en orden, un componente a la vez resulta inmediatamente evidente cuál es la mejor selección o si vale la pena continuar con la ruta de combinación seleccionada.

**Lista de figuras**

15

A continuación, se describirán a modo de ejemplo realizaciones de la invención, haciendo referencia a los dibujos adjuntos, en los que:

20

Las Figuras 1, 2, 3, 4 y 5 muestran realizaciones ejemplares de un aparato de procesamiento de datos;  
 Las Figuras 6, 7 y 8 muestran realizaciones ejemplares de componentes de solución y una solución;  
 Las Figuras 9A, 9B, 9C, 9D, 9E, 9F, 9G, 9H, 9I, 9J y 9K muestran una realización ejemplar de generación de una solución, y;  
 La Figura 10 muestra una realización ejemplar de un aparato de procesamiento de datos.

25

**Descripción de las realizaciones**

30

Las realizaciones que se divulgan a continuación son meros ejemplos. La invención no se limita a las realizaciones descritas, sino que son ejemplos de formas de implementación factibles. También se pueden combinar características de varias realizaciones a menos que sean específicamente contradictorias o alternativas con respecto a su implementación técnica. La palabra "que comprende" significa que la realización descrita incluye las características a las que se hace referencia, pero también puede haber características adicionales que no se mencionan. Debe observarse que las figuras presentan diagramas de bloques simplificados en los que solo se describen las características esenciales desde el punto de vista de las realizaciones; un experto en la técnica encontrará obvio que el aparato de procesamiento de datos también puede incluir muchas otras características y estructuras. Sin embargo, dichas otras partes pueden ser irrelevantes con respecto a la invención real y, por lo tanto, no es necesario analizarlas con mayor detalle en la presente. Además, debe notarse que, aunque los elementos se describen como separados, algunos de ellos pueden estar integrados en un elemento físico, es decir, los elementos descritos pueden ser elementos lógicos.

40

La Figura 1 describe un aparato de procesamiento de datos 100, que comprende una memoria 102 que incluye un código de programa informático 104, y al menos dos procesadores 110, 116, que están configurados para ejecutar el código de programa informático 104.

45

El código de programa informático 104 comprende un programa de componentes 112, 118 que se ejecuta en al menos dos procesadores 110, 116 para formar componentes de solución 106 compilados de puntos en paralelo, y para almacenar los componentes de solución 106 formados en la memoria 102. En una realización ejemplar, los componentes de solución 106 no están distribuidos como tales entre los procesadores 110, 116, sino que cada procesador 110, 116 tiene acceso a todos los componentes almacenados 106. Si los tiempos de búsqueda desde la memoria 102 se vuelven demasiado altos como resultado de superposiciones, se puede realizar el número requerido de copias de los componentes de solución 106. Debido a que los requisitos de memoria son pequeños, esto se puede llevar a cabo. El identificador de la biblioteca de componentes 106 puede distribuirse entre los procesadores 110, 116. Los componentes de solución pueden formarse 106 de conformidad con reglas predeterminadas: si el componente de solución es, por ejemplo, una ruta de distribución es posible determinar, por ejemplo, la longitud y el tiempo máximos para la ruta, o el número máximo de puntos, para no superar el tiempo de conducción permitido por el convenio colectivo de trabajo.

50

60

Además, el código de programa informático 104 comprende un programa de solución 114, 120 que se ejecuta en paralelo en al menos dos procesadores 110, 116 para generar una solución 108 agregando un componente de solución 106 a la vez, leído desde la memoria 102, a la solución 108 basándose en un punto clave, y para almacenar el componente de solución 106 agregado a la solución 108 en la memoria 102. Con la ayuda de un punto clave, se pueden definir componentes de solución 106 próximos entre sí. En una realización ejemplar, se selecciona un punto clave que ya tiene el coste de colocación más bajo en relación con el componente de solución 106 ya seleccionado para la solución 108. En una realización ejemplar, se selecciona un punto clave que pertenece al primer componente de solución 106 adecuado. En una realización ejemplar, se selecciona un punto clave que pertenece al mejor componente de solución 106 que se va a combinar. En una realización

65

ejemplar, se puede utilizar potencialmente una estrategia del tipo *look ahead* para anticipar el efecto de la selección realizada y concluir en base a ello cuál es la mejor solución: de esta forma, se pueden revisar varios tipos de cadenas potenciales de selecciones de la misma manera que, por ejemplo, se ponderan los efectos de una jugada sobre jugadas futuras en ajedrez. En una realización ejemplar, se selecciona el componente de solución 106 que contiene el mejor punto clave determinado de acuerdo con algún criterio de selección.

En una realización ejemplar, cada procesador 110, 116 genera una solución de forma totalmente independiente y, por lo tanto, pueden tener una gran superposición. Cada procesador 110, 116 puede construir de forma independiente una solución completa. En una realización ejemplar, a cada procesador 110, 116 se le puede asignar un punto de inicio diferente, porque el componente de solución 106 seleccionado en primer lugar tiene un efecto importante sobre los seleccionados posteriormente. El punto de inicio diferente, es decir, el primer componente de solución seleccionado se puede determinar para cada procesador 110, 116 de acuerdo con reglas predeterminadas de tal manera que, por ejemplo, se seleccionen los componentes de solución 106 que estén lo más alejados posible entre sí. Alternativamente, la selección del punto de inicio para cada procesador 110, 116 se puede llevar a cabo, por ejemplo, con un número aleatorio de modo que el algoritmo de selección se inicialice para cada procesador 110, 116 con su propio número aleatorio. La influencia de la aleatoriedad también puede ampliarse: una de las mejores opciones puede seleccionarse aleatoriamente en cada etapa como el componente de solución 106 posterior. También pueden realizarse aleatoriamente varios tipos de movimientos de mejora en diferentes etapas. No es necesario combinar las soluciones 108 producidas por diferentes procesadores 110, 116, solo los componentes de solución 106 a medida que se construyen las soluciones 108. A medida que se forma la solución 108, la combinación se lleva a cabo simplemente actualizando la estructura de información incluida en la solución 108. Por lo tanto, en todo momento puede haber una pluralidad de soluciones 108 en proceso; en principio, cada procesador 110, 116 construye su propia solución 108 o al menos una parte independiente de la solución 108 utilizando los mismos componentes de solución 106, por lo que las actualizaciones paralelas no presentan ningún problema.

Si los procesadores 110, 116 forman partes independientes de la solución 108, se puede facilitar la combinación de la solución 108 de tal manera que los puntos se dividan en subconjuntos separados de antemano, es decir, que no se superpongan. En este caso, la combinación es principalmente una cuestión tecnológica de informes. De este modo, por ejemplo, si tomamos como ejemplo la limpieza de nieve, donde una ciudad se ha dividido en tres áreas de contrato, cada área de contrato es una parte independiente que no tiene nada en común con las otras áreas. La limpieza de nieve de la ciudad consta entonces únicamente de estas áreas de contrato que se han optimizado por separado.

Si cada uno de los procesadores 110, 116 crea su propia solución perfecta 108, obviamente se consumen más recursos que si se generara una sola solución 108 pero, por otro lado, resolver problemas de optimización tan difíciles es, de todos modos, andar a tientas, en lo que una cantidad sustancial de trabajo "se va por el desagüe". La mejor manera de combinar diferentes tipos de buenos componentes de solución 106 solo se puede encontrar probando tantas combinaciones como sea posible, y las realizaciones ejemplares descritas lo permiten y utilizan la potencia del ordenador tanto como sea posible. Como resultado final, se obtiene una solución 108, con el mejor valor para su función objetivo, o bien un conjunto de mejores soluciones 108 que son las mejores según se mide con al menos un indicador, y el usuario selecciona entonces la más adecuada para el propósito.

Como se aprecia en la Figura 1, hay al menos dos procesadores 110, 116, es decir, puede haber más de ellos: como se describe en la Figura 1, puede haber N piezas de procesadores 110, 116, 122, donde N es un número entero mayor que dos. Por lo tanto, el programa de solución 114, 120, 126 puede ejecutarse en paralelo en más de dos procesadores 110, 116, 122, hasta un máximo de N procesadores 110, 116, 122. El programa de componentes 112, 118, 124 también se ejecuta en al menos dos procesadores 110, 116, 122, es decir, puede ejecutarse en más, hasta un máximo de N procesadores 110, 116, 122.

En una realización ejemplar, el primer procesador 110 comprende un microprocesador, un núcleo de procesador o un procesador de gráficos, y el segundo procesador 116 comprende un microprocesador, un núcleo de procesador o un procesador de gráficos.

En la realización ejemplar de la Figura 2, los procesadores 110, 116, 122 se implementan como microprocesadores 202, 204, 206 con lo que el programa de componentes 112, 118, 124 y el programa de solución 114, 120, 126 pueden ejecutarse en paralelo en hasta N microprocesadores 202, 204, 206.

En la realización ejemplar de la Figura 3, los procesadores 110, 116, 122 se implementan como núcleos de procesador 302, 304, 306 de un procesador multinúcleo 300 con lo que el programa de componentes 112, 118, 124 y el programa de solución 114, 120, 126 pueden ejecutarse en paralelo en hasta N núcleos de procesador 302, 304, 306.

En la realización ejemplar de acuerdo con la Figura 4, los procesadores 110, 116 se implementan como una unidad central de procesamiento (CPU) 400 y una unidad de procesamiento de gráficos (GPU) 402, por lo que el

programa de componentes 112, 118 y el programa de solución 114, 120 pueden ejecutarse en paralelo en la CPU 400 y la GPU 402. En una realización ejemplar, la unidad de procesamiento principal 400 se implementa como un microprocesador. Las unidades de procesamiento de gráficos 402 pueden implementarse utilizando una tarjeta gráfica independiente. La tarjeta gráfica puede incluir una gran cantidad de unidades de procesamiento de gráficos 402, hasta tres mil, por ejemplo. Además, una tarjeta gráfica puede tener su propia memoria. En la memoria de la tarjeta gráfica, se puede almacenar una copia de los componentes de solución 106, porque la unidad de procesamiento de gráficos 420 es muy potente y las operaciones de memoria dentro de la tarjeta gráfica son rápidas, pero la transferencia de datos en la dirección de la unidad de procesamiento principal 400 es relativamente lenta.

Sobre la base de estas realizaciones ejemplares descritas en las Figuras 2, 3 y 4, se pueden formar varios tipos de combinaciones híbridas.

En la realización ejemplar de la Figura 5, los procesadores 110, 116, 122, 500 se implementan como núcleos de procesador 302, 304, 306 de un procesador multinúcleo 300 y como una unidad de procesamiento de gráficos 302, con lo que el programa de componentes 112, 118, 124, 502 y el programa de solución 114, 120, 126, 504 pueden ejecutarse en paralelo en hasta N+1 procesadores 110, 116, 122, 500.

Generalmente, el procesador 110, 116, 122 puede implementarse como un microprocesador, una unidad de procesamiento de gráficos, un núcleo de procesador de un procesador multinúcleo, un procesador de un procesador paralelo, pero también utilizando, además de o en lugar de, al menos uno de los siguientes: un circuito integrado estándar, un circuito integrado de aplicación específica (ASIC), un sistema en un chip (SoC), un producto estándar de aplicación específica (ASSP), un procesador digital de señales o un chip de ordenador de propósito especial.

En una realización ejemplar, el procesador 110, 116, 122 es parte de un ordenador digital electrónico 100. El ordenador 100 puede comprender una memoria de acceso aleatorio (RAM), una unidad central de procesamiento (CPU) y un reloj de sistema. La unidad central de procesamiento puede comprender una serie de registros, una unidad lógica aritmética (ALU) y una unidad de control. La unidad de control está controlada por una secuencia de comandos de programa 104, que se ha transferido a la unidad central de procesamiento desde la memoria de acceso aleatorio. La unidad central de procesamiento puede incluir una serie de microcomandos para operaciones básicas. La implementación de los microcomandos puede variar dependiendo del diseño de la unidad central de procesamiento. Los comandos de programa 104 pueden estar codificados en un lenguaje de programación, que puede ser un lenguaje de programación de alto nivel, como C o Java, o un lenguaje de programación de nivel inferior, como un lenguaje de máquina o un ensamblador. El ordenador 100 también puede tener un sistema operativo que puede proporcionar servicios de sistema para un programa informático 104 escrito por medio de comandos de programa. Con referencia a la descripción anterior, la CPU puede ser un microprocesador o un procesador multinúcleo, y el ordenador 100 puede comprender además una unidad de procesamiento de gráficos. La memoria 102 puede ser una memoria de acceso aleatorio, pero el ordenador 100 puede tener una memoria no volátil en la que se almacena el código del programa informático 104. En el nivel más bajo, la información puede procesarse como bits (o qubits, si el ordenador es un ordenador cuántico).

Como se describe en la Figura 1, el código de programa informático 132 puede almacenarse en un medio de distribución 134 desde el cual es transferible al ordenador 100, por lo que el código de programa informático 104, cuando se ejecuta en al menos dos procesadores 110, 116, 122, forma los componentes de solución 106, y cuando se ejecuta en al menos dos procesadores 110, 116, 122, forma la solución 108. En una realización ejemplar, el código de programa informático 132, 104 puede estar en formato de código fuente, formato de código objeto, formato ejecutable o en un formato de transición. El medio de distribución 134 puede comprender al menos lo siguiente: cualquier entidad que permita la distribución del programa, medios de memoria, memoria de solo lectura, señal de comunicaciones, medios de almacenamiento legibles por ordenador no transitorios.

Las Figuras 6, 7 y 8 muestran realizaciones ejemplares de los componentes de solución 106 y una solución 108.

La Figura 6 tiene los puntos 600, 602, 604, 606, 608, 610, 612, 614, 616, de los cuales se generan los componentes de solución 106. El componente de solución K1 620 contiene los puntos 600, 602 y 604. El componente de solución K2 622 contiene los puntos 606, 608 y 610. El componente de solución K3 624 contiene los puntos 614 y 616. El componente de solución K4 626 contiene los puntos 606 y 612. Como se muestra en la Figura 6, un punto puede pertenecer a varios componentes de solución: en nuestro ejemplo, el punto 606 pertenece tanto al componente de solución K2 622 como al componente de solución K4 626. La Figura 6 también muestra que no todos los puntos necesitan pertenecer a cualquier componente de solución: en nuestro ejemplo, el punto 618 no fue colocado en ningún componente de solución (en esta etapa todavía).

En una realización ejemplar, el programa de solución 114, 120, 126 genera adicionalmente una solución 108 de modo que en cada procesador 110, 116, 122, la solución 108 se genera independientemente. De acuerdo con la Figura 7, puede haber una pluralidad de soluciones 108: en nuestro ejemplo, la solución 700 generada en el

procesador 1 110, y la solución 702 generada en el procesador 2 116. De estas, el aparato de procesamiento de datos 100 puede seleccionar automáticamente una solución 700/702 utilizando un criterio de calidad. Alternativamente, el usuario selecciona una de las soluciones 700/702, posiblemente soportada por el aparato de procesamiento de datos 100 de modo que el aparato de procesamiento de datos 100 presente información sobre las soluciones 700, 702 al usuario de acuerdo con al menos un indicador.

En la realización ejemplar de la Figura 8, el programa de componentes 112, 118, 124 divide adicionalmente los puntos 600, 602, 604, 606, 608, 610, 612, 614, 616, 618 en al menos dos subconjuntos específicos del procesador 800, 810 y genera los componentes de solución 802, 804, 812, 814 por separado para cada procesador 110, 116, 122, y el programa de solución 114, 120, 126 genera adicionalmente la solución 108 de tal manera que en cada uno de los al menos dos procesadores 110, 116, 122 la solución 820, 822 se genera específicamente para el procesador mediante el uso de los componentes de solución específicos del procesador 802, 804, 812, 814. El aparato de procesamiento de datos 100 luego combina las soluciones generadas 820, 822 en la solución final 108.

El procedimiento descrito en el que se utilizan los puntos 600-618 para generar los componentes de solución 620, 622, 624, 802, 804, 812, 814, de los que se genera además la solución 108, 700, 702, 820, 822, se puede aplicar a la optimización de rutas en logística. Los componentes de solución que se construyen en la optimización de rutas pueden ser rutas individuales y/o partes de las mismas.

Un segundo ejemplo de realización se refiere a la planificación de turnos de trabajo. Un tercer ejemplo de realización, de entre una pluralidad de posibles aplicaciones, se refiere a la planificación de la producción, es decir, a la distribución y asignación de tareas entre diferentes recursos. Esto se puede realizar aplicando básicamente los mismos principios que se utilizan para la optimización de rutas.

A continuación, se examinan las Figuras 9A, 9B, 9C, 9D, 9E, 9F, 9G, 9H, 9I, 9J y 9K, que muestran realizaciones ejemplares de la generación de una solución 108 con respecto a la optimización de rutas en logística.

En la Figura 9A, el elemento rectangular 900 es un nodo con el que se deben unir los componentes de solución. En relación con la optimización de rutas, el nodo puede ser, por ejemplo, una terminal, un depósito, un almacén o similar. En nuestro ejemplo, solo hay un nodo 900, pero también podría haber más de ellos. Los componentes de solución se generan a partir de los puntos 902-940 con los programas de componentes 112, 118, 124 que se ejecutan en paralelo.

En la Figura 9B, se ha generado un primer componente de solución 950, que contiene los puntos 902, 904, 906, 908, 910, 912, 914, 916, 918 y 920.

En la Figura 9C, se ha generado un segundo componente de solución 952, que contiene los puntos 922, 924, 926 y 928.

En la Figura 9D, se ha generado un tercer componente de solución 954, que contiene los puntos 930, 932, 910 y 916.

En la Figura 9E, se ha generado un cuarto componente de solución 956, que contiene los puntos 930, 932, 916, 934 y 936.

La Figura 9F describe los cuatro componentes de solución 950, 952, 954 y 956 generados con el programa de componentes 112.

Luego, se inicia la generación de la solución 108 con los programas de solución 114, 120, 126 ejecutándose en paralelo.

En la Figura 9G, se ha seleccionado un primer componente de solución 950. Una vez seleccionado el primer componente de solución 950, se busca un punto clave con respecto al mismo. En nuestro ejemplo, el punto clave se define como el punto más cercano, es decir, el punto cuyo coste de colocación en la ruta en cuestión sería el más bajo. En esta realización ejemplar, el punto 928 es el primer punto clave. Después de esto, se busca si existe una ruta entre los componentes de solución almacenados 952, 954, 956, que incluya el punto clave 928 pero no un único punto superpuesto con respecto a los componentes de solución seleccionados previamente, es decir, en este ejemplo, el primer componente de solución 950. Se encontró dicho componente de solución: este es el segundo componente de solución 952. En una realización ejemplar, el programa de solución 114, 120, 126 selecciona adicionalmente para la solución 108 el componente de solución 952 que tiene el mejor punto clave 928 y que no presenta superposiciones con los componentes de solución 950 ya seleccionados para la solución 108.

En la Figura 9H después de esto, se busca nuevamente el punto clave más próximo al primer componente de

solución 950, no seleccionado hasta ahora. En esta realización ejemplar, el punto 930 es el siguiente punto clave. Después de la identificación del punto clave 930, se buscan nuevamente rutas a partir de los componentes de solución 954, 956 almacenados, aún no utilizados, que incluyen el punto clave 930 y la menor cantidad posible de puntos de superposición con los componentes de solución 950, 952 ya seleccionados. Ahora se encuentran dos componentes de solución 954, 956 que incluyen el punto clave, pero ambos tienen puntos de superposición con los componentes de solución ya seleccionados, en este caso, el componente de solución 950.

De acuerdo con la Figura 9I, el cuarto componente de solución 956 se selecciona de entre los componentes de solución 954, 956, porque tiene menos puntos superpuestos y su longitud proporcional, teniendo en cuenta la cantidad de puntos y la longitud del recorrido, es mejor. Naturalmente, el criterio de selección también puede ser diferente.

Luego, el punto de superposición 916 del primer componente de solución 950 y el cuarto componente de solución 956 se elimina de la ruta del cuarto componente de solución 956, después de lo cual el cuarto componente de solución 960 modificado es como se muestra en la Figura 9J .

En una realización ejemplar, el programa de solución 114, 120, 126 selecciona adicionalmente para la solución 108 el componente de solución 956 que incluye el punto clave 930 pero presenta superposiciones con los componentes de solución 950, 952 ya seleccionados para la solución 108, y que presenta el mejor valor de eficiencia después de haber eliminado las superposiciones.

Finalmente, se intenta situar los puntos 938 y 940 en la última ruta generada 960, lo que se consigue, con lo que se obtiene una situación de acuerdo con la Figura 9K, en la que el cuarto componente de solución final 962 consiste en los puntos 930, 932, 938, 934, 940 y 936. En principio, la búsqueda continúa de esta manera hasta que se hayan seleccionado todos los puntos o hasta que no se encuentren más componentes que incluyan el punto clave en la biblioteca de componentes de solución.

En una realización ejemplar, el programa de solución 114, 120, 126 coloca adicionalmente, después de la eliminación de superposiciones, puntos de solución individuales todavía no seleccionados en ese momento en los componentes de solución de la solución 108, que ya han sido seleccionados, en nuestro ejemplo los puntos 938, 940 en el componente de solución 956 del que finalmente se genera el componente de solución 962.

Además, es posible (no se muestra en las Figuras 9A-9K) que, en una realización, el programa de solución 114, 120, 126 genere adicionalmente un componente de solución dedicado para dichos puntos faltantes de la solución 108, para los cuales no se puede encontrar ningún componente de solución y que no se pueden colocar en los componentes de solución de la solución 108, que ya han sido seleccionados.

Como se muestra en la Figura 9K, la solución final 108 se genera a partir de los componentes de solución 950, 952 y 962.

En las realizaciones ejemplares de las Figuras 9A-9K, el área de aplicación del código de programa 104 comprende la optimización de rutas en logística, mediante lo cual:

- los componentes de solución 950, 952, 954, 956 comprenden rutas individuales y/o partes de rutas individuales;
- la evaluación de las soluciones 108 se basa en al menos uno de los siguientes criterios: longitud de las rutas, capacidad de utilización de las rutas, tiempo total de las rutas, asignación justa de las rutas, adecuación de las rutas a los límites de tiempo, compatibilidad de las rutas, prioridad de las rutas, pero naturalmente se pueden utilizar muchos otros criterios;
- los puntos clave 928, 930 se obtienen definiendo puntos no enrutados, en nuestro ejemplo estos fueron los puntos no enrutados más cercanos en relación con el primer componente de solución seleccionado;
- las correcciones están basadas en la colocación y/o cambios de puntos individuales; y
- la solución combinada 108 comprende un conjunto compatible 950, 952, 962 de rutas para un área y/o período de tiempo predeterminado .

En otra realización ejemplar, el área de aplicación del código de programa informático 104 comprende la planificación de turnos de trabajo, mediante lo cual:

- los componentes de solución comprenden turnos de trabajo individuales y/o partes de turnos de trabajo individuales;
- la evaluación de las soluciones está basada, por ejemplo, en la cantidad de tareas no asignadas, la cantidad de horas extras, las necesidades de los empleados y/o el cumplimiento de los deseos de los empleados;
- los puntos clave se obtienen definiendo tanto temporal como regionalmente el punto más cercano en relación con los puntos ya seleccionados;

- las correcciones están basadas en la colocación y/o cambios de tareas individuales; y
- la solución combinada comprende un conjunto compatible de turnos de trabajo asignados a los empleados para ubicaciones físicas y/o períodos de tiempo predefinidos.

5 Un problema en la planificación de turnos de trabajo es agrupar los puntos en tales componentes de solución 106 de modo que la solución 108 pueda ser compilada a partir de ellos sin tener que volver a calcular los costos en la etapa de combinación. Si diferentes componentes de solución 106 tienen turnos de trabajo del mismo empleado, se verifica en la etapa de combinación que haya tiempo adecuado para descansar entre ellos y que el empleado tenga suficientes días de vacaciones y días libres. Esto tiene tres rutas diferentes: 1) Si los componentes de solución 106 son partes de turnos de trabajo, se verifica la compatibilidad en la etapa de combinación. 2) De la misma manera, si los turnos de trabajo están colocados uno después del otro, se verifica su compatibilidad en la etapa de combinación. 3) El componente de solución 106 corresponde a las acciones de un empleado durante el período de tiempo seleccionado, lo que en principio disminuye el número de componentes de solución, pero por otro lado permite la distribución de las mismas partes de un todo entre diferentes procesadores 110, 116, por lo que habrá un uso práctico adecuado para ellos.

En una tercera realización ejemplar, el área de aplicación del código de programa informático 104 comprende la planificación de la producción, mediante lo cual:

- los componentes de solución comprenden planes de uso para máquinas individuales;
- la evaluación de las soluciones está basada, por ejemplo, en los costes, la fiabilidad o el grado de utilización de las máquinas;
- los puntos clave se obtienen en función de la similitud entre los requisitos de recursos o tiempo de las tareas laborales;
- las correcciones están basadas en la variación de los tiempos y la asignación de tareas laborales entre diferentes recursos; y
- la solución combinada comprende un plan de producción integral para un período de tiempo predeterminado.

30 Lo novedoso en las realizaciones ejemplares descritas es la combinación de los componentes de solución 106 de tal manera que durante la etapa de combinación se realizan operaciones correctivas, incluyendo eliminaciones, adiciones y reemplazos. Lo novedoso además de lo anterior es la realización de las combinaciones de forma heurística, apoyándose en los puntos clave 928, 930 con lo que no es necesario examinar todas las combinaciones posibles. Debido al uso de correcciones multiformes y puntos clave 928, 930 la solución no es obvia. Otra novedad es la realización de las operaciones de combinación en paralelo. Las realizaciones ejemplares resuelven los mayores problemas, optimizan más rápidamente los problemas desafiantes del mundo real y mejoran la calidad de los resultados.

40 Aunque el enfoque de cobertura de conjuntos permite superposiciones y se han publicado varios tipos de operadores de corrección en relación con el problema en cuestión, no se conoce un procedimiento integral en el que se permitan tanto las superposiciones como los puntos faltantes, así como sus correcciones, en la combinación de los componentes de solución 106. Sin embargo, un procedimiento de este tipo permite compilar buenas soluciones 108 de manera eficaz y en paralelo, sin necesidad de configurar y mantener un vasto registro base en la memoria 102.

45 Por último, se examina la Figura 10, que muestra una realización ejemplar del aparato de procesamiento de datos 100.

50 En primer lugar, se lleva a cabo una construcción paralela 1000 de los componentes de solución. Los componentes de solución 106 pueden almacenarse en una estructura de datos compacta 1004, por ejemplo, en una tabla, que sólo contiene la información necesaria desde el punto de vista de la descripción del componente de solución. En la memoria 102, sólo puede almacenarse un conjunto limitado de componentes de solución 106 compilados, para los que se ha evaluado el mejor valor de eficiencia 1002. La definición del valor de eficiencia depende de la aplicación, pero por regla general depende de los costes y del grado de utilización de los recursos.

55 Después de esto, cada procesador que está en uso toma decisiones 1008 de forma independiente desde diferentes puntos de partida agregando componentes a la solución uno a uno en paralelo en la construcción 1006. En cada etapa de adición, se selecciona como base de los exámenes el componente de solución 106 que tiene el mejor punto clave con respecto a los componentes ya seleccionados en la evaluación 1010. Esto evita tener que recorrer todas las combinaciones y la búsqueda puede interrumpirse en etapas tempranas, si no se pueden encontrar componentes de solución 106 adecuados para agregar. El procedimiento agrega al conjunto de los seleccionados componentes de solución 106 que contienen un punto clave, que no tienen superposiciones directas. Esto se puede hacer, porque el examen solo contiene buenos componentes de solución 106, en lo que respecta a sus valores de eficiencia. Si no hay ningún componente de solución 106 adecuado en el conjunto almacenado, primero se examinarán los componentes de solución 106, que incluyen un punto clave, pero al

mismo tiempo también partes de solución superpuestas con los componentes de solución 106 ya seleccionados. De estos, se selecciona el componente de solución 106 que tiene el mejor valor de eficiencia después de que se hayan eliminado las superposiciones. Debido a que habrá espacio libre en los componentes de solución 106 después de que se hayan eliminado las superposiciones, se hace un esfuerzo para colocar partes individuales de una solución, aún no seleccionadas en ese momento, en los componentes de solución 106 ya seleccionados para maximizar el uso de energía en cada etapa. Si no hay componentes de solución 106 que contengan un punto clave en la base de datos 1004 que se está utilizando, se hace un esfuerzo para incluir otros componentes de solución 106. Finalmente, si no se puede encontrar ningún componente de solución 106 para las partes faltantes de la solución 108, y no se pueden colocar en los componentes de solución 106 ya seleccionados, se configura un componente de solución 106 propio para ellas. El procedimiento de corrección utilizado se implementa de modo que vea inmediatamente qué correcciones y costos establecen la compatibilidad, sin realizar los cambios por sí mismos. En la práctica, esto se lleva a cabo enumerando por separado los cambios necesarios después de que se hayan evaluado las correcciones 1012. Por último, se devuelve la cantidad necesaria de las mejores soluciones 1016. Todas las partes del algoritmo se pueden ejecutar en paralelo, sin esperas. El conjunto 1014 de los mejores componentes de la solución almacenada se puede ampliar con los componentes de las mejores soluciones totales compiladas.

La combinación de los componentes de solución 106 puede realizarse de diferentes maneras alternativas. Por ejemplo, la selección de los componentes de solución 106 que se van a combinar puede realizarse de diferentes maneras, o puede evaluarse un conjunto mayor de combinaciones diferentes antes de la selección. Los componentes de solución 106 que se van a almacenar pueden seleccionarse de varias maneras y pueden dividirse a su vez en partes más pequeñas. Pueden realizarse adiciones de más componentes de solución 106 a la vez, y las adiciones pueden realizarse simultáneamente desde una pluralidad de puntos de partida. Las correcciones pueden ser muy diversas y variables. El procedimiento también puede ser de autoaprendizaje, es decir, concentrarse primero en combinaciones costosas, por ejemplo. Alternativamente, puede identificarse en cada etapa el mejor componente de solución 106 que se va a combinar en lugar del primero adecuado, al mismo tiempo que anticipa los efectos de las selecciones realizadas, si es necesario, mediante una estrategia de tipo *look ahead* (de anticipación). Como parte de la compilación de las soluciones 108, también es posible realizar cargas que mejoren la decisión, con una heurística de mejora adecuada para el propósito, por ejemplo. Los componentes de solución seleccionados y no seleccionados 106 de la solución también pueden reemplazarse para potenciar la búsqueda.

Será evidente para un experto en la técnica que, a medida que avance la tecnología, la idea básica de la invención se puede implementar de muchas maneras diferentes. Por lo tanto, la invención y sus realizaciones no se limitan a los ejemplos descritos anteriormente, sino que pueden variar dentro del alcance de las reivindicaciones.

40

45

REIVINDICACIONES

1. Un aparato de procesamiento de datos (100) para la optimización de rutas en logística, que comprende una memoria (102) que incluye un código de programa informático (104), y al menos dos procesadores (110, 116), que están configurados para ejecutar el código de programa informático (104), comprendiendo dicho código de programa informático (104):
- un programa de componentes (112, 118) que se ejecuta en paralelo en los al menos dos procesadores (110, 116) para formar respectivos componentes de solución que comprenden una pluralidad de puntos (902-940), y para almacenar los componentes de solución formados en la memoria (102), en el que los componentes de solución comprenden una pluralidad de rutas individuales (106; 950, 952, 954, 956) que comprenden la pluralidad de puntos;
- caracterizado por:**
- un programa de solución (114, 120) se ejecuta en paralelo en los al menos dos procesadores (110, 116), en el que a cada uno de los al menos dos procesadores (110, 116) se le asigna un punto de inicio diferente, para generar una solución (108) mediante la adición de una pluralidad de componentes de solución leídos secuencialmente desde la memoria (102), determinándose los componentes de solución mediante:
- determinar un punto clave, el punto clave proporcionado en un componente de solución y definir un punto que proporcione un costo más bajo de proporcionar dicha ruta logística si se incorpora en un componente de solución adicional,
- seleccionar una primera solución (950) y determinar un punto clave (928) para la primera solución (950);
- determinar un segundo componente de solución (952) que comprende el punto clave (928) y que además no comprende un punto común con un primer componente de solución (950) de manera que el primer componente de solución (950) no tiene puntos superpuestos con el segundo componente de solución (952);
- determinar un punto clave adicional (932) para la primera solución (950) y determinar un componente de solución no utilizado (954, 956) que comprende el punto clave adicional (932) y la menor cantidad posible de puntos superpuestos con los componentes de solución (950, 952);
- seleccionar componentes de solución hasta que se hayan seleccionado todos los puntos (902-940) o no se puedan encontrar más componentes, incluido un punto clave, en la biblioteca de componentes de solución; y
- en el que el programa de solución (114, 120) coloca adicionalmente, después de la eliminación de superposiciones, puntos de solución individuales (938, 940) todavía no seleccionados en ese momento, en los componentes de solución (956, 962) de la solución (108), que ya han sido seleccionados.
2. El aparato de procesamiento de datos según la reivindicación 1, en el que el programa de solución (114, 120) genera adicionalmente la solución (108) de manera que en cada procesador (110, 116) la solución (108) se genera independientemente.
3. El aparato de procesamiento de datos según la reivindicación 1, en el que el programa de componentes (112, 118) divide adicionalmente los puntos (600, 602, 604, 606, 608, 610, 612, 614, 616, 618) en al menos dos subconjuntos específicos de procesador (800, 810) y genera los componentes de solución (802, 804, 812, 814) de ellos por separado para cada procesador (110, 116), y el programa de solución (114, 120) genera adicionalmente la solución (108) de tal manera que en cada uno de los al menos dos procesadores (110, 116) la solución (820, 822) se genera específicamente para el procesador mediante el uso de componentes de solución específicos del procesador (802, 804, 812, 814).
4. El aparato de procesamiento de datos según una cualquiera de las reivindicaciones precedentes, en el que el programa de solución (114, 120) genera además, por último, un componente de solución específico para aquellos puntos faltantes de la solución (108) para los que no se puede encontrar ningún componente de solución y que no se pueden colocar en los componentes de solución de la solución (108) que ya han sido seleccionados.
5. El aparato de procesamiento de datos según una cualquiera de las reivindicaciones precedentes, en el que:
- una evaluación de las soluciones (108) se basa en al menos uno de los siguientes: longitud de las rutas, capacidad de uso de las rutas, tiempo total de las rutas, asignación justa de las rutas, idoneidad de las rutas para los límites de tiempo, compatibilidad de las rutas, prioridad de las rutas;
- los puntos clave se obtienen determinando puntos no enrutados; y
- las correcciones están basadas en la colocación y/o cambios de puntos individuales.

5 6. El aparato de procesamiento de datos según una cualquiera de las reivindicaciones precedentes, en el que un primer procesador de los al menos dos procesadores (110, 116) comprende un microprocesador, un núcleo de procesador o un procesador de gráficos, y un segundo procesador de los al menos dos procesadores (110, 116) comprende un microprocesador, un núcleo de procesador o un procesador de gráficos.

10 7. Un medio de distribución legible por ordenador (130) que comprende un código de programa informático (104), ejecutable en al menos dos procesadores (110, 116), comprendiendo dicho código de programa informático (104):

15 un programa de componentes (112, 118) que se ejecuta en paralelo en los al menos dos procesadores (110, 116) para formar respectivos componentes de solución que comprenden una pluralidad de puntos (902-940), y para almacenar los componentes de solución formados en la memoria (102), en el que los componentes de solución comprenden una pluralidad de rutas individuales (106; 950, 952, 954, 956) que comprenden la pluralidad de puntos;

**caracterizado por:**

20 un programa de solución (114, 120) se ejecuta en paralelo en los al menos dos procesadores (110, 116), en el que a cada uno de los al menos dos procesadores (110, 116) se le asigna un punto de inicio diferente, para generar una solución (108) mediante la adición de una pluralidad de componentes de solución leídos secuencialmente desde la memoria (102), estando el componente de solución determinado al:

25 determinar un punto clave, el punto clave proporcionado en un componente de solución y definir un punto que proporcione un coste más bajo de proporcionar dicha ruta logística si se incorpora en un componente de solución adicional,

seleccionar una primera solución (950) y determinar un punto clave (928) para la primera solución (950);

30 determinar un segundo componente de solución (952) que comprende el punto clave (928) y que además no comprende un punto común con un primer componente de solución (950) de manera que el primer componente de solución (950) no tiene puntos superpuestos con el segundo componente de solución (952);

35 determinar un punto clave adicional (932) para la primera solución (950) y determinar un componente de solución no utilizado (954, 956) que comprende el punto clave adicional (932) y la menor cantidad posible de puntos superpuestos con los componentes de solución (950, 952);

seleccionar los componentes de solución hasta que se hayan seleccionado todos los puntos (902-940) o no se puedan encontrar más componentes, incluido un punto clave, en la biblioteca de componentes de solución; y

40 en el que el programa de solución (114, 120) coloca adicionalmente, después de la eliminación de superposiciones, puntos de solución individuales (938, 940) todavía no seleccionados en ese momento, en los componentes de solución (956, 962) de la solución (108), que ya han sido seleccionados.

45

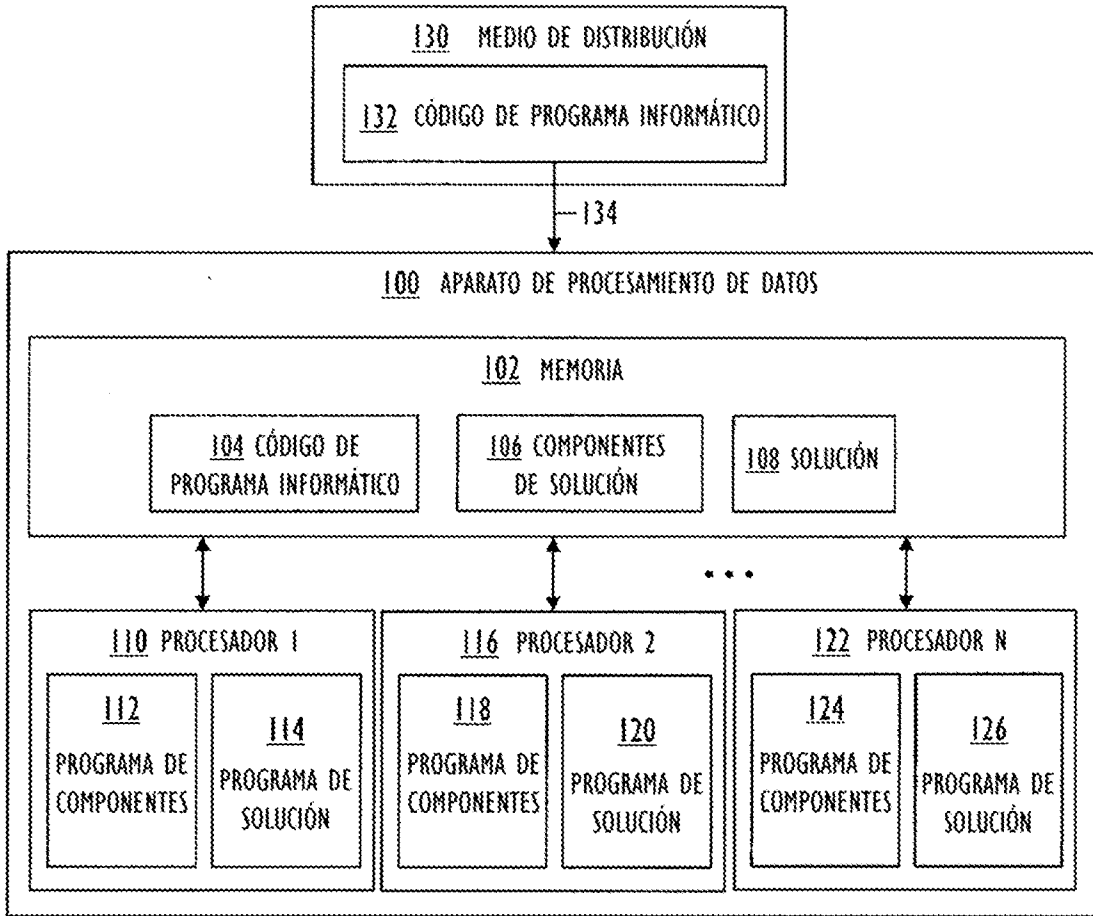


FIG. 1

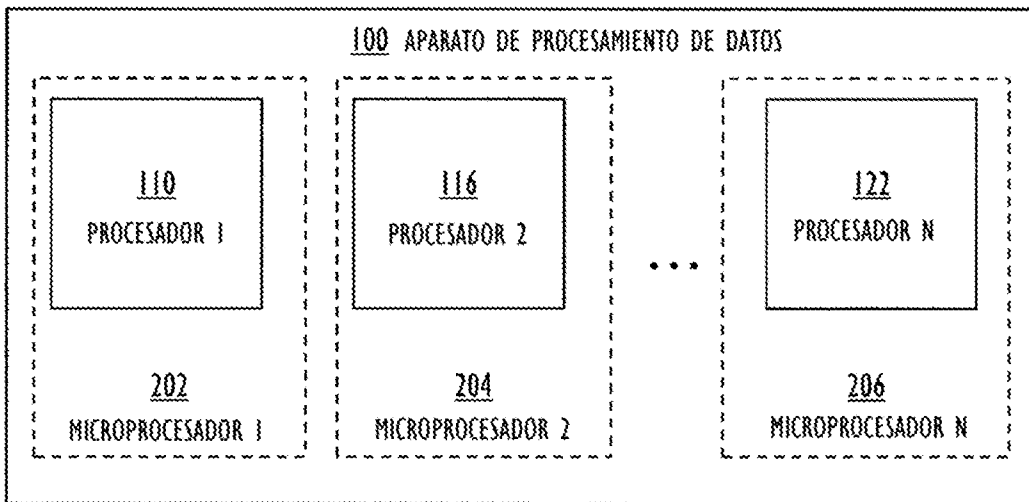


FIG. 2

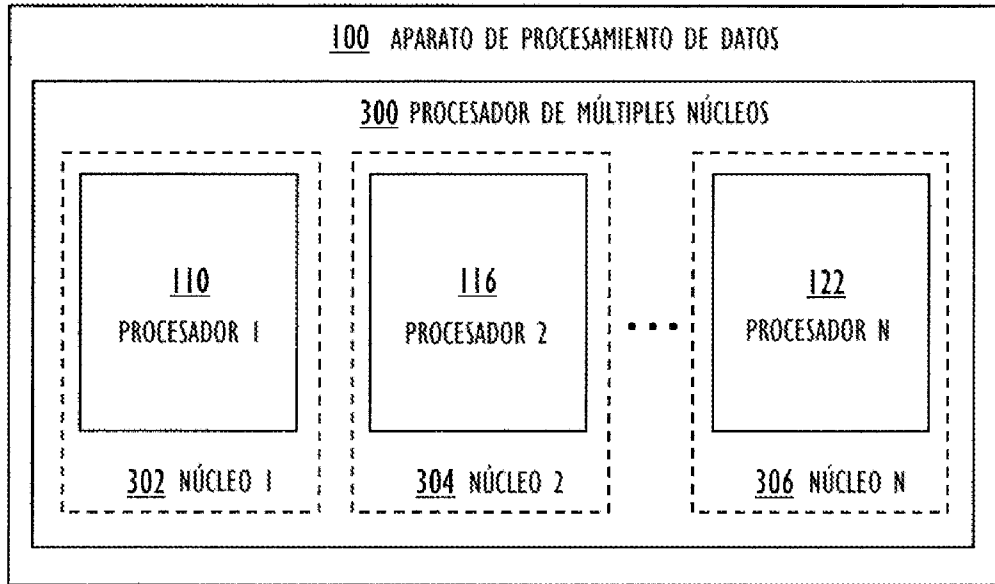


FIG. 3

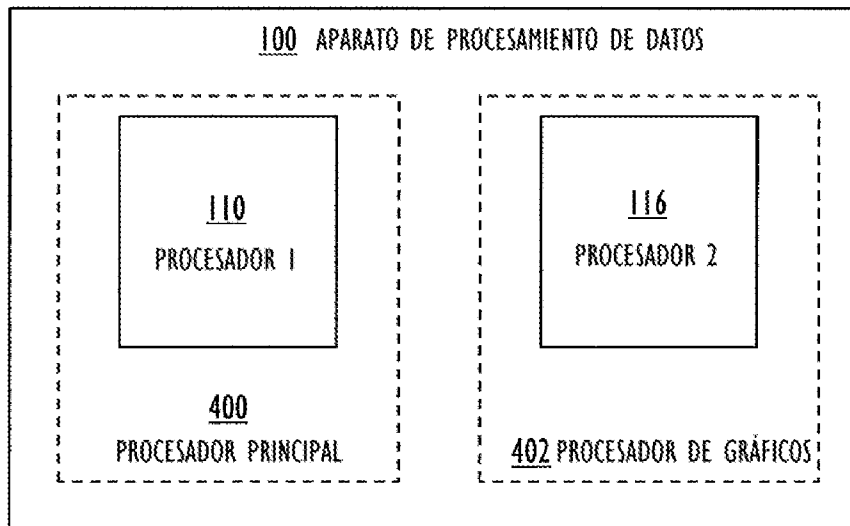


FIG. 4

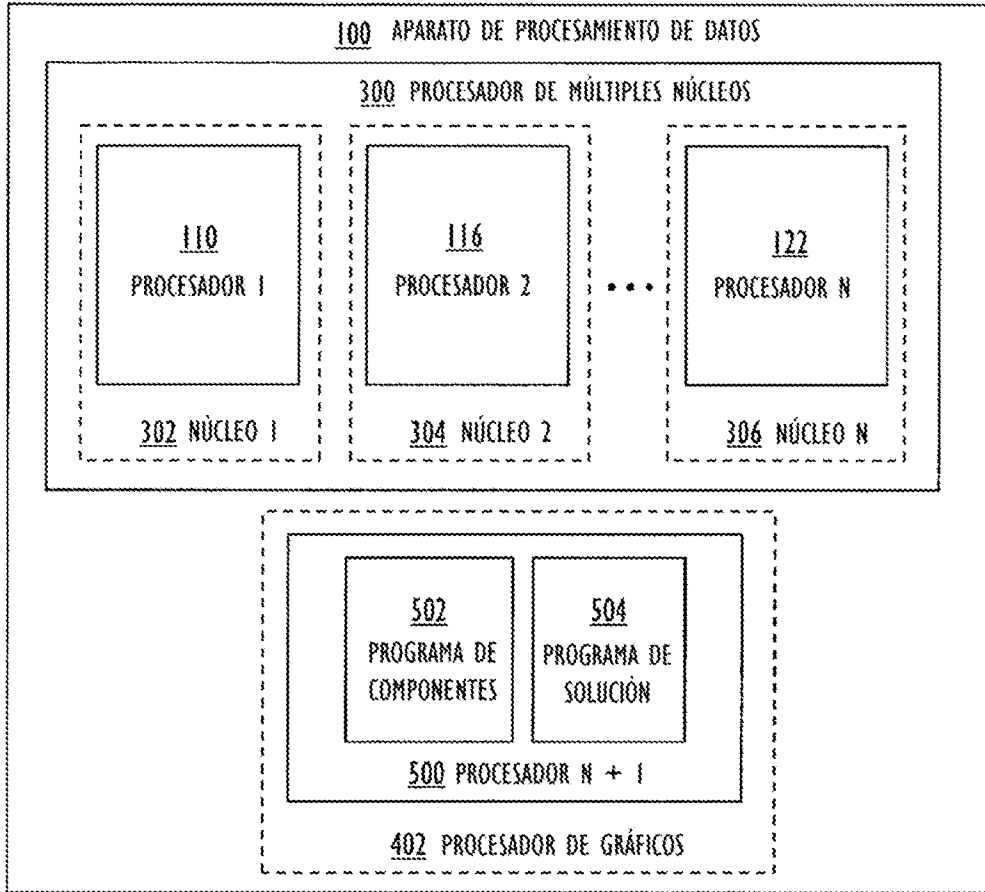


FIG. 5

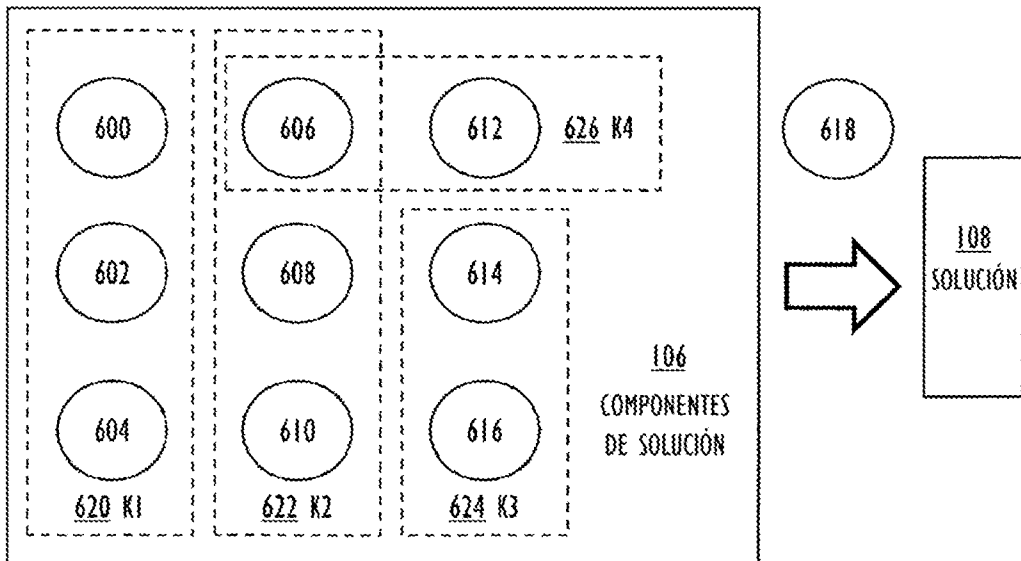


FIG. 6

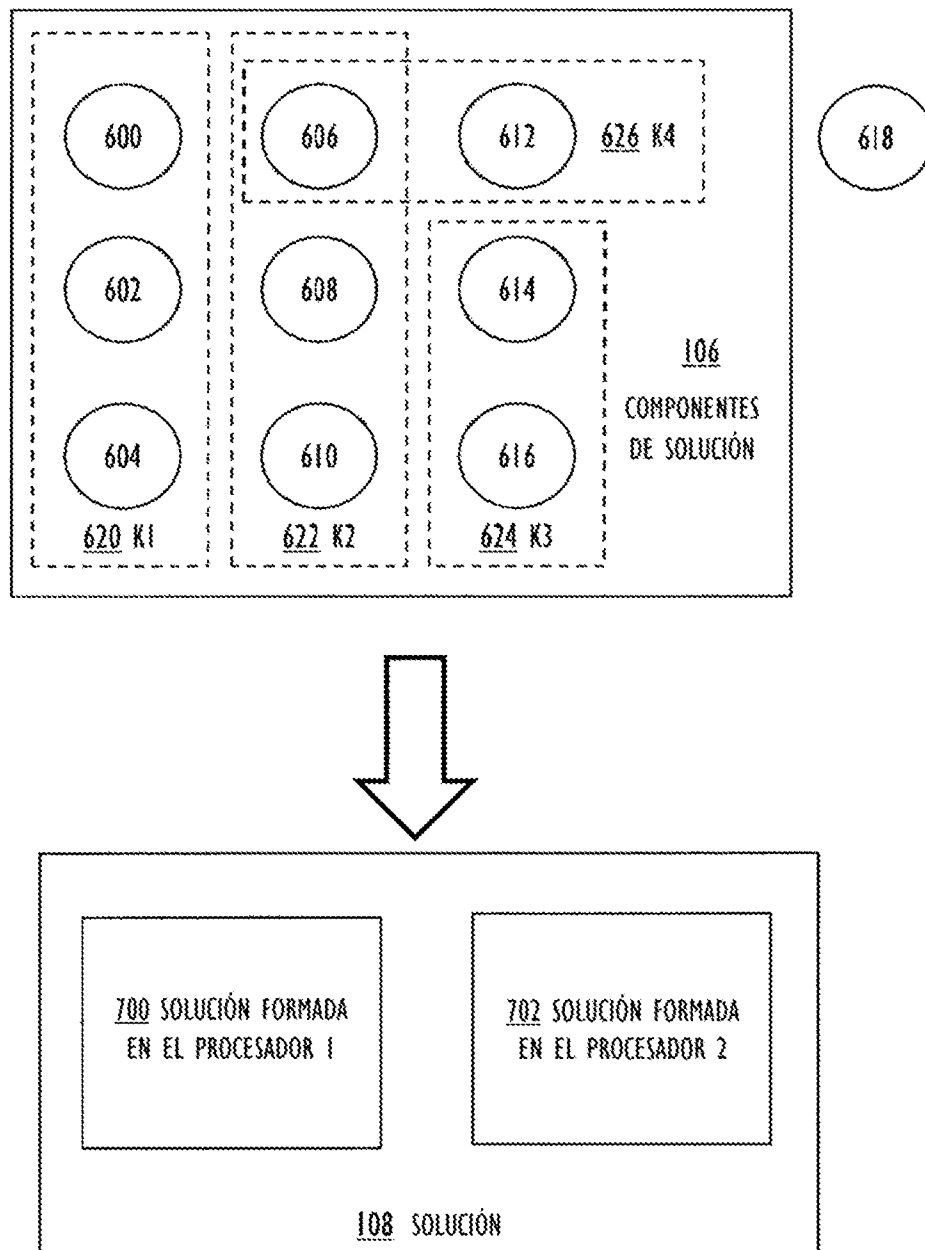


FIG. 7

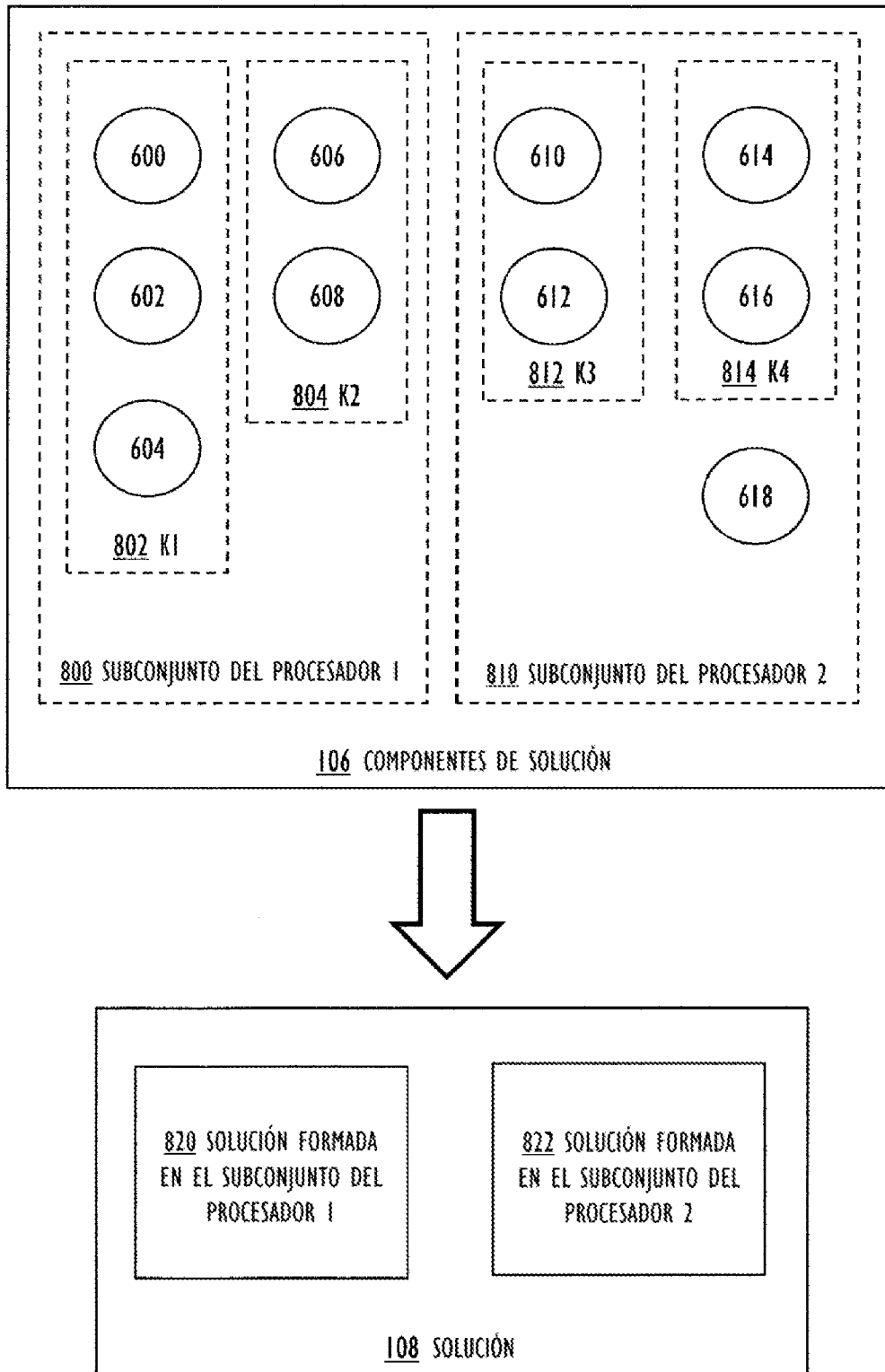


FIG. 8

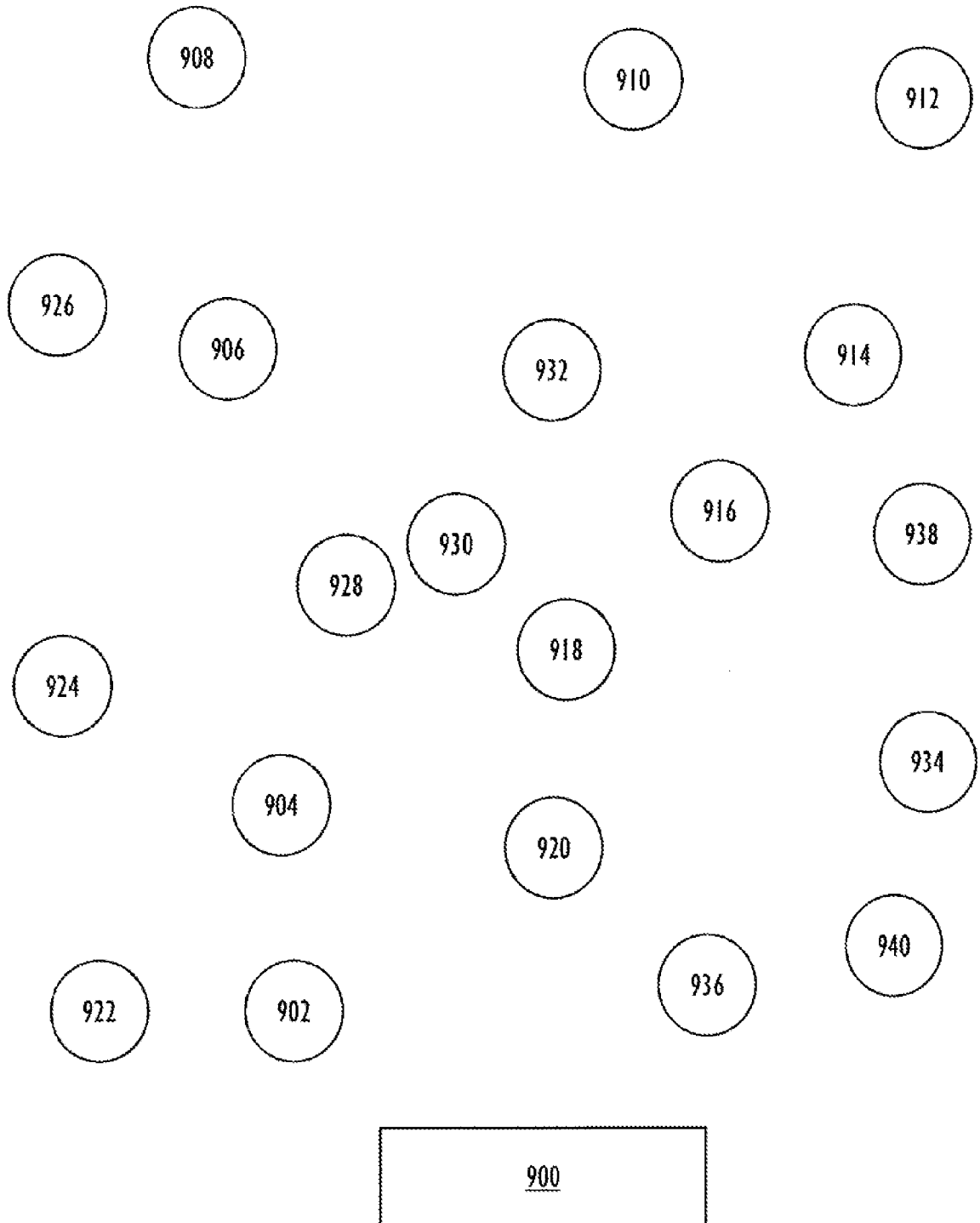


FIG. 9A

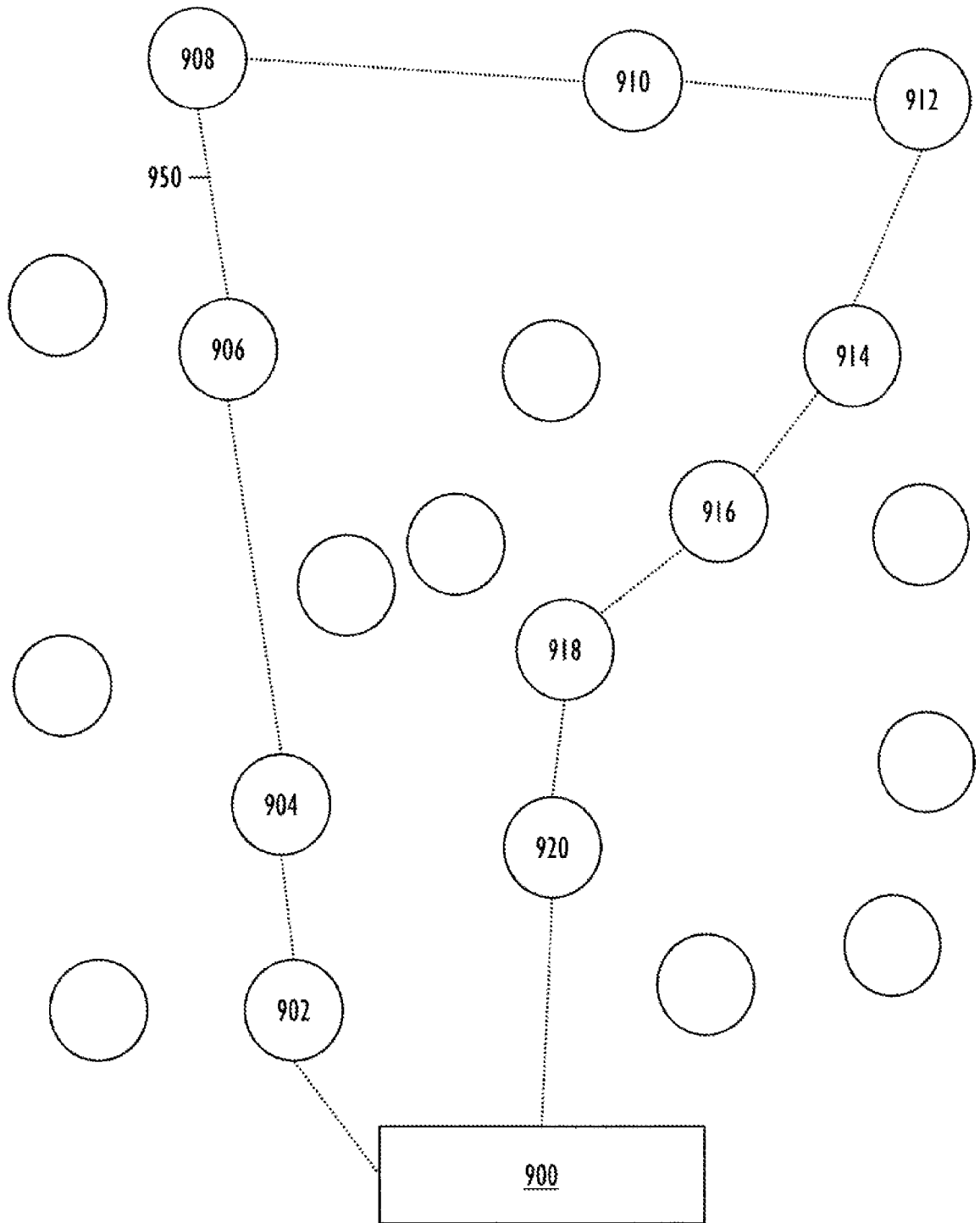


FIG. 9B

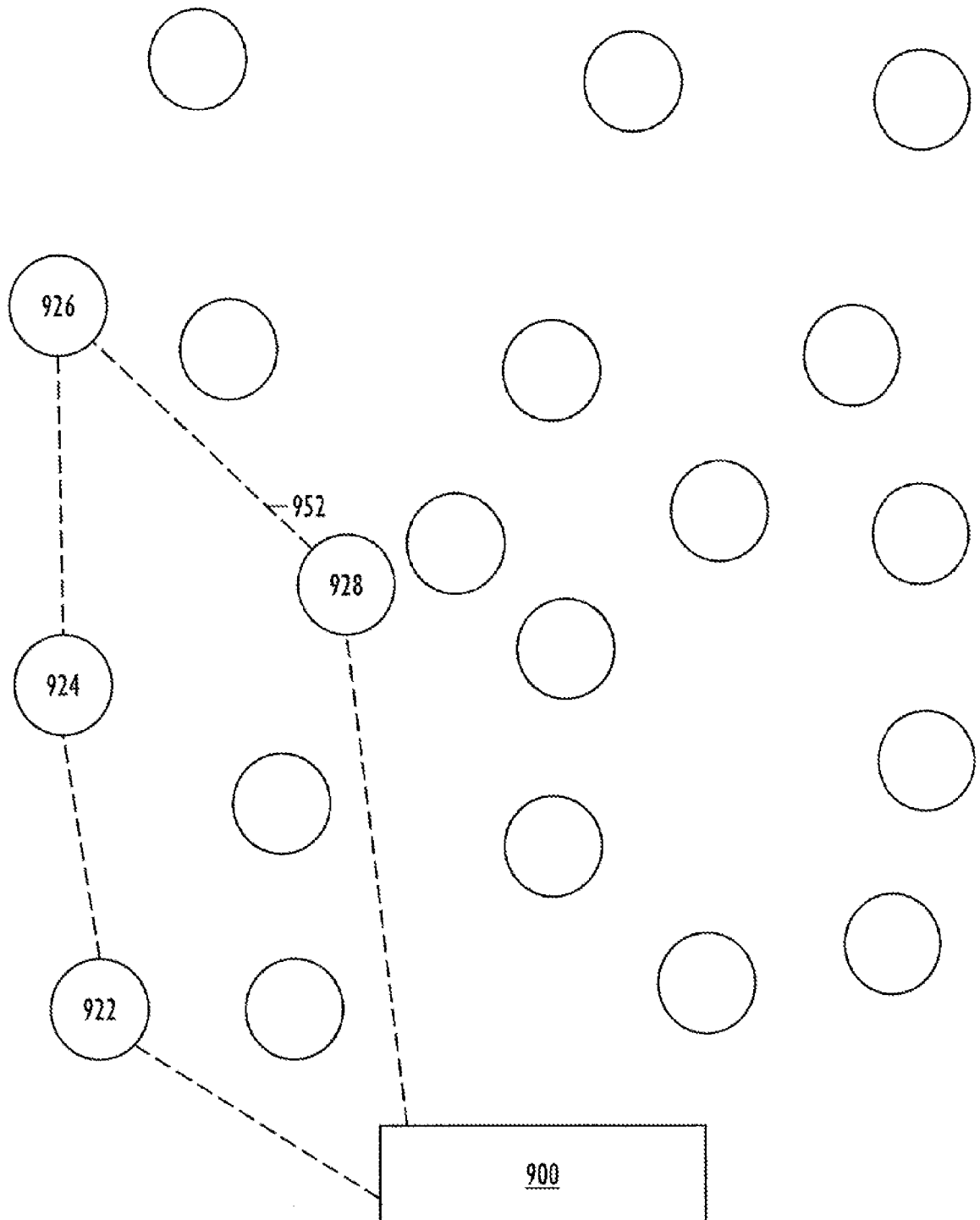


FIG. 9C

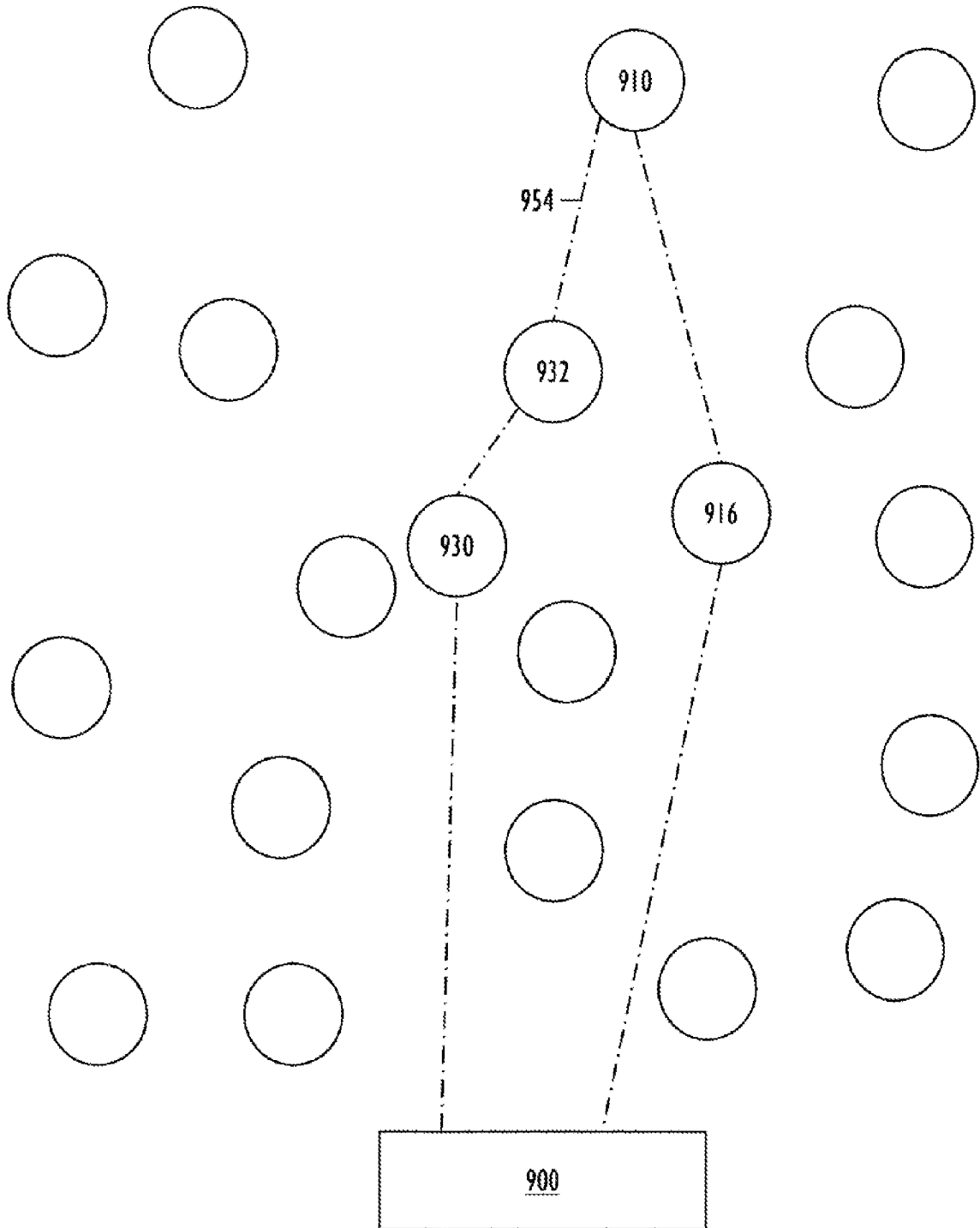


FIG. 9D

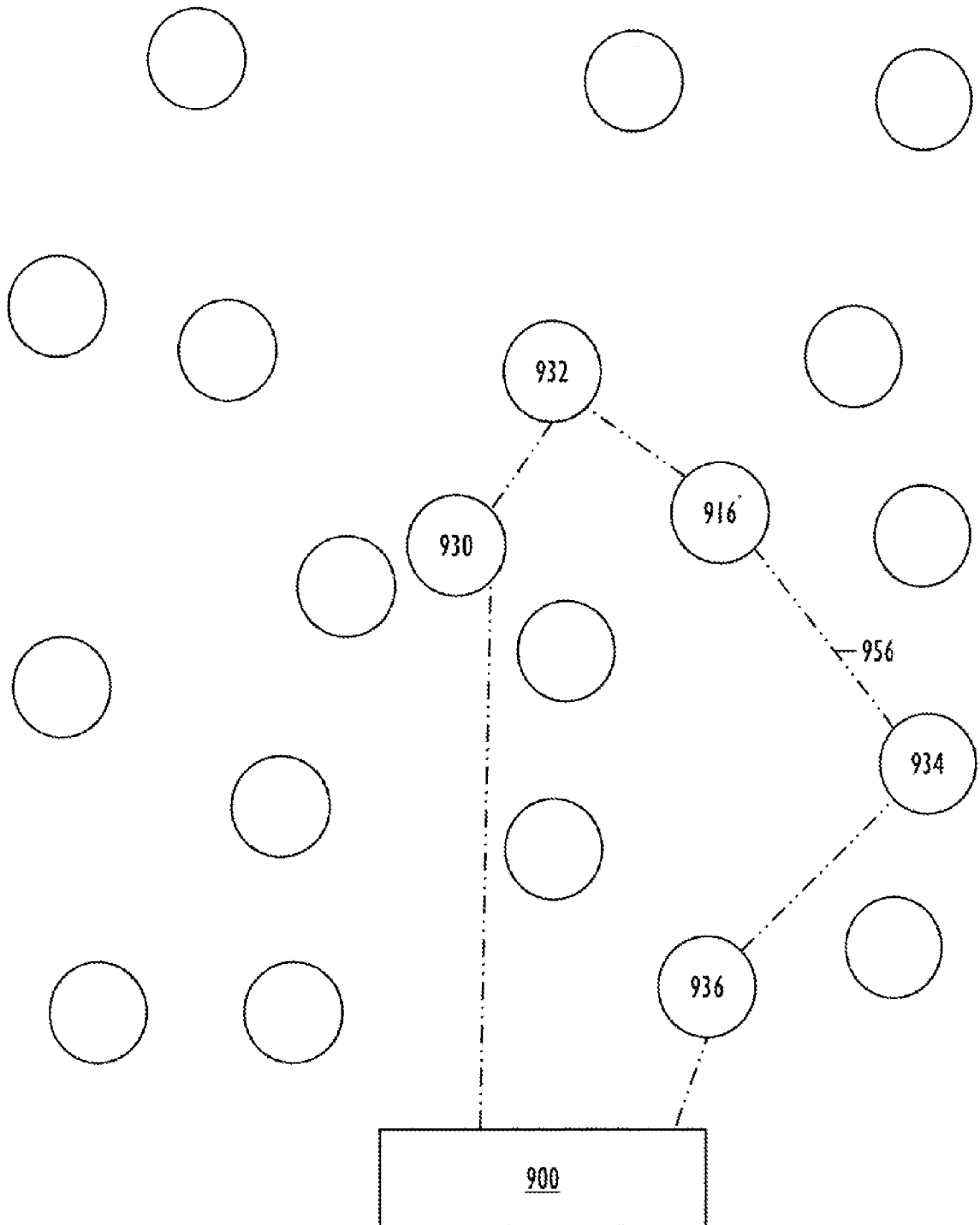


FIG. 9E

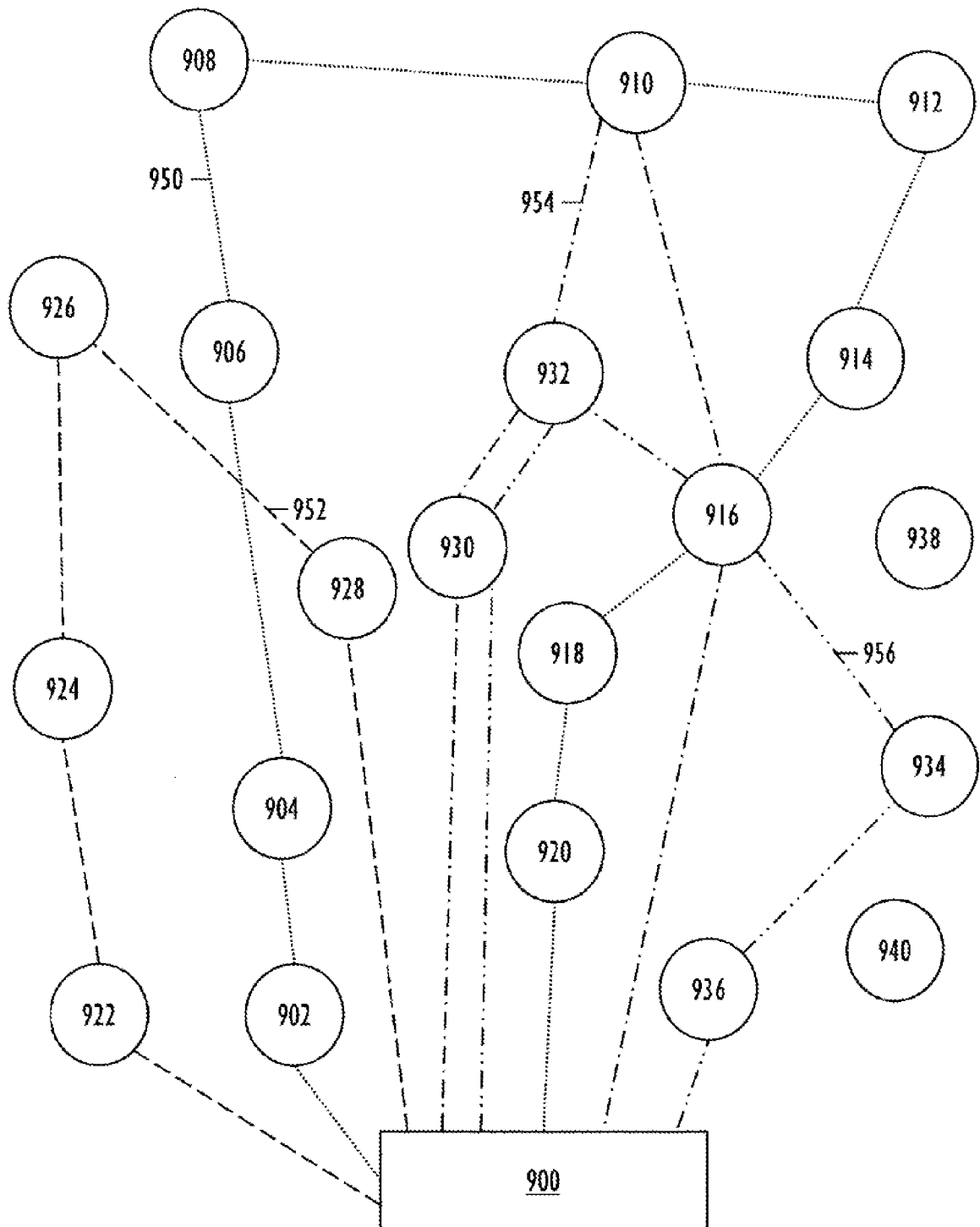


FIG. 9F

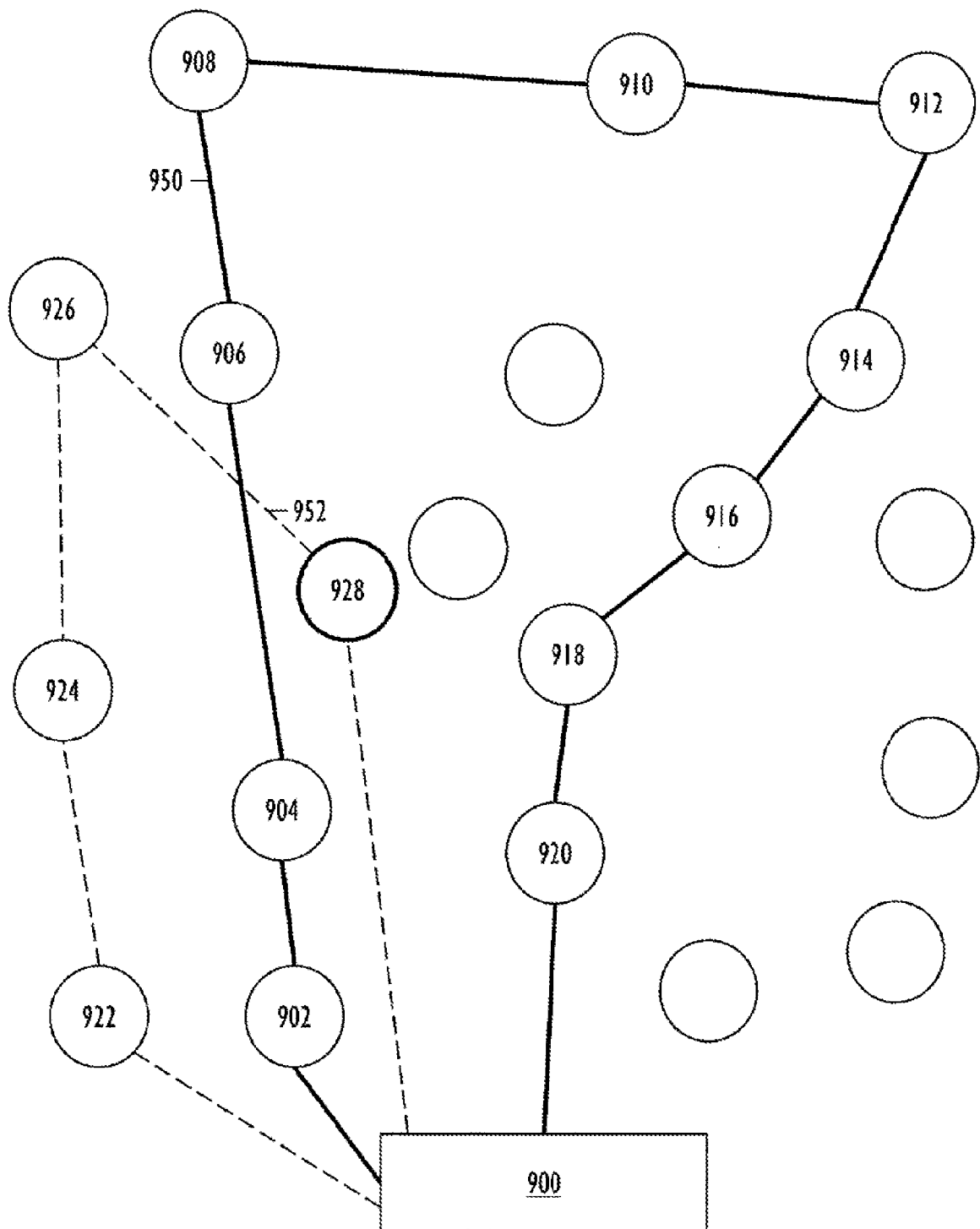


FIG. 9G

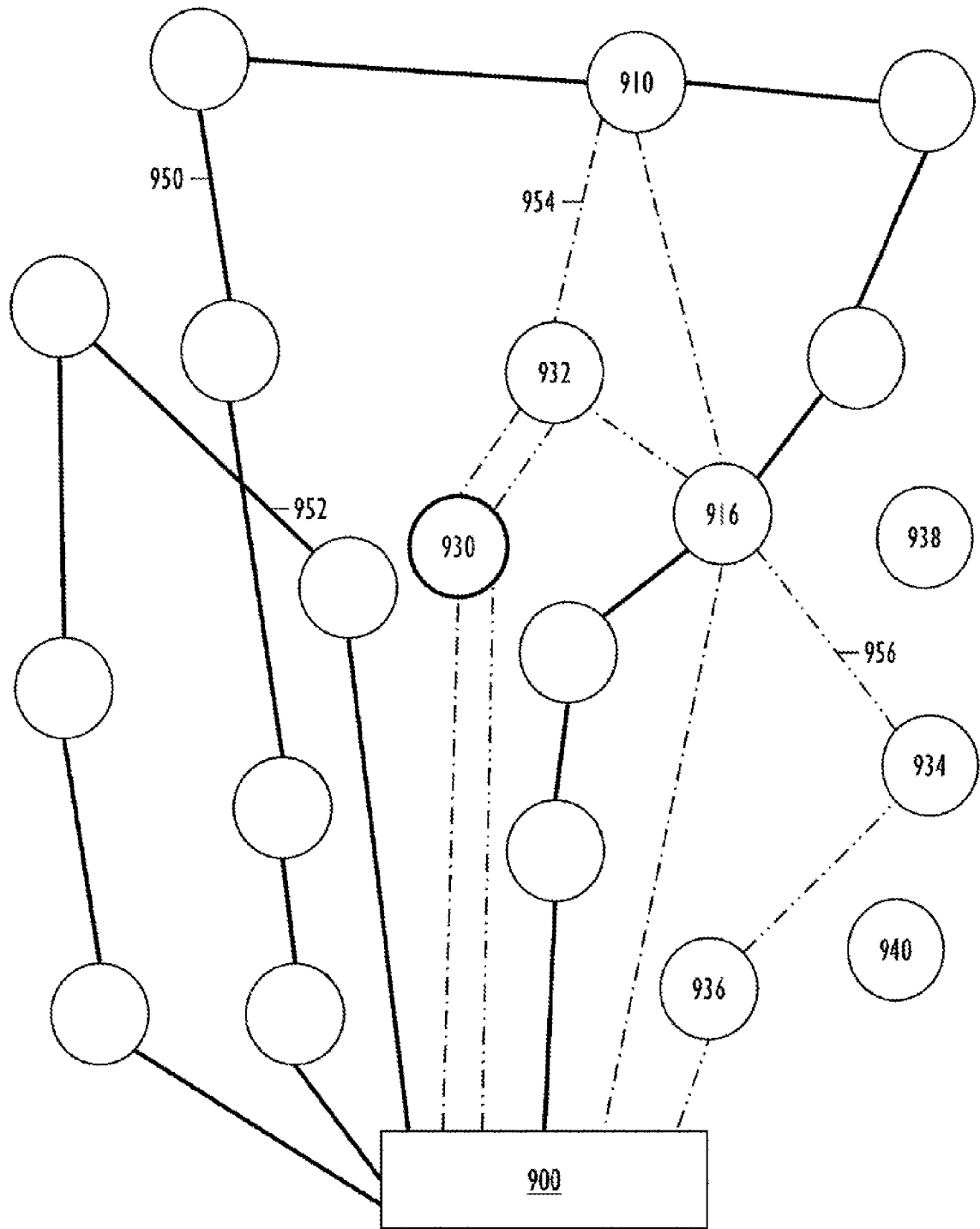


FIG. 9H

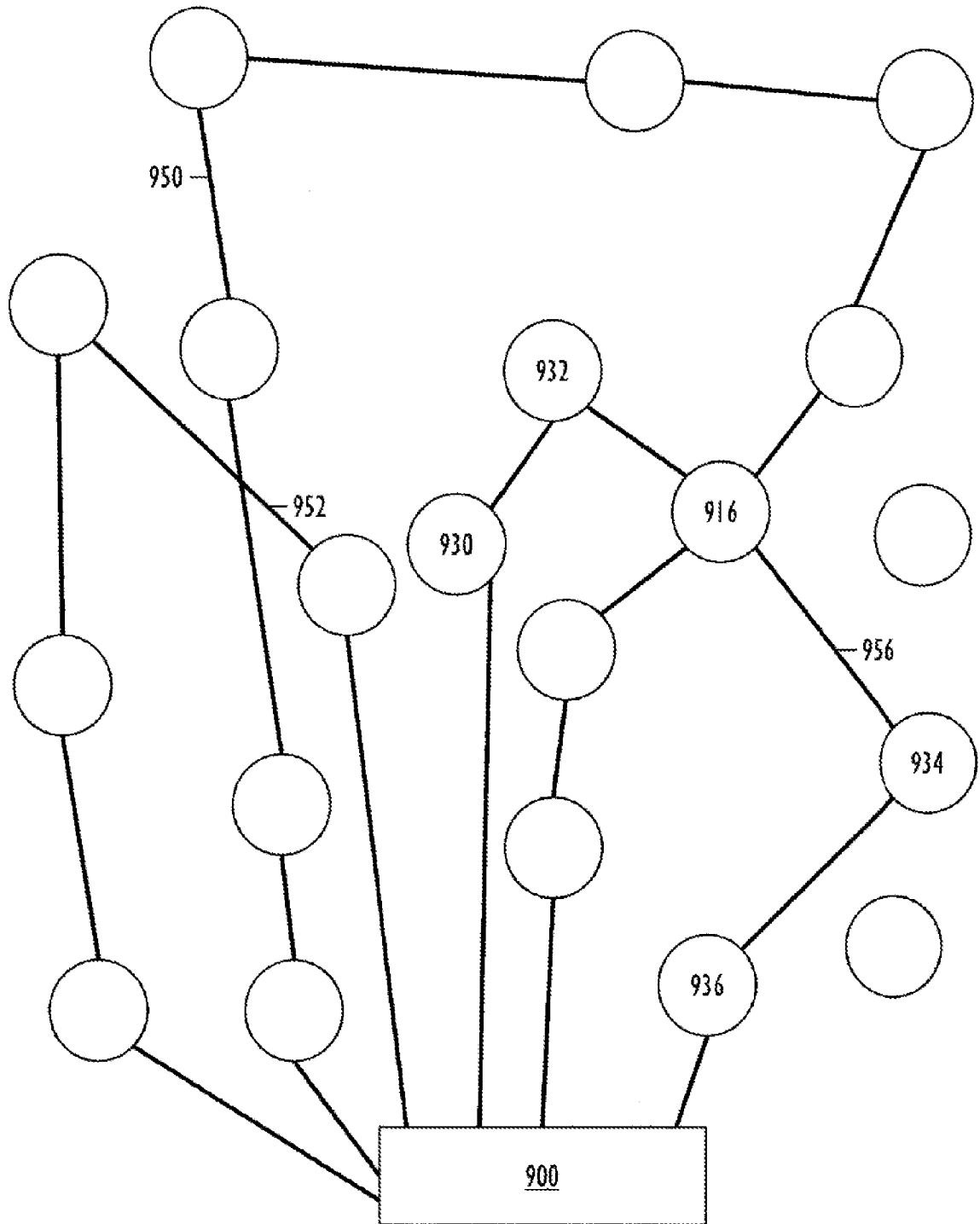


FIG. 91

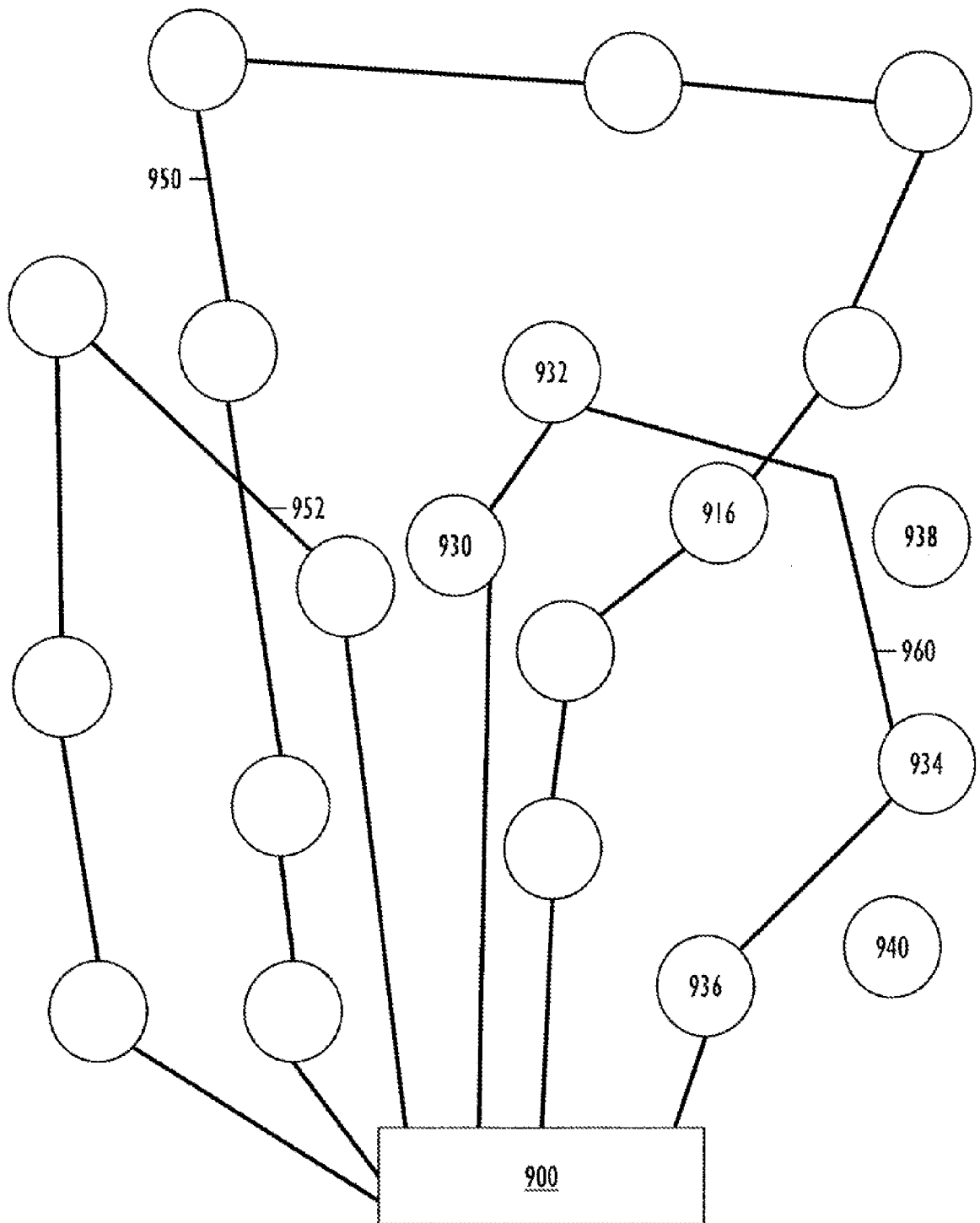


FIG. 9J

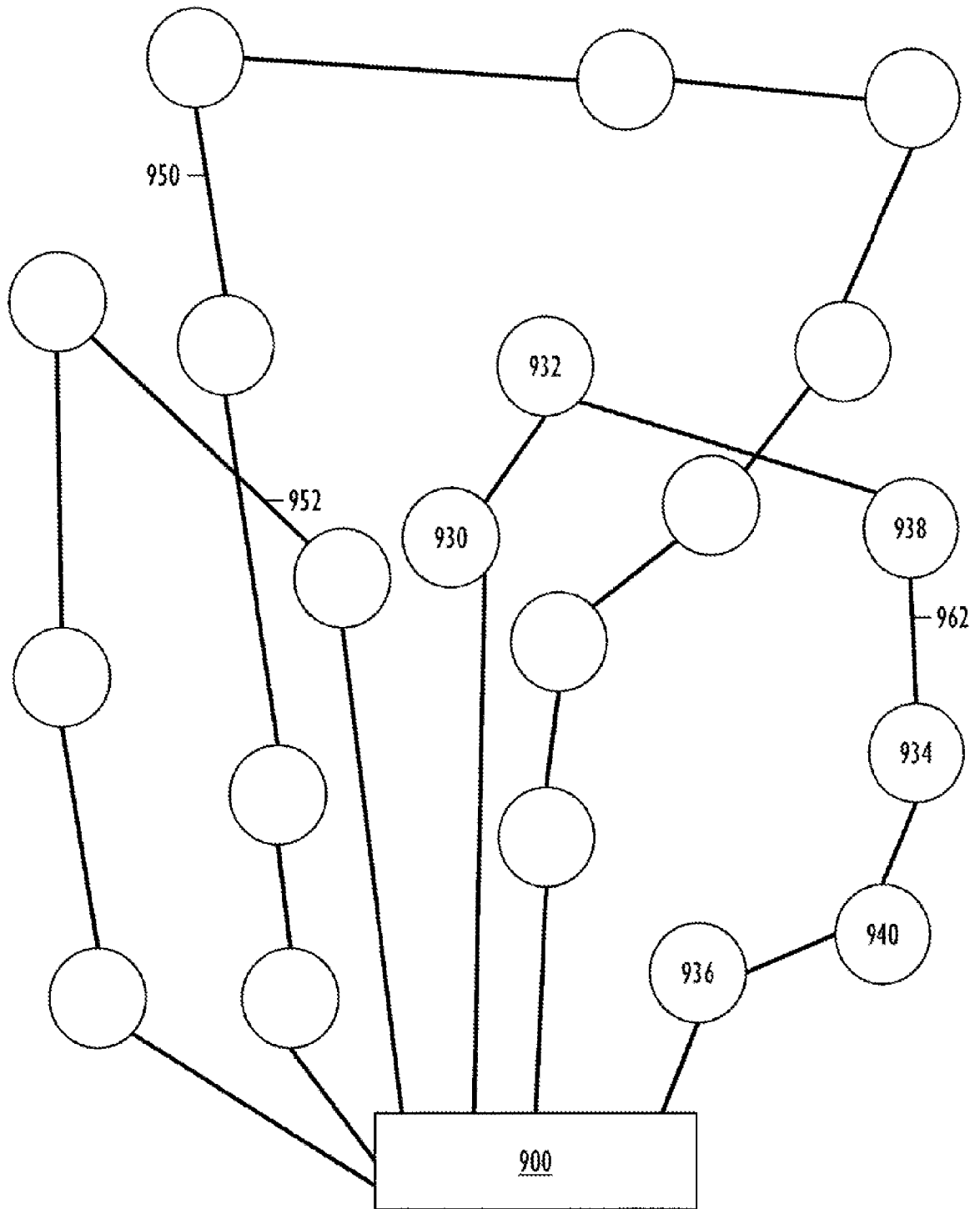


FIG. 9K

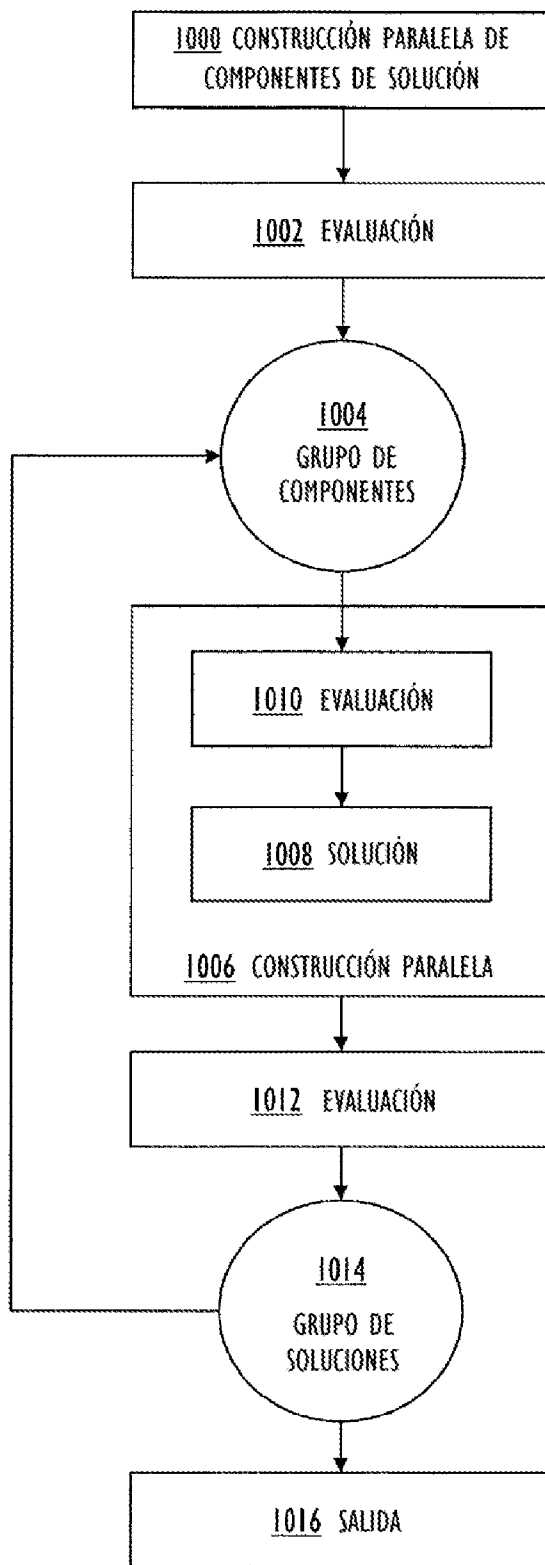


FIG. 10