

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 390 489**

51 Int. Cl.:
H03M 13/00 (2006.01)
H03M 13/11 (2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

- 96 Número de solicitud europea: **07701215 .1**
96 Fecha de presentación: **08.01.2007**
97 Número de publicación de la solicitud: **1977524**
97 Fecha de publicación de la solicitud: **08.10.2008**

54 Título: **Códigos de comprobación de paridad de baja densidad (LDPC) de tasa compatible con H-ARQ para aplicaciones de alto rendimiento**

30 Prioridad:
06.01.2006 US 756758 P
03.07.2006 US 480597

45 Fecha de publicación de la mención BOPI:
13.11.2012

45 Fecha de la publicación del folleto de la patente:
13.11.2012

73 Titular/es:
QUALCOMM INCORPORATED (100.0%)
5775 MOREHOUSE DRIVE
SAN DIEGO, CALIFORNIA 92121, US

72 Inventor/es:
EL-KHAMY, MOSTAFA;
HOU, JILEI y
BHUSHAN, NAGA

74 Agente/Representante:
FÀBREGA SABATÉ, Xavier

ES 2 390 489 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

Códigos de comprobación de paridad de baja densidad (LDPC) de tasa compatible con H-ARQ para aplicaciones de alto rendimiento

ANTECEDENTES

5 **Campo**

La presente invención se refiere en general a las comunicaciones y, en particular, a codificación de tasa compatible con solicitudes automáticas híbridas (H-ARQ) para aplicaciones de alto rendimiento.

Antecedentes

10 En general, en el caso de transmisión de datos, un receptor observa cada bit recibido en presencia de ruido o distorsión, obteniéndose solamente una indicación del valor del bit. En estas circunstancias, los valores observados se interpretan como una fuente de bits de "blandos". Un bit blando indica una estimación preferida del valor del bit, es decir un uno o un cero, junto con alguna indicación de la fiabilidad de esa estimación. Aunque el número de errores puede ser relativamente bajo, un pequeño número de errores o nivel de distorsión puede dar como resultado que los datos sean inservibles o, en el caso de errores de transmisión, puede ser necesario retransmitir los datos.

15 Con el fin de proporcionar un medio para comprobar los errores y, en algunos casos, corregir los errores, pueden codificarse datos binarios para introducir una redundancia cuidadosamente diseñada. La codificación de una unidad de datos produce lo que se denomina comúnmente palabra de código. Debido a su redundancia, una palabra de código incluye normalmente más bits que la unidad de entrada de datos a partir de la cual se generó la palabra de código.

20 Cuando se reciben o se procesan señales que surgen a partir de palabras de código transmitidas, la información redundante incluida en la palabra de código observada en la señal puede utilizarse para identificar y/o corregir errores en o eliminar distorsión de la señal recibida con el fin de recuperar la unidad de datos original. Tal comprobación y/o corrección de errores pueden implementarse como parte de un proceso de decodificación. En ausencia de errores, o en el caso de errores o una distorsión que pueden corregirse, la decodificación puede utilizarse para recuperar a partir de los datos fuente que se están procesando la unidad de datos original que se codificó. En el caso de errores que no pueden recuperarse, el proceso de decodificación puede producir alguna indicación de que los datos originales no pueden recuperarse completamente. Estas indicaciones de fallos de decodificación pueden utilizarse para iniciar retransmisiones de los datos.

25 30 Los sistemas de comunicaciones necesitan funcionar normalmente a varias tasas diferentes. Una manera de hacer que la implementación sea lo más sencilla posible y proporcionar la codificación y decodificación a las diferentes tasas es utilizar códigos de comprobación de paridad de baja densidad (LDPC). En particular, los códigos LDPC se utilizan para generar códigos de mayor tasa perforando códigos de menor tasa. Con los sistemas de comunicaciones que utilizan adaptación de tasa puede ser necesario ajustar la tasa de transmisión de datos según las condiciones y necesidades del sistema. La adaptación de tasa permite que un sistema de comunicaciones, y sus componentes, adapten de manera flexible y eficaz la tasa de transmisión de datos a las condiciones de canal actuales. Por ejemplo, los diseños habituales de corrección de errores seleccionan un código fijo, presentando el código una determinada tasa y capacidad de corrección. Para añadir flexibilidad con el fin de manejar diferentes cantidades de datos que presentan diferentes necesidades de protección contra los errores, la tasa se adapta a condiciones de canal variables en el tiempo, además de compensar parámetros que no se conocen en profundidad.

35 40 Para una codificación flexible de canal, los bits de datos pueden agruparse en bloques de tamaño variable, y estos bloques pueden codificarse con diferentes cantidades de redundancia. En lugar de utilizar varios códigos de corrección diferentes de errores para codificar los diferentes grupos de bits, es deseable utilizar un único código padre que pueda adaptarse a varias tasas. Esto se denomina codificación de tasa compatible. Con el rápido desarrollo de los sistemas de comunicaciones móviles, existe una necesidad en la técnica de códigos LDPC de tasa compatible adecuados para aplicaciones de ARQ híbrido de alto rendimiento.

45 50 Los procedimientos LDPC pueden recuperar palabras de código en entornos muy ruidosos. Dos ejemplos de modelos de canal de comunicaciones son el canal de borrado binario (BEC) y el canal simétrico binario (BSC). Estos canales se ilustran en las Figuras 1A y 1B. La entrada es binaria ya que comprende bits que tienen el valor 0 ó 1.

Con respecto a las salidas, el BEC tiene una salida de 0, 1 o de borrado. La probabilidad de que un bit se haya transferido correctamente es $1-p$ y la probabilidad de que un bit se haya transferido de manera errónea, es decir, se haya borrado, es p . Con el BSC, la salida es 0 ó 1. La probabilidad de que un bit se haya transferido correctamente

es $1-p$ y la probabilidad de que un bit se haya transferido de manera errónea, es decir, se haya recibido un cero cuando se envió un uno y viceversa, es p .

Los códigos LDPC son códigos lineales que se generan a partir de grafos bipartitos dispersos. Un grafo bipartito, también llamado bigrafo, es un conjunto de vértices de grafo divididos en dos conjuntos disjuntos de manera que ningún par de vértices de grafo del mismo conjunto son adyacentes. Un grafo bipartito es un caso especial de un grafo k -partito, donde $k = 2$. Si hay p y g vértices de grafo en dos conjuntos, el grafo bipartito completo se denota $K_{p,q}$. Disperso quiere decir que el código LDPC es una clase especial de códigos de bloques lineales cuya matriz de comprobación de paridad H tiene una baja densidad de unos. Esta dispersión permite una decodificación de baja complejidad, lo que da lugar a una implementación sencilla. La Figura 2 es un ejemplo de un grafo de este tipo. El grafo contiene V nodos variables, v_i en lado izquierdo, y C nodos de comprobación, c_j en el lado derecho. Los V nodos variables también pueden denominarse nodos sistemáticos o de mensaje.

El grafo bipartito puede representarse mediante una matriz de adyacencia simplificada H con C filas y V columnas. Véase la Figura 3. La Figura 3 es un ejemplo de una matriz binaria dispersa H de comprobación de paridad $C \times V$. Si un nodo de comprobación j -ésimo está conectado a un nodo variable i -ésimo mediante una arista, es decir, los dos nodos son adyacentes, entonces hay un uno en la columna i -ésima y en la fila j -ésima de una matriz de comprobación de paridad H . Es decir, la intersección de una fila i -ésima y una columna j -ésima contiene un "1" cuando una arista une el vértice correspondiente y un "0" cuando no hay ninguna arista. Se dice que un nodo variable y un nodo de restricción (o de comprobación) son adyacentes si están conectados mediante una arista en el grafo. Las aristas, E , del grafo conectan nodos variables con nodos de restricción o de comprobación, y cada arista corresponde a una entrada distinta de cero en la matriz de comprobación de paridad. Cada fila de la matriz H corresponde a un nodo de comprobación y representa una comprobación de paridad y cada columna corresponde a uno de los nodos variables y representa un símbolo demodulado. Puesto que hay 5 nodos variables y 3 nodos de comprobación, la matriz incluye 5 columnas y 3 filas.

El número de símbolos demodulados o de nodos variables N es la longitud del código LDPC. El número de elementos distintos de cero en una fila (columna) se define como el peso d_c (d_v) de la fila (columna). Un grafo o código regular es uno en el que todos los nodos variables tienen el mismo grado, por ejemplo j , y todos los nodos de restricción tienen el mismo grado, por ejemplo k . En este caso, decimos que el código es un código regular (j, k). Por otro lado, un código irregular tiene nodos de restricción y/o nodos variables de grados diferentes. Por ejemplo, algunos nodos variables pueden ser de grado 4, otros de grado 3 y otros de grado 2.

A cada nodo variable se le asocia un bit de palabra de código. Una secuencia de bits asociados uno a uno con la secuencia de nodos variables es una palabra de código del código si y solo si para cada nodo de restricción, los bits adyacentes de la restricción (a través de su asociación con nodos variables) suman cero módulo dos, es decir, comprenden un número par de unos. En algunos casos, parte de estos bits pueden perforarse o pueden conocerse, como se describe posteriormente en mayor detalle. La perforación es la acción de eliminar bits de una palabra de código para proporcionar, en efecto, una palabra de código más corta. En el caso de grafos LDPC, esto significa que algunos de los nodos variables del grafo corresponden a bits que no se transmiten en realidad.

Los decodificadores y algoritmos de decodificación utilizados para decodificar palabras de código LDPC funcionan intercambiando mensajes dentro del grafo a lo largo de las aristas y actualizando estos mensajes llevando a cabo cálculos en los nodos basados en los mensajes entrantes. Tales procedimientos se denominarán generalmente procedimientos de paso de mensajes. Cada nodo variable del grafo está dotado inicialmente de un bit blando, denominado valor recibido, que indica una estimación del valor del bit asociado determinado mediante observaciones de, por ejemplo, el canal de comunicaciones. De manera ideal, las estimaciones de los diferentes bits son estadísticamente independientes. Esta idea puede violarse en la práctica. Una palabra recibida comprende una colección de valores recibidos.

El grado de un nodo se refiere al número de aristas conectadas a ese nodo. Esta característica se ilustra en la matriz H mostrada en la Figura 3, donde el número de aristas que inciden en v_i es igual al número de unos en la columna i , y se denomina grado de nodo variable $d(v_i)$. Asimismo, el número de aristas conectadas a c_j es igual al número de unos en la fila j y se denomina grado de nodo de comprobación $d(c_j)$.

En aplicaciones de ARQ híbrido, todos los símbolos de datos se transmiten durante la primera transmisión con un código de tasa alta. Si el paquete no se decodificó de manera satisfactoria, el transmisor enviará más símbolos de redundancia para aumentar la fiabilidad del paquete ya transmitido. Estos símbolos de redundancia cuando se combinan con el paquete recibido anteriormente forman otra palabra de código de una tasa relativamente inferior ya que los bits de información se perforan mediante los símbolos de redundancia añadidos. El transmisor enviará de manera incremental símbolos de redundancia, reduciéndose por tanto la tasa global del paquete transmitido hasta que el receptor reconozca la decodificación satisfactoria del paquete. Los códigos LDPC estructurados de tasa compatible de la técnica anterior pueden tener la propiedad no deseada de que el tamaño de bloque de información en el nivel de protografo no sea constante, lo que los hace no adecuados para aplicaciones H-ARQ. Además, en la

técnica anterior, los códigos LDPC estructurados de tasa baja pueden tener la propiedad no deseada de que los nodos variables perforados tengan grados muy altos. Por lo tanto, puede ser difícil perforarlos para conseguir tasas altas.

- 5 En varias publicaciones se describe la generación de códigos LDPC según diferentes procedimientos. El artículo científico "A Rate-Compatible Family of Protograph-Based LDPC Codes Built by Expurgation and Lengthening" de Sam Dolinar, *IEEE International Symposium on Information Theory*, septiembre de 2005, páginas 1627 a 1631, XP010845826, describe un procedimiento en el que a diferencia de los procedimientos alternativos, códigos de diferentes tasas crean una combinación de expurgación y alargamiento (equivalente a una extensión no habitual) para producir códigos de menor tasa.
- 10 El artículo científico "Rate-Compatible Low-Density Parity-Check Codes" de Tian et al., *IEEE International Symposium on Information Theory*, septiembre de 2004, página 153, XP010749864, se refiere a la generación de códigos LDPC seleccionando cuidadosamente distribuciones de grados, seguido de una combinación de anulación de información y perforación de paridad.
- 15 Un procedimiento de generación de nodos de comprobación de paridad de baja densidad se conoce por el artículo científico "Low-rate LDPC codes with simple protograph structure" de Divsalar et al., *IEEE International Symposium on Information Theory*, septiembre de 2005, páginas 1622 a 1626, XP010845825. Inspirado en códigos 'Acumular-Repetir-Acumular' y en códigos concatenados híbridos, este documento extiende la generación a tasas bajas.
- 20 Otro documento científico titulado "Design of Low-Density Parity-Check (LDPC) Codes for Deep-Space Applications" de Andrews et al., *Interplanetary Network (IPN) Progress Report, Jap Propulsion Laboratory*, 15 de noviembre de 2004, páginas 1 a 14, XP002454100, compara numerosos códigos LDPC utilizando un conjunto de métricas adecuadas para comunicaciones en el espacio profundo. Esta técnica anterior, entre otras, da a conocer códigos de protografo.

Sumario de la invención

- 25 En vista de lo anterior, las características descritas de la presente invención se refieren en general a uno o más sistemas, procedimientos y/o aparatos mejorados para comunicar voz.
- En una realización, la presente solicitud de patente comprende un procedimiento y aparato para generar protografos de tasa baja a partir de protografos de tasa alta, que comprenden copiar un grafo base, permutar puntos de extremo de aristas de un mismo tipo en copias del grafo base para producir un grafo permutado, y podar nodos de entrada sistemáticos del grafo permutado y las aristas conectadas a los mismos.
- 30 En otra realización, la presente solicitud de patente comprende un procedimiento y aparato para generar códigos de tasa alta a partir de códigos de tasa baja, que comprende perforar un subconjunto de bits de palabras de código, donde la etapa de perforar un subconjunto de bits de palabra de código comprende una perforación regular-irregular del subconjunto de bits de palabra de código, una perforación aleatoria de nodos variables o una perforación de nodos progresiva de nodos variables para obtener un código deseado a partir de un código anterior.
- 35 Un alcance adicional de la aplicabilidad del presente procedimiento y aparato resultará evidente a partir de la siguiente descripción detallada, reivindicaciones y dibujos. Sin embargo, debe entenderse que aunque la descripción detallada y los ejemplos específicos se refieren a realizaciones preferidas de la invención, sólo se proporcionan a modo de ilustración, puesto que varios cambios y modificaciones dentro del alcance de la invención resultarán evidentes a los expertos en la técnica.
- 40 La invención proporciona una solución según el contenido de las reivindicaciones independientes.

Breve descripción de los dibujos

Las características, objetos y ventajas del procedimiento y aparato descritos en este documento resultarán más evidentes a partir de la descripción detallada expuesta a continuación cuando se toma junto con los dibujos, en los que caracteres de referencia similares identifican partes correspondientes en todos ellos y en los que:

- 45 La Figura 1A es un ejemplo de un canal de borrado binario (BEC) con una probabilidad de borrado p .
- La Figura 1B es un ejemplo de un canal simétrico binario (BSC) con una probabilidad de error p .
- La Figura 2 es un ejemplo de un grafo bipartito.
- La Figura 3 es un ejemplo de una matriz de comprobación de paridad H binaria esparcida $M \times N$.
- La Figura 4 es un ejemplo de un protografo que consiste en $V = 4$ nodos variables y $C = 3$ nodos de comprobación,

conectados por $E = 8$ aristas.

La Figura 5A es un diagrama de flujo que ilustra las etapas que se ejecutan cuando se generan protografos de tasa baja a partir de protografos de tasa alta.

5 La Figura 5B es un diagrama de flujo que ilustra las etapas que se ejecutan cuando se generan protografos de tasa baja a partir de protografos de tasa alta cuando la tasa de diseño es $1/m$.

La Figura 6 ilustra un ejemplo de un grafo base B.

La Figura 7 ilustra $n-1$ copias del grafo base B, donde $n = 4$.

La Figura 8 ilustra un grafo base $B_{(n-1)}$ que se ha generado permutando los puntos de extremo de las aristas del mismo tipo en las $n-1$ copias del grafo base B.

10 La Figura 9 ilustra una versión podada del grafo permutado $B_{(n-1)}$.

La Figura 10 ilustra la optimización de un grafo permutado $B_{(n-1)}$ añadiendo una arista desde el nodo de comprobación 0 al nodo variable 8.

La Figura 11A representa un código ARCA con una tasa de $\frac{1}{2}$ según una base de protografo.

La Figura 11B muestra un protografo con el nodo variable 3 perforado, proporcionando una tasa de $\frac{2}{3}$.

15 La Figura 11C muestra un protografo con el nodo variable 4 perforado, proporcionando una tasa de $\frac{2}{3}$.

La Figura 12 es un diagrama de flujo que ilustra las etapas ejecutadas en el procedimiento de perforación regular-irregular de la presente solicitud de patente.

La Figura 13A muestra un grafo con un código ARCA no perforado con una tasa de $\frac{1}{2}$.

La Figura 13B muestra un grafo que tiene un patrón de perforación para un código de tasa 0,6.

20 La Figura 13C muestra un grafo que tiene un patrón de perforación para una tasa de $\frac{2}{3}$.

La Figura 14 es un diagrama de flujo que ilustra las etapas ejecutadas en el procedimiento de búsqueda aleatoria de la presente solicitud de patente.

Las Figuras 15A y 15B son diagramas de flujo que ilustran las etapas ejecutadas en el procedimiento de perforación progresiva de nodos de la presente solicitud de patente.

25 La Figura 16 es un diagrama de bloques funcionales de un sistema que comprende un codificador, un decodificador y un canal de comunicaciones.

La Figura 17 es un diagrama de bloques funcionales que ilustra una realización de un terminal de acceso.

La Figura 18A es un diagrama de bloques funcionales que ilustra las etapas que se ejecutan cuando se generan protografos de tasa baja a partir de protografos de tasa alta.

30 La Figura 18B es un diagrama de bloques funcionales que ilustra las etapas que se ejecutan cuando se generan protografos de tasa baja a partir de protografos de tasa alta cuando la tasa de diseño es $1/m$.

La Figura 19 es un diagrama de bloques funcionales que ilustra las etapas ejecutadas en el procedimiento de perforación regular-irregular de la presente solicitud de patente.

35 La Figura 20 es un diagrama de bloques funcionales que ilustra las etapas ejecutadas en el procedimiento de búsqueda aleatoria de la presente solicitud de patente.

Las Figuras 21A y 21B son diagramas de bloques funcionales que ilustran las etapas ejecutadas en el procedimiento de perforación progresiva de nodos de la presente solicitud de patente.

Descripción detallada

40 La expresión "a modo de ejemplo" se utiliza en este documento con el sentido de "que sirve como ejemplo, instancia o ilustración". Cualquier realización descrita en este documento como "a modo de ejemplo" no debe considerarse necesariamente como preferida o ventajosa sobre otras realizaciones.

Una estación de abonado HDR, denominada en este documento terminal de acceso (AT) 106, puede ser móvil o

estacionaria, y puede comunicarse con una o más estaciones base HDR 820, denominadas en este documento transceptores de conjunto de módems (MPT). Un terminal de acceso 106 transmite y recibe paquetes de datos a través de uno o más transceptores de conjunto de módems 820 con respecto a un controlador de estación base HDR 810, denominado en este documento controlador de conjunto de módems (MPC) 810. Los transceptores de conjunto de módems 820 y los controladores de conjunto de módems 810 son parte de una red llamada red de acceso (AN) 122. Una red de acceso 122 transporta paquetes de datos entre múltiples terminales de acceso 106. La red de acceso 122 puede conectarse además a redes adicionales externas a la red de acceso 122, tales como una intranet corporativa o la red Internet, y puede transportar paquetes de datos entre cada terminal de acceso 106 y tales redes externas 122. Un terminal de acceso 106 que ha establecido una conexión activa de canal de tráfico con uno o más transceptores de conjunto de módems 820 se denomina terminal de acceso activo 106 y se dice que está en un estado de tráfico. Se dice que un terminal de acceso 106 que está en proceso de establecer una conexión activa de canal de tráfico con uno o más transceptores de conjunto de módems 820, está en un estado de configuración de conexión. Un terminal de acceso 106 puede ser cualquier dispositivo de datos que se comunica a través de un canal inalámbrico 556 o a través de un canal cableado, por ejemplo utilizando cables coaxiales o de fibra óptica. Un terminal de acceso 106 puede ser además cualquiera de entre una pluralidad de tipos de dispositivos incluyendo, pero sin limitarse a, una tarjeta de PC, una memoria flash compacta, un módem externo o interno, o un teléfono inalámbrico o cableado. El enlace de comunicaciones a través del cual el terminal de acceso 106 envía señales al transceptor de conjunto de módems 820 se llama enlace inverso. El enlace de comunicaciones a través del cual un transceptor de conjunto de módems 820 envía señales a un terminal de acceso 106 se llama enlace directo.

Generación de protografos de tasa baja a partir de protografos de tasa alta

En una primera realización, la solicitud de patente comprende un procedimiento sistemático de baja complejidad para generar códigos LDPC estructurados de tasa baja. Estos códigos de tasa baja pueden ejecutar turbocódigos estándar y pueden tener una estructura intrínseca adecuada para una decodificación paralela de tasa alta. Estos códigos tienen un buen rendimiento en el canal de ruido blanco gaussiano medio (AWGN) y son robustos frente a borrados y perforaciones. Es decir, estos códigos LDPC estructurados de tasa baja pueden tener umbrales bajos en el canal AWGN así como un buen rendimiento en el canal de borrado. Estos códigos también son adecuados para aplicaciones de alto rendimiento. Estos códigos pueden servir como un código padre en una familia de tasa compatible para su utilización en aplicaciones de ARQ híbrido.

Empezando por un protografo 10 con una tasa relativamente alta, la presente solicitud de patente describe un procedimiento sistemático para construir protografos 10 de tasas inferiores. El protografo de tasa baja 10 se obtiene a partir de uno de mayor tasa utilizando un enfoque 'copiar y permutar' en el que un protografo más grande 10 (es decir, el protografo de tasa alta 10) se construye a partir del de tasa alta (o grafo base 12) utilizando un crecimiento de aristas progresivo (PEG) para maximizar la cintura del protografo 10 así como para mejorar su criterio de información de ciclo extrínseco. Después se poda un número calculado de nodos de entrada 20 y las aristas 15 conectadas a los mismos hasta alcanzar la tasa deseada. En función de la evolución de la densidad en los canales AWGN y de borrado, el protografo 10 puede optimizarse después adicionalmente añadiendo/permutando un número muy pequeño de aristas 15 del protografo 10. (El análisis de evolución de densidad es un procedimiento conocido en la técnica que puede llevarse a cabo en un protografo 10 para determinar si la decodificación proporcionará o no arbitrariamente una pequeña probabilidad de errores de bit en un gran grafo derivado 10. En esta técnica, los mensajes que se definen en el procedimiento de paso de mensajes se tratan como variables aleatorias, calculándose sus distribuciones).

Un protografo 10 puede ser cualquier grafo bipartito. Sin embargo, es normalmente un grafo con un número pequeño de nodos. Un protografo típico 10 consiste en un conjunto de nodos variables o de mensaje 20, un conjunto de nodos de comprobación 25 y un conjunto de aristas 15. Cada arista 15 conecta un nodo variable 20 con un nodo de comprobación 25. Además, se permiten aristas paralelas 15, de manera que el mapeo puede no ser 1:1.

Como un ejemplo sencillo, considérese el protografo 10 mostrado en la Figura 4. Este grafo consiste en $V = 4$ nodos variables 20 y $C = 3$ nodos de comprobación 25, conectados por $E = 8$ aristas 15. Los 4 nodos variables 20 del protografo se denotan " v_1, v_2, v_3, v_4 ", y los tres nodos de comprobación " c_1, c_2, c_3 ". Por sí mismo, este grafo 10 puede reconocerse como un grafo bipartito de un código LDPC ($n = 4, k = 1$), en este caso un código de repetición, donde k y n son la longitud de los bloques de información y de las palabras de código, respectivamente. El grafo bipartito de la Figura 4 se utiliza para determinar un código LDPC irregular de longitud cuatro y con un cuarto de tasa. La longitud de valor cuatro indica que hay cuatro nodos variables 20 v_1, v_2, v_3, v_4 , cada uno identificado con un bit de la palabra de código x_1, x_2, x_3, x_4 , (y sin perforación en este caso). Un cuarto de tasa indica que hay tres nodos de comprobación 25 con respecto a cuatro nodos variables 20.

Una ventaja del presente procedimiento y aparato es que los protografos resultantes 10 no tienen un alto grado de

5 nodos variables perforados V 20 con alta conectividad. Otra ventaja es que la complejidad del diseño del protografo 10 es baja. Pueden obtenerse códigos adecuados para perforarse y extenderse para generar códigos de tasa compatible con H-ARQ. La estructura del código da como resultado un paralelismo intrínseco, rápidas estructuras de decodificación y alta tasa de salida. El protografo resultante 10 está estructurado y puede expresarse en términos de acumuladores, repetidores y permutadores, lo que da como resultado estructuras de codificación rápida.

Generación de códigos válidos de tasa baja a partir de códigos base de mayor tasa

Cuando se generan estos códigos, se supone lo siguiente:

10 El protografo base B tiene N nodos variables 20 y C nodos de comprobación 25. Se perforan (ocultan) N_t de los nodos variables 20.

El número de nodos variables 20 y de nodos de comprobación 25 en el grafo base B 12 es, respectivamente, N y C.

El grafo base B 12 tiene N-C nodos de entrada sistemáticos 20 y N_t nodos redundantes perforados 20, donde $K = N - C$. Es decir, N_t es el número de nodos variables perforados 20 en el protografo B.

La tasa del protografo base (tasa alta) es $R_B = K / (N - N_t)$.

15 La Figura 5A es un diagrama de flujo que ilustra a continuación las etapas que se ejecutan cuando se generan protografos de tasa baja 10 de tasa R_t , a partir de protografos de tasa alta 10 con una tasa R_B (90):

Generar p copias del grafo base 12 B (etapa 91).

Construir otro protografo 10 B_p a partir de las p copias de B utilizando el procedimiento PEG o sus variantes para maximizar la cintura del protografo 10 (etapa 92).

20 Construir el protografo 10 L a partir de B_p podando $pK \frac{R_B - R_t}{R_B(1 - R_t)}$ nodos de entrada sistemáticos 20 y las aristas 15 conectadas a ellos 10 (etapa 93).

Optimizar la selección de los nodos de entrada podados 20 para obtener umbrales AWGN y de borrado válidos (etapa 94).

25 Puede realizarse una optimización adicional utilizando un pequeño número de operaciones de arista 15 tales como añadir, eliminar y permutar dichas aristas 15 de dicho grafo permutado podado 10 (etapa 95).

El procedimiento PEG y sus variantes pueden tener algún grado de aleatoriedad en ellos. En ese caso, la construcción PEG puede repetirse hasta alcanzar un número máximo de iteraciones (etapa 96); es decir, se repiten las etapas 92 a 95 hasta alcanzar un número máximo de iteraciones.

A continuación se describe un caso especial del procedimiento en el que se supone lo siguiente:

30 La tasa de un grafo base 12, B, es de 0,5.

La tasa de diseño es $1/m$ ($1/m < 1/2$) (se introduce m para evitar confusiones con n que es la longitud del código).

La Figura 5B es un diagrama de flujo que ilustra a continuación las etapas que se ejecutan cuando se generan protografos de tasa baja 10 a partir de protografos de tasa alta 10 (100):

35 Etapa 110 - Copiar el grafo base o padre 12, B, m-1 veces. La Figura 6 ilustra un ejemplo de un grafo base B 12. La Figura 7 ilustra m-1 copias del grafo base B 12, donde $m = 4$.

40 Etapa 120 - Construir un grafo permutado $B_{(m-1)}$ 14 permutando los puntos de extremo de aristas 15 del mismo tipo en las n-1 copias del grafo base B 12. La Figura 6 ilustra esta etapa. En una realización, la permutación de las aristas 15 se implementa utilizando el procedimiento de crecimiento de aristas progresivo (PEG) para maximizar la cintura del grafo 14. El crecimiento de aristas progresivo implica establecer aristas 15 o conexiones entre nodos variables (o de símbolos) 20 y nodos de comprobación 25 por cada arista. La idea fundamental es hallar el nodo de comprobación más distante 25 y después colocar una nueva arista 15 que conecte el nodo de símbolo 20 y el nodo de comprobación más distante 25. En la etapa 120, el procedimiento PEG se utilizó para construir el grafo permutado $B_{(m-1)}$ 14. Pueden utilizarse otras variaciones del procedimiento PEG para permutar las aristas, tales como el procedimiento ACE-PEG, el procedimiento PEG cíclico o el procedimiento ACE-PEG cíclico.

45 A continuación se explica qué es la cintura. Si dos vértices x e y del grafo están conectados, su distancia $d(x, y)$ se

define entonces como la longitud (número de aristas) de la trayectoria más corta que los une. Una trayectoria cerrada con aristas 15 que empieza en x y termina en x se denomina un ciclo de x. La cintura g se refiere al ciclo más corto de un grafo.

5 Como un ejemplo sencillo que ilustra las etapas 110 y 120, considérese el grafo base 12 mostrado en la Figura 6. Este grafo consiste en N = 5 nodos variables 20 representados por círculos y C = 3 nodos de comprobación 25 representados por cajas cuadradas. Los nodos variables 20 y los nodos de comprobación 25 están conectados por E = 14 aristas 15. Por sí mismo, este grafo puede reconocerse como el grafo de Tanner de un código LDPC (n=5, k=2), donde k = N-C.

10 Para una tasa objetivo de 1/4, m = 4. Puede obtenerse un grafo más grande mediante una operación 'copiar y permutar', ilustrada en las Figuras 7 y 8. En la Figura 7, el grafo base B 12 de la Figura 6 se ha copiado tres veces (m-1 = 3). En este caso, las tres copias están superpuestas de manera que los vértices del mismo tipo estén próximos entre sí, pero el grafo global consiste en tres subgrafos desconectados.

15 En la Figura 8, los puntos de extremo de las tres copias de cada arista 15 del grafo base 12 se han permutado entre las tres copias de los nodos variables correspondientes 20 y los nodos de comprobación 25. Después de esta permutación de los puntos de extremo de las aristas 15, los tres grafos base 12 están ahora interconectados para producir un grafo permutado B_(m-1) 14.

Etapa 125 - Si el procedimiento PEG involucra una semilla aleatoria, repetir la etapa anterior durante un número limitado de iteraciones (es decir, se ha alcanzado un número máximo de iteraciones (etapa 127)) y elegir el grafo permutado B_(m-1) 14 que tenga el menor número de ciclos más cortos (con la mayor cintura) (etapa 129).

20 Etapa 130 - Podar (N-C)(m-2) nodos de entrada sistemáticos 20 del protografo permutado B_(m-1) 14 y las aristas 15 conectadas a los mismos para obtener otro protografo, P 17. Véase la Figura 9, la cual ilustra una versión podada del grafo permutado B_(m-1) 14 en la que se han podado 4 nodos sistemáticos (o variables) 20. Los nodos podados no contienen ningún número, es decir, están en blanco. Como se ha indicado anteriormente, m = 4, N = 5 nodos variables 20 y C = 3 nodos de comprobación 25 y K = 5 - 3 = 2. Por lo tanto, (N-C)(m-2) = (5-3)(4-2) = (2)(2) = 4. En la etapa 130, el presente procedimiento y aparato poda un total de (N-C)(m-2) de los nodos de entrada sistemáticos 20. Estos nodos se eligen normalmente como cualquier (m-2) copias de los nodos sistemáticos 20 del protografo base B 12. En la mayoría de los casos, la selección de las copias que se podan no afecta al rendimiento debido a la simetría del protografo B_(m-1) 14 a lo largo de las copias. Sin embargo, si no hay tal simetría, es factible intentar todas las (N-C)(m-2) combinaciones posibles de nodos de entrada sistemáticos 20 y elegir la combinación que proporcionará el umbral más bajo. Esto se debe al pequeño tamaño del protografo 10, que realiza la búsqueda del umbral del protografo 10 utilizando un proceso rápido de evolución de densidad. Además, pueden elegirse los nodos sistemáticos de entrada podados 20 de manera aleatoria utilizando un enfoque de enfriado simulado o de gradiente descendente.

35 Etapa 140 - Puesto que el grafo permutado y podado P 17 es un protografo 10 con un código de tasa 1/m, se realiza una optimización adicional del protografo 10 añadiendo, quitando y permutando aristas 15 del grafo podado y permutado P 17. Véase la Figura 10, la cual ilustra la optimización del grafo podado y permutado P 17 añadiendo una arista 15 desde el nodo de comprobación 0 hasta el nodo variable 8.

40 En la etapa 140 se añade cuidadosamente un número muy pequeño de aristas 15 suficiente para obtener otros protografos 10 con un umbral mejor que el del P 17. Esta etapa puede realizarse manualmente. Sin embargo, la etapa 140 también puede estar automatizada, es decir, realizar una acción de arista 15, mantenerla si da como resultado un umbral inferior y descartarla en caso contrario. La acción de arista 15 significa añadir, quitar aristas 15 o permutar puntos de arista 15. En un proceso automatizado, la arista 15 de interés, así como las acciones de arista 15, se elegirán aleatoriamente.

45 Sea N', C', N_t' el número de nodos variables 20, de nodos de comprobación 25 y de nodos variables perforados 20 del protografo P 17. Entonces, puede verificarse que la tasa del protografo resultante, P 17, es 1/n;

$$\frac{N' - C'}{N' - N_t'} = \frac{(m - 1)(N - C) - (N - C)(m - 2)}{(m - 1)(N - N_t) - (N - C)(m - 2)} = \frac{1}{m} \text{ Ec. 1}$$

50 Debe observarse que puesto que el protografo P 17 se obtiene a partir de B_(m-1) 14 podando nodos 20 y aristas 15, entonces P 17 es un subgrafo de B_(m-1) 14. Esto implica que la cintura de P 17 es al menos tan grande como la cintura de B_(m-1) 14. Entonces, maximizar la cintura de B_(m-1) 14 es equivalente a una maximización limitada de la cintura de P 17. Además, si G es el grafo elevado 18 construido a partir de P 17, G 18 no tiene bucles si P 17 no tiene bucles. Además, es posible construir G 18 con una mayor cintura si la cintura de P 17 es mayor. Esto es una propiedad favorable en G 18 ya que una mayor cintura corresponde normalmente a un suelo de error bajo en la región de alta relación de señal a ruido (SNR).

El nuevo código de tasa $1/m$ se construye a partir del protografo $B_{(m-1)}$ mediante un enfoque 'copiar y permutar', en donde el número de copias se calcula según el tamaño deseado k de datos útiles del código.

Generación de códigos de tasa compatible para aplicaciones de ARQ híbrido mediante perforación

5 La presente solicitud de patente está dirigida a generar familias de códigos LDPC de tasa compatible (RC) adecuados para aplicaciones de ARQ híbrido de alto rendimiento. Los códigos de tasa compatible son una familia de códigos anidados con diferentes tasas, donde los bits de palabra de código de los códigos de tasa superior están empotrados en los códigos de tasa inferior y, por tanto, pueden codificarse utilizando un único par codificador/decodificador. Dicho de otro modo, todos los bits de una palabra de código de un código con una tasa dada aparecen en una de las palabras de código de cualquier código de tasa inferior. Son de particular interés en sistemas de datos por paquetes que permiten solicitudes de retransmisión, tales como sistemas de solicitudes de repetición automática con corrección de errores en recepción (ARQ/FEC), para conseguir una eficacia de funcionamiento deseada con un mayor grado de flexibilidad.

15 Lo que sigue es una descripción general del funcionamiento de la codificación de tasa compatible. i) Inicialmente, el transmisor 2608 envía una palabra de código de un código de tasa alta a través del canal ruidoso (es decir, el transmisor 2608 intenta minimizar el número de bits enviados a través del canal. ii) Dada la palabra de código corrupta recibida, el receptor 2610 intenta llevar a cabo la decodificación (corrección de errores y detección de errores). Si el receptor 2610 tiene problemas corrigiendo errores y recuperando el mensaje original, envía un mensaje de retroalimentación informando al transmisor 2608 de que la palabra de código no tiene la suficiente redundancia como para proteger la información contra los errores, y solicita bits adicionales para que ayuden a recuperar la información. Por lo tanto, según esta información de retroalimentación, el codificador 552 decide o bien transmitir bits de redundancia adicionales o proceder con el siguiente bloque de información. iii) Una vez que el transmisor 2608 recibe la solicitud desde el receptor 2610, construye una palabra de código de menor tasa y transmite solamente bits adicionales correspondientes a la redundancia adicional. Debe observarse que debido a la naturaleza anidada de los códigos RC, no es necesario retransmitir todos los bits del código de tasa superior que ya se han enviado, lo que hace que los códigos RC sean muy eficientes.

30 En una realización, estos códigos de tasa baja se utilizan como códigos padre de familias de tasa compatible. El presente procedimiento y aparato comprende perforar códigos de tasa baja de manera compatible con la tasa para generar códigos de tasas mayores, tal y como se muestra. Es decir, el presente procedimiento y aparato comprende un procedimiento de búsqueda novedoso e innovador de patrones de perforación de tasa compatible para hallar códigos de tasa alta a partir de códigos madre de tasa baja. Cuando se utiliza la perforación para generar códigos de tasa compatible, por ejemplo cambiar la tasa de un código una tasa superior, un codificador perfora (elimina) un subconjunto de los bits de palabra de código. En el aparato dado a conocer, el codificador 552 genera un conjunto completo de bits de paridad, pero algunos no se transmiten, es decir, no se perforan. El decodificador 537 lleva a cabo borrados donde se perforan las paridades y ejecuta el procedimiento de decodificación como en un caso no perforado.

40 En la técnica anterior, los procedimientos para obtener códigos de tasa superior a partir de códigos LDPC estructurados de tasa inferior se basaban en interconectar nodos de datos de entrada sistemáticos 20 con el grafo 10 para obtener un código de tasa superior. Otro enfoque (propuesto para conjuntos LDPC irregulares) es anular símbolos de información de un código de tasa superior para obtener un código de tasa inferior. A partir de la descripción anterior, puede observarse que estos enfoques pueden no ser de tasa compatible para aplicaciones de ARQ híbrido, ya que todos los códigos de una familia de tasa compatible con ARQ híbrido pueden tener el mismo tamaño de bloque de información a través de las diferentes tasas.

45 El umbral AWGN es la relación de señal a ruido (SNR) más baja en la que la probabilidad de error puede ser cero con una decodificación de tipo *Belief Propagation*, ya que la longitud de bloque y el número de iteraciones tienden a infinito cuando el canal es un canal AWGN.

50 Una realización del presente procedimiento de generación de una familia de códigos de tasa compatible es diseñar códigos robustos de tasa baja mediante los procedimientos descritos anteriormente. Estos códigos servirán como código padre para la familia de tasa compatible. Pueden obtenerse códigos de tasa superior en la familia perforando cuidadosamente símbolos de redundancia. Pueden obtenerse códigos de tasa inferior en la familia extendiendo el código mediante la adición de nodos redundantes.

55 La presente solicitud de patente da a conocer diferentes realizaciones de patrones de perforación. Aunque la idea de perforar un código LDPC no es totalmente nueva, perforar códigos LDPC estructurados de una manera compatible con la tasa es una ventaja de la presente solicitud de patente con respecto a la técnica anterior. Además, la perforación de códigos LDPC a partir de conjuntos irregulares dada a conocer en la técnica anterior no puede aplicarse directamente a los códigos LDPC estructurados. Esto se debe al hecho de que no pueden tener en cuenta la presencia de otros nodos perforados en el grafo 10.

Además, el patrón de perforación para conjuntos irregulares con una distribución de grado dada se determina optimizando la fracción de cada conjunto de nodos variables 20 en el grafo 10, con un determinado grado, a perforar. Por otro lado, en muchos casos de códigos LDPC estructurados, los nodos variables no perforados y no sistemáticos 20 son del mismo grado. Además, los procedimientos de perforación desarrollados en esta solicitud de patente se aplican a cualquier código LDPC y no solamente a códigos LDPC estructurados.

Perforación en el nivel de protografo

Normalmente, cuando se perfora un código LDPC se toma un código LDPC de tasa $R_p = k/n$, donde k y n son la longitud de los bloques de información y de las palabras de código, respectivamente. Para generar un código con una nueva tasa, debe perforarse un subconjunto de bits 30 de la palabra de código y enviarse los bits no perforados al receptor 2610. Se asume que el decodificador 557 conoce la posición de los bits perforados 30 en la palabra de código.

Una manera sencilla de perforar códigos LDPC estructurados es perforar los nodos variables deseados 20 en el nivel de protografo 10. Esto tiene la ventaja de que el patrón de perforación será regular en el grafo elevado 10 y será el mismo a través de los diferentes tamaños de bloque. Desafortunadamente, debido al pequeño tamaño de los protografos 10 sólo puede conseguirse un número limitado de tasas perforando en el nivel de protografo 10. Además, puede mostrarse analizando un protografo de tipo 'acumular-repetir-comprobar-acumular' (ARCA) 10, que perforar en el nivel de protografo 10 puede no dar como resultado códigos válidos.

Las Figuras 11A a C representan protografos 10 para un código ARCA sencillo de tasa $\frac{1}{2}$. En la Figura 11A, puesto que el protografo 10 solo tiene dos nodos variables no sistemáticos y no perforados (V_3 & V_4), las tasas que pueden conseguirse perforando en el nivel de protografo 10 son $\frac{2}{3}$ ó 1. Un protografo para el código base ARCA de tasa $\frac{1}{2}$ se muestra en la Figura 11A. En la Figura 11A, el nodo de comprobación 25 C_0 , representado por una caja cuadrada con un cero en su interior, está conectado con una doble arista a un nodo variable perforado V_1 20 representado mediante un círculo con un 1 en su interior. Esto es equivalente a que esté conectado a dos nodos variables perforados 20 del grafo elevado. Esto se aplica también al nodo de comprobación 25 C_1 . Sin embargo, el nodo de comprobación 25 C_2 está conectado solamente a un nodo variable perforado 20 V_1 . Observando las ecuaciones del algoritmo *Belief Propagation* (BP) (que se utiliza para decodificar códigos LDPC) y recordando que la fiabilidad de un nodo variable perforado 20 es inicialmente cero, se observa que los mensajes enviados desde un nodo de comprobación 25 que está conectado a más de un nodo variable perforado 20 pueden ser siempre cero.

Sin embargo, si un nodo de comprobación 25 está conectado solamente a un nodo variable perforado 20 V , la fiabilidad de este nodo variable perforado 20 se mejora mediante los mensajes desde los otros nodos variables 20 conectados a este mismo nodo de comprobación 25. Como resultado, inicialmente los nodos de comprobación 25 C_0 y C_1 no enviarán ninguna información hasta que la fiabilidad del nodo variable 20 V_1 mejore mediante la información enviada al mismo desde el nodo de comprobación 25 C_2 . De este modo se inicia el paso iterativo de mensajes y funciona el algoritmo BP.

Sin embargo, debe observarse que perforar cualquiera de los nodos variables 20 V_3 o V_4 , representados por un círculo con un tres o un cuatro en su interior respectivamente, para obtener una tasa de $\frac{2}{3}$ dará como resultado un nodo de comprobación 25 C_2 conectado a dos nodos variables perforados 20 V_1 y V_3 , en la Figura 11B, o a tres nodos variables V_1 y dos V_4 en la Figura 11C. La Figura 11B muestra un nodo variable 20 V_3 perforado, mientras que la Figura 11C muestra un nodo variable 20 V_4 perforado. Por tanto, todos los nodos de comprobación 25 del grafo están conectados a más de un nodo perforado y la información transmitida desde los nodos de comprobación a los nodos variables 20 es siempre cero, pudiendo no iniciarse el paso iterativo de mensajes. Por tanto, puede no ser preferible la perforación regular perforando en el nivel de protografo 10.

Procedimiento de perforación regular-irregular

En lugar de ejecutarse una perforación regular en el protografo 10, puede perforarse de manera regular los nodos variables redundantes 20 en el grafo elevado 10. Esto proporciona flexibilidad a la hora de elegir cualquier familia de tasas requeridas. El patrón de perforación será lo más regular posible con respecto a los códigos anteriores de la familia. Para las tasas más altas, se vuelve más irregular y aleatorio con respecto al código padre. Denótese $C\{R_m\}$ el código en la familia de tasa compatible con tasa R_m . El procedimiento de perforación regular-irregular 200 ejecuta las siguientes etapas ilustradas en el diagrama de flujo de la Figura 12.

Etapa 210 - Elegir un código padre $C\{R_0\}$ con la tasa base deseada R_0 , donde $C\{R_0\}$ denota el código con tasa R_0 .

Etapa 220 - Para cada tasa R_m del conjunto de tasas deseadas R_1, R_2, \dots, R_p , ($R_p > \dots > R_2 > R_1$), hallar el conjunto de nodos variables redundantes no perforados 20 V_i en un código anterior $C\{R_{m-1}\}$, donde $C\{R_{m-1}\}$ denota el código anterior con tasa de código R_{m-1} . R_{m-1} es la tasa del código anterior que va a perforarse para proporcionar un código con la tasa deseada R_m .

Etapa 230 - Sea r_{m-1} la cardinalidad 50 del conjunto de nodos variables redundantes no perforados 20 V_r en $C\{R_{m-1}\}$. (La cardinalidad r_{m-1} 50 se refiere al número de elementos del conjunto de nodos variables redundantes no perforados 20 V_r en $C\{R_{m-1}\}$, donde $C\{R_{m-1}\}$ denota el código anterior y R_{m-1} es la tasa del código anterior que va a perforarse para proporcionar un código con la tasa deseada R_m).

- 5 Etapa 240 - Calcular el número p_m , donde p_m es el número de nodos variables 20 a perforar para pasar de una tasa anterior R_{m-1} a una tasa deseada R_m mediante:

$$p_m = \text{Suelo} \left(\frac{N - (N - C)R_m}{R_m - N_{t_{m-1}}} \right) \quad \text{Ec. 2,}$$

10 donde R_m es la tasa deseada, N y C son el número de nodos variables 20 y de nodos de comprobación 25 en el grafo elevado respectivamente, $N_{t_{m-1}}$ es el número total de nodos variables perforados 20 en el grafo elevado 10 correspondiente al código $C\{R_{m-1}\}$ y $\text{Suelo}()$ representa una función de redondeo que es una función que devuelve el entero más grande inferior o igual al número entre paréntesis.

Etapa 250 - Calcular alfa 52 y beta 54 de manera que el patrón de perforación en el conjunto no perforado sea regular.

$$\text{Etapa 253 - alfa} = \text{Suelo} \left[\frac{(r_{m-1} - p_m)}{(p_m - 1)} \right] \quad \text{Ec. 3}$$

$$15 \quad \text{Etapa 256 - beta} = r_{m-1} - \text{alfa} (p_m - 1) - p_m \quad \text{Ec. 4}$$

Etapa 260 - Perforar un patrón que se obtiene en función de alfa 52 y beta 54 en el conjunto no perforado de cardinalidad 50 r_{m-1} . Lo siguiente es un ejemplo de un patrón de este tipo que va a perforarse en el conjunto no perforado:

$$[N_1 \{ \text{alfa nodos} \} N_2 \{ \text{alfa nodos} \} \dots N_m \{ \text{beta nodos} \}] \quad \text{Ec. 5,}$$

20 N_1, N_2, \dots, N_{p_m} son los índices de los nodos variables perforados 20 del conjunto de nodos redundantes no perforados 20 de cardinalidad r_{m-1} . Es decir, se perfora un nodo 20 y se dejan alfa nodos 20 sin perforar. Esto se repite hasta que se perfora el nodo variable p_m -ésimo 20. Esto deja beta nodos 20 sin perforar al final del conjunto.

25 El procedimiento de perforación regular-irregular se muestra en los grafos 13A a 13C. La Figura 13A muestra los nodos redundantes no perforados 20 en un código construido de tipo 'Acumular-Repetir-Comprobar-Acumular' (ARCA) de tasa $\frac{1}{2}$. La Figura 13B muestra el patrón de perforación para el código de tasa 0,6.

El patrón de perforación para el código de tasa $\frac{2}{3}$ mostrado en la Figura 13C es regular con respecto a los nodos no perforados 20 en el código de tasa 0,6 pero es irregular con respecto a los del código de tasa 0,5. Por este motivo hay algunos nodos de comprobación 25 con no más de un nodo variable perforado 20 conectado a los mismos cuando se inicia la decodificación iterativa.

30 Procedimiento de perforación aleatoria

35 Tal y como se ha mostrado en la realización anterior, los patrones de perforación regular e irregular no dan siempre como resultado una familia ideal de códigos. Puesto que los nodos perforados 20 se tratan como nodos borrados desde el punto de vista del algoritmo de decodificación, es preferible que el código padre tenga un buen rendimiento en el canal AWGN, así como en el canal de borrado. Una perforación aleatoria en los nodos variables redundantes no perforados 20 aparecerá como ruido aleatorio en el canal de borrado y los nodos variables perforados 20 se recuperarán si el código tiene un buen rendimiento en el canal de borrado. Sin embargo, en la técnica anterior se ha observado que seguir el análisis del canal de borrado ha demostrado que hay una tasa de corte R_c que depende de las distribuciones de grado del código y de la tasa del código padre, de manera que no puede obtenerse un código con una tasa $R > R_c$, perforando el código padre, que funcione bien según varíe la densidad.

40 Para superar las deficiencias de la técnica anterior, la presente solicitud de patente da a conocer una realización que comprende un procedimiento para construir una familia de tasa compatible buscando el mejor patrón de perforación aleatoria en el grafo elevado 10 de una manera rápida. El procedimiento se llama procedimiento de búsqueda aleatoria 300 y comprende las siguientes etapas. Estas etapas se ilustran en el diagrama de flujo de la Figura 14.

45 Etapa 310 - Empezar con un 'buen' código con la tasa base deseada R_0 , $C\{R_0\}$, es decir, elegir un código padre $C\{R_0\}$ con una tasa base deseada R_0 , donde $C\{R_0\}$ denota el código con tasa R_0 .

Definir win_snr como el umbral SNR (para el canal AWGN) del código (ganador) proporcionado por el procedimiento de búsqueda aleatoria.

5 Etapa 320 - Para cada R_m del conjunto de tasas deseadas, R_1, R_2, \dots, R_p , ($R_p > \dots > R_2 > R_1$), inicializar un valor inicial de SNR, win_snr , a infinito, donde R_1, R_2, \dots, R_p representan el conjunto de tasas en la familia de tasa compatible:

$win_snr = INF$ (infinito) Ec. 6.

Etapa 330 - Encontrar el conjunto de nodos variables redundantes no perforados $20 V_r, T$, en $C\{R_{m-1}\}$, donde $C\{R_{m-1}\}$ denota el código con tasa R_{m-1} . R_{m-1} denota que ésta es la tasa anterior que se está perforando para crear la nueva tasa deseada R_m .

10 Etapa 340 - Calcular el número de nodos variables 20 a perforar, p_m , para pasar de la tasa R_{m-1} a la R_m (como en el procedimiento irregular descrito anteriormente).

Etapa 350 - Elegir de manera aleatoria p_m nodos variables $20 V$ a partir del conjunto T de nodos variables redundantes no perforados 20 , donde p_m representa el número de nodos variables 20 a perforar.

Etapa 353 - Construir un código a partir de la tasa anterior R_{m-1} perforando los p_m nodos variables 20 .

15 Etapa 357 - Modificar la densidad para comprobar si el código generado consigue un error despreciable en una relación de señal a ruido (SNR) igual a win_snr .

Etapa 359 - ¿Consigue el código generado un error despreciable con win_snr ?

Si es así, entonces

20 Etapa 360 - Buscar de manera iterativa el nuevo umbral AWGN SNR (véase el párrafo 91) de este código reconstruido a través del intervalo SNR comprendido entre $-INF$ y el valor inicializado de SNR, win_snr . En una realización, esto se realiza bisecando de manera iterativa el intervalo SNR para seleccionar una SNR de prueba y ver si la SNR de prueba obtiene una probabilidad de error cero variando la densidad. El proceso se repite hasta que se consiga una precisión deseada en la SNR. El valor win_snr vuelve a fijarse para ser el umbral de este código que consigue un error cero. (Puede observarse que el nuevo valor de win_snr es más pequeño que el valor anterior).

25 Etapa 365 - Fijar este código reconstruido como el código ganador.

Etapa 367 - Fijar $win_snr =$ nuevo umbral

Ec. 7

Etapa 370 - Si el código no consigue un error despreciable con el valor win_snr , es decir, No en la etapa 359, entonces pasar a la etapa 372.

30 Etapa 372 - Repetir el procedimiento de búsqueda aleatoria hasta un número máximo de repeticiones volviendo a la etapa 350. El número máximo de repeticiones se determina calculando los recursos disponibles. En una implementación, es 5.

Como puede observarse a partir de la descripción anterior, el procedimiento de búsqueda aleatoria es un procedimiento voraz ya que busca el mejor código en cada tasa de diseño R_m después de hallar el mejor código en la tasa de diseño $R_{\{m-1\}}$.

35 Procedimiento de perforación progresiva de nodos 400

Otra realización utiliza una perforación progresiva de nodos 400. Con la perforación progresiva de nodos 400, para una determinada tasa de diseño, se elige de manera progresiva el patrón de perforación que:

Condición A - maximice el número de nodos de comprobación 25 C en el grafo 10 que están conectados solamente a un nodo variable perforado 20 a la vez que

40 Condición B -minimice el número medio de nodos variables perforados 20 conectados a cada nodo de comprobación 25 y

Condición C - maximice la conectividad entre los nodos de comprobación 25, conectados solamente a un nodo variable perforado 20, y los otros nodos variables perforados 20.

45 El presente procedimiento y aparato perfora el nodo variable 20 del conjunto de nodos variables no perforados 20, que está conectado al menor número de nodos de comprobación 25 que están a su vez conectados solamente a un

nodo variable perforado 20. Si hay más de un nodo variable 20 de este tipo, entonces se perfora el nodo variable 20 del conjunto que tiene el número mínimo de nodos variables perforados 20 conectados a sus nodos de comprobación adyacentes 25. Si hay más de un nodo variable de este tipo 20, se elige un nodo variable 20 de manera aleatoria. Todo el proceso puede repetirse para obtener otro código de la misma tasa deseada pero con otro patrón de perforación. Después se elige el código con el mayor resultado de perforación.

El resultado de perforación de un grafo de Tanner con nodos perforados 20 es una medida que indica hasta qué punto los nodos de comprobación 25 con un nodo variable perforado 20 están conectados a los otros nodos de comprobación 25 que están a su vez conectados a los nodos variables perforados 20.

Una ventaja de la perforación progresiva de nodos es que asigna el patrón de perforación para una determinada tasa, de manera que una perforación adicional de este código dará como resultado buenos códigos de tasa superior. Los códigos diseñados tendrán buenos umbrales AWGN y de borrado sobre toda la variedad de familias, así como buenos suelos de error. Esto es especialmente cierto para los códigos de tasa superior de la familia.

El razonamiento que subyace a la condición A es que un nodo de comprobación 25 (de grado mayor que 1) con solamente un nodo variable perforado 20 conectado al mismo transmitirá información distinta de cero al único nodo variable perforado 20 y el único nodo variable perforado 20 puede recuperarse de un error. Maximizar el número de nodos de comprobación 25 solamente con un nodo variable perforado 20 conectado puede dar como resultado que otros nodos de comprobación 25 tengan un gran número de nodos perforados 20 conectados a los mismos 25. Un nodo de comprobación 25 con más de un nodo variable perforado 20 conectado transmitirá información cero a todos los nodos variables perforados 20 a no ser que todos menos uno de estos nodos variables perforados 20 se recuperen mediante un paso de mensajes desde otros nodos de comprobación 25. Por lo tanto, es deseable minimizar el número máximo de nodos variables perforados 20 conectados a cualquier nodo de comprobación 25, lo que implica la condición B. Esto también implica que es deseable que los nodos de comprobación 25 sin nodos variables perforados 20 o con solamente un nodo variable perforado 20 conectado a los mismos 25 tengan una alta conectividad con los otros nodos de comprobación 25 con más de un nodo variable perforado 20, que es la condición C.

A continuación se describen las etapas ejecutadas cuando se utiliza el procedimiento de perforación progresiva de nodos 400. Estas etapas se ilustran en el diagrama de flujo de la Figura 15.

Etapas 410 - Para cada tasa deseada R_m del conjunto de tasas deseadas, obtener $C\{R_m\}$ a partir de $C\{R_{m-1}\}$ perforando de manera progresiva p_m nodos variables 20 del conjunto T de los nodos variables redundantes no perforados 20 de cardinalidad $50 r_{m-1}$; donde n es la longitud de la palabra de código y p_{m-1} es el número total de nodos variables perforados 20 en el grafo anterior 10. Además, $C\{R_m\}$ denota el código con tasa R_m , $C\{R_{m-1}\}$ denota el código con tasa R_{m-1} y R_{m-1} denota que ésta es la tasa anterior que está perforándose para crear la nueva tasa deseada R_m .

Etapas 420 - Inicialización

Etapas 422 - Para cada nodo de comprobación 25 (conectado a nodos variables redundantes 20), calcular $F(C)$ = número de nodos variables redundantes perforados 20 conectados al nodo de comprobación 25. ($F(C) = 1$ significa que un nodo de comprobación 25 está conectado solamente a un nodo variable perforado 20).

Etapas 424 - Hallar el conjunto R de nodos variables redundantes no perforados 20.

Etapas 426 - Para cada nodo variable redundante no perforado 20, calcular $G(V_r) =$ número de nodos de comprobación adyacentes 25 a los nodos variables redundantes no perforados 20 con $F(C) = 1$, es decir, el número de nodos de comprobación 25 conectados a los nodos variables redundantes no perforados 20, donde el nodo de comprobación 25 está conectado solamente a un nodo variable redundante perforado 20, y

Etapas 428 - Para cada nodo variable redundante no perforado 20, calcular

$$H(V_r) = \text{Sum} [F(C) \mid C \text{ es adyacente a } V_r] \quad \text{Ec. 8.}$$

Es decir, $H(V_r)$ es la suma del número de nodos variables redundantes perforados 20 conectados al nodo de comprobación 25, donde el nodo de comprobación 25 está conectado a un nodo variable redundante no perforado 20.

Etapas 430 - Perforar para hallar el nodo variable perforado 20 mientras $p < p_m$ (es decir, el número de nodos variables perforados 20 p en esta etapa $<$ el número de nodos variables perforados 20 para la tasa deseada, p_m), lo que comprende las siguientes etapas:

Etapa 432 – Hallar $T = \{V_r \text{ en } R \mid G(V_r) = \min[G(V_r)]\}$ Ec. 9,

es decir, hallar un conjunto de nodos variables no perforados 20 T en el conjunto de nodos variables no perforados 20 R para los que el número de nodos de comprobación adyacentes 25 conectados a solamente un nodo variable no perforado 20 es un valor mínimo.

5 Etapa 434 - Si $|T|=1$; $V_p = T$ Ec. 10,

es decir, si solo un nodo variable redundante no perforado 20 tiene el valor mínimo de $G(V_r)$ y el tamaño del conjunto T hallado en la etapa 4a es 1, entonces fijar que el nodo variable 20 que va a perforarse en esta etapa sea el único nodo variable 20 hallado en el conjunto T.

Etapa 435 - Si $|T| > 1$, ejecutar etapas 436-439 Ec. 11,

10 es decir, si hay más de un nodo variable 20 en el conjunto T hallado en la etapa 432, ejecutar las etapas 436 a 439.

Etapa 436 - Hallar $U = \{V_r \text{ en } T \mid H(V_r) = \min [H(V_r) \mid V_r \text{ in } T]\}$ Ec. 12,

es decir, hallar el subconjunto U de los nodos variables no perforados 20 del conjunto T, hallado en la etapa 432, en el que la suma del número de nodos variables perforados 20 conectados a nodos de comprobación 25 que están conectados a los nodos variables no perforados 20 es un valor mínimo.

15 Etapa 437 - Si $|U|=1$, entonces $V_p = U$, Ec. 13

es decir, si el tamaño del subconjunto U hallado en la etapa 436 es 1, entonces fijar que el nodo variable perforado 20 sea el único nodo variable 20 del conjunto U.

Etapa 438 - Si $|U| > 1$, elegir V_p de U de manera aleatoria Ec. 14

20 es decir, si el tamaño del conjunto U hallado en la etapa 436 es mayor que 1, entonces elegir los nodos variables 20 a perforar de entre los nodos variables no perforados 20 del subconjunto U de la etapa 436 de manera aleatoria.

Etapa 439 - Perforar el nodo variable elegido 20 y eliminar el nodo variable elegido y perforado 20 del conjunto de nodos variables no perforados 20 R.

Etapa 440 - Actualizar las funciones ejecutando las etapas 442 y 444.

25 Etapa 442 - Para cada nodo de comprobación 25 adyacente al nodo variable perforado 20, actualizar $F(C)$, donde $F(C)$ es igual al número de nodos variables redundantes perforados 20 conectados al nodo de comprobación 25.

30 Etapa 444 - Para cada nodo variable 25 adyacente al nodo de comprobación 25, actualizar $G(V_r)$ y $H(V_r)$, donde $G(V_r)$ = al número de nodos de comprobación 25 conectados a nodos variables redundantes no perforados 20, donde el nodo de comprobación 25 C esta conectado solamente a un nodo variable redundante perforado 20 y $H(V_r)$ = a la suma de nodos variables redundantes perforados 20 conectados al nodo de comprobación 25, donde el nodo de comprobación 25 está conectado a nodos variables redundantes no perforados 20.

Etapa 446 - $p=p+1$ Ec. 15

es decir, ir a la siguiente iteración. Hallar el siguiente nodo que va a perforarse. La iteración se detiene cuando $p = p_m$, es decir, cuando todos los nodos a perforar a esta tasa se han perforado.

35 Las etapas del procedimiento de perforación progresiva de nodos implementan las condiciones A y B cuando el nodo variable 20 se perfora con el menor número nodos de comprobación 25 conectados al mismo, donde los nodos de comprobación 25 están conectados a un nodo variable perforado 20. Esto se debe al hecho de que perforar este nodo variable 20 da como resultado que estos nodos de comprobación 25 hayan estado conectados en cambio a dos nodos variables perforados 20. Si hay más de un nodo variable 20 de este tipo, entonces se elige el nodo variable 20 con el menor número de nodos variables perforados 20 en el segundo nivel del árbol de soporte
40 expandido a partir de este nodo variable 20. Ésta es una manera eficaz de implementar la condición B. En la mayoría de realizaciones, puede elegirse entre más de un nodo variable candidato 20 para la perforación, es decir, $|U| = 1$. En el procedimiento de perforación progresiva de nodos descrito, un nodo variable 20 que va a perforarse se elige de manera aleatoria de entre este conjunto.

A continuación, se define el resultado de perforación de un nodo de comprobación 25, $S(C)$, como la suma de los nodos variables perforados 20 conectados a los nodos de comprobación 25 a los que se llega en una expansión de dos niveles del árbol de soporte del nodo de comprobación 25 y se calcula de la siguiente manera,

$$S(C) = \text{Sum}[F(C') \mid \text{nodo de comprobación 25 } C' \text{ adyacente al nodo variable 20 } V \text{ y nodo variable 20 } V \text{ adyacente al nodo de comprobación 25 } C].$$

El resultado de perforación del grafo 10 g , S_g , es la suma de los resultados de perforación de todos los nodos de comprobación 25 que tienen solamente uno de sus nodos variables adyacentes 20 perforado,

$$S_g = \text{Sum}[S(C) \mid C \in g \ \& \ F(C)=1] \quad \text{Ec. 17.}$$

El resultado de perforación es una aproximación y una manera eficaz de medir hasta qué punto los nodos de comprobación 25 con un nodo variable perforado 20 están conectados a los otros nodos de comprobación 25 con nodos variables perforados 20. Por tanto, el procedimiento de perforación progresiva de nodos 400 se modifica ejecutando las siguientes etapas enumeradas a continuación para implementar la condición C:

Procedimiento de perforación progresiva de nodos (bucle externo)

Etapas 450 - Maximizar la conectividad entre los nodos de comprobación con solamente un nodo variable perforado 20 y los otros nodos variables perforados 20 ejecutando las etapas 452 a 459.

$$\text{Etapas 452 - Inicializar } S_{g,0} = 0, t = 1, \quad \text{Ec. 18}$$

es decir, inicializar el resultado de perforación del grafo 10 g , S_g , a cero en la primera iteración, $t = 1$.

Mientras ($t < n^\circ$ _máximo_iteraciones) {

Etapas 454 - Cambiar la semilla aleatoria.

Etapas 456 - Ejecutar el procedimiento de perforación progresiva de nodos 400 en el código $C\{R_{m-1}\}$ para obtener un código perforado deseado $C_t\{R_m\}$, donde $C\{R_{m-1}\}$ es el código para la tasa anterior.

Etapas 458 - Calcular el resultado de perforación del grafo 10 asociado con el código perforado deseado $C_t\{R_m\}$, S_g , t .

$$\text{Etapas 459 - Si } S_{g,t} > S_{g,t-1}, \text{ fijar que } C\{R_m\} \text{ sea } C_t\{R_m\} \quad \text{Ec. 19}$$

(elegir el patrón de perforación con el mayor resultado de perforación), es decir, si el resultado de perforación del grafo 10 asociado con el código perforado deseado $C_t\{R_m\}$ es mayor que el resultado de perforación del grafo anterior 10 en la iteración $t-1$, elegir el patrón de perforación con el mayor resultado de perforación.

$$\text{Etapas 460 - } t = t + 1 \quad \text{Ec. 20,}$$

es decir, ir a la siguiente iteración.

La Figura 16 ilustra un sistema 550 que incluye un codificador 552, un decodificador 557 y un canal de comunicaciones 556. El codificador 552 incluye un circuito de codificación 553 que procesa datos de entrada A para producir una palabra de código X. La palabra de código X incluye, para fines de detección y/o corrección de errores, alguna redundancia. La palabra de código X puede transmitirse a través del canal de comunicaciones 556. Como resultado de las distorsiones producidas por el canal de comunicaciones 556, pueden perderse o corromperse partes de la palabra de código transmitida. Desde el punto de vista del decodificador, puede interpretarse que los bits perforados 30 se han perdido.

En el receptor los bits de software 30 se insertan en la palabra recibida para sustituir a los bits perdidos o perforados 30. Un borrado indica que bits 30 se han perdido en la transmisión.

El decodificador 557 intentará reconstruir la palabra de código completa X a partir de la palabra recibida Y y después lleva a cabo una operación de decodificación de datos para producir datos A a partir de la palabra de código reconstruida X.

El decodificador 557 incluye un decodificador de canal 558 para reconstruir la palabra de código completa X a partir de la palabra recibida Y. Además, incluye un decodificador de datos 559 para eliminar la información redundante incluida en la palabra de código para producir los datos de entrada originales A a partir de la palabra de código

reconstruida X.

Se apreciará que las palabras recibidas generadas con codificación LDPC pueden procesarse llevándose a cabo en las mismas operaciones de decodificación LDPC, por ejemplo operaciones de detección y corrección de errores, para generar una versión reconstruida de la palabra de código original. La palabra de código reconstruida puede someterse después a una decodificación de datos para recuperar los datos originales que se codificaron. El proceso de decodificación de datos puede ser, por ejemplo, seleccionar simplemente un subconjunto específico de los bits 30 de la palabra de código reconstruida. Las etapas ilustradas en las Figuras 5, 12, 14, 15A y 15B pueden almacenarse como instrucciones en forma de software o firmware 42 ubicadas en una memoria 45 del codificador 552. Estas instrucciones pueden ejecutarse por el circuito de codificación 553. Las etapas ilustradas en las Figuras 5, 12, 14, 15A y 15B también pueden almacenarse como instrucciones en forma de software o firmware 43 ubicadas en una memoria 46 del decodificador 557. Estas instrucciones pueden ejecutarse por el decodificador de canal 558.

La Figura 17 es un diagrama de bloques funcionales que ilustra una realización de un AT 106. El AT 106 incluye un procesador 2602 que controla el funcionamiento del AT 106. El procesador 2602 también puede denominarse CPU. Una memoria 2605, que puede incluir tanto una memoria de solo lectura (ROM) como una memoria de acceso aleatorio (RAM), proporciona instrucciones y datos al procesador 2602. Una parte de la memoria 2605 también puede incluir una memoria de acceso aleatorio no volátil (NVRAM). Las etapas ilustradas en las Figuras 5, 12, 14, 15A y 15B pueden almacenarse como instrucciones en forma de software o firmware 42 ubicadas en la memoria 2605. Estas instrucciones pueden ejecutarse por el procesador 2602.

El AT 106, que puede construirse como un dispositivo de comunicaciones inalámbricas, tal como un teléfono celular, también puede incluir una carcasa 2607 que contiene un transmisor 2608 y un receptor 2610 para permitir la transmisión y la recepción de datos, tales como comunicaciones de audio, entre el AT 2606 y una ubicación remota, tal como un AN 122. El transmisor 2608 y el receptor 2610 pueden combinarse en un transceptor 2612. Una antena 2614 está acoplada al alojamiento 2607 y acoplada eléctricamente al transceptor 2612. También pueden utilizarse antenas adicionales (no mostradas). El funcionamiento del transmisor 2608, del receptor 2610 y de la antena 2614 es ampliamente conocido en la técnica y no es necesario que se describa en este documento.

El AT 106 también incluye un detector de señales 2616 utilizado para detectar y cuantificar el nivel de las señales recibidas por el transceptor 2612. El detector de señales 2616 detecta tales señales como energía total, energía piloto por elementos de seudoruido (PN), densidad espectral de potencia y otras señales conocidas en la técnica.

Un cambiador de estados 2626 del AT 106 controla el estado del dispositivo de comunicaciones inalámbricas en función de un estado actual y de señales adicionales recibidas por el transceptor 2612 y detectadas por el detector de señales 2616. El dispositivo de comunicaciones inalámbricas puede funcionar en uno cualquiera de una pluralidad de estados.

El AT 106 también incluye un determinador de sistema 2628 utilizado para controlar el dispositivo de comunicaciones inalámbricas y determinar a qué sistema proveedor de servicios debe dirigirse el dispositivo de comunicaciones inalámbricas cuando determina que el sistema proveedor de servicios actual es inadecuado.

Los diversos componentes del AT 106 están acoplados entre sí mediante un sistema de buses 2630 que puede incluir un bus de alimentación, un bus de señales de control y un bus de señales de estado, además de un bus de datos. Sin embargo, por motivos de claridad, los diversos buses se ilustran en la Figura 7 como el sistema de buses 2630. El AT 106 también puede incluir un procesador de señales digitales (DSP) 2609 para utilizarse en el procesamiento de señales. Un experto en la técnica apreciará que el AT 106 ilustrado en la Figura 7 es un diagrama de bloques funcionales en lugar de una lista de componentes específicos.

Los procedimientos y aparatos de la Figura 5A descritos anteriormente se llevan a cabo mediante los medios correspondientes y bloques funcionales ilustrados en la Figura 18A. Dicho de otro modo, las etapas 90, 91, 92, 93, 94, 95 y 96 de la Figura 5A corresponden a los medios y a bloques funcionales 1090, 1091, 1092, 1093, 1094, 1095 y 1096 de la Figura 18A.

Los procedimientos y aparatos de la Figura 5B descritos anteriormente se llevan a cabo mediante los medios y bloques funcionales ilustrados en la Figura 18B. Dicho de otro modo, las etapas 100, 110, 120, 125, 130 y 140 de la Figura 5B corresponden a los medios y bloques funcionales 1100, 1110, 1120, 1125, 1130 y 1140 de la Figura 18B.

Los procedimientos y aparatos de la Figura 12 descritos anteriormente se llevan a cabo mediante los medios y bloques funcionales ilustrados en la Figura 19. Dicho de otro modo, las etapas 200, 210, 220, 240, 250, 253, 256 y 260 de la Figura 12 corresponden a los medios y bloques funcionales 1200, 1210, 1220, 1240, 1250, 1253, 1256 y 1260 de la Figura 19.

Los procedimientos y aparatos de la Figura 14 descritos anteriormente se llevan a cabo mediante los medios y

bloques funcionales ilustrados en la Figura 20. Dicho de otro modo, las etapas 300, 310, 320, 330, 340, 350, 353, 357, 359, 360, 365, 367 y 372 de la Figura 14 corresponden a los medios y bloques funcionales 1300, 1310, 1320, 1330, 1340, 1350, 1353, 1357, 1359, 1360, 1365, 1367 y 1372 de la Figura 20.

5 Los procedimientos y aparatos de la Figura 15A descritos anteriormente se llevan a cabo mediante los medios y bloques funcionales ilustrados en la Figura 21. Dicho de otro modo, las etapas 400, 410, 420, 422, 424, 426, 428, 430, 432, 434, 435, 436, 437, 438, 439, 440, 442, 444, 446 y 450 de la Figura 15A corresponden a los medios y bloques funcionales 1400, 1410, 1420, 1422, 1424, 1426, 1428, 1430, 1432, 1434, 1435, 1436, 1437, 1438, 1439, 1440, 1442, 1444, 1446 y 1450 de la Figura 21.

10 Los procedimientos y aparatos de la Figura 15B descritos anteriormente se llevan a cabo mediante los medios y bloques funcionales ilustrados en la Figura 22. Dicho de otro modo, las etapas 452, 454, 456, 458, 459 y 460 de la Figura 15B corresponden a los medios y bloques funcionales 1452, 1454, 1456, 1458, 1459 y 1460 de la Figura 22.

15 Los expertos en la técnica entenderán que los datos, instrucciones, comandos, información, señales, bits, símbolos y fragmentos de información a los que pueden haberse hecho referencia a lo largo de la descripción anterior se representan de manera ventajosa mediante voltajes, corrientes, ondas electromagnéticas, partículas o campos magnéticos, partículas o campos ópticos, o cualquier combinación de los mismos. Los expertos en la técnica apreciarán además que los diversos bloques lógicos, módulos, circuitos y etapas de algoritmo ilustrativos descritos con relación a las realizaciones divulgadas en este documento pueden implementarse como hardware electrónico, como software informático, o como combinaciones de ambos. Los diversos componentes, bloques, módulos, circuitos y etapas ilustrativos se han descrito de manera genérica en lo que respecta a su funcionalidad. Si tal funcionalidad se implementa en hardware o en software depende de la aplicación particular y de las limitaciones de diseño impuestas en el sistema global. Los expertos en la técnica reconocerán la intercambiabilidad del hardware y del software bajo estas circunstancias y el mejor modo de implementar la funcionalidad descrita para cada aplicación particular. Como ejemplos, los diversos bloques lógicos, módulos, circuitos y etapas de algoritmo ilustrativos descritos con relación a las realizaciones dadas a conocer en este documento pueden implementarse o llevarse a cabo con un procesador de señales digitales (DSP), con un circuito integrado de aplicación específica (ASIC), con una matriz de puertas de campo programable (FPGA) o con otro dispositivo de lógica programable, puerta discreta o lógica de transistor, componentes de hardware discretos tales como, por ejemplo, registros y FIFO, un procesador que ejecuta un conjunto de instrucciones de firmware, cualquier módulo de software programable convencional y un procesador, o con cualquier combinación de los mismos diseñada para realizar las funciones descritas en este documento. El procesador puede ser de manera ventajosa un microprocesador pero, como alternativa, el procesador puede ser cualquier máquina de estados, disposición de elementos lógicos, dispositivo de lógica programable, microcontrolador, controlador o procesador convencionales. El módulo de software puede residir en memoria RAM, memoria flash, memoria ROM, memoria EPROM, memoria EEPROM, registros, disco duro, un disco extraíble, un CD-ROM o en cualquier otra forma de medio de almacenamiento conocido en la técnica. Un procesador a modo de ejemplo está acoplado de manera ventajosa al medio de almacenamiento para leer información de, y escribir información en, el medio de almacenamiento. Como alternativa, el medio de almacenamiento puede ser una parte integrante del procesador. El procesador y el medio de almacenamiento pueden residir en un ASIC. El ASIC puede residir en un teléfono u otro terminal de usuario. Como alternativa, el procesador y el medio de almacenamiento pueden residir en un teléfono u otro terminal de usuario. El procesador puede implementarse como una combinación de un DSP y un microprocesador, o como dos microprocesadores junto con un núcleo DSP, etc.

45 Se han mostrado y descrito realizaciones preferidas de la presente invención. Sin embargo, a un experto en la técnica le resultará evidente que pueden realizarse numerosas alteraciones en las realizaciones dadas a conocer en este documento sin alejarse del alcance de la invención. Por lo tanto, la presente invención no está limitada, excepto según las siguientes reivindicaciones.

Realizaciones preferidas adicionales de la invención y ejemplos relacionados con la invención:

1.- Un procedimiento de generación de protografos de tasa baja a partir de protografos de tasa alta, que comprende:

copiar un grafo base;

50 permutar puntos de extremo de aristas de un mismo tipo en copias de dicho grafo base para producir un grafo permutado; y

podar nodos de entrada sistemáticos en dicho grafo permutado y dichas aristas conectadas a los mismos.

2.- El procedimiento según la realización 1, en el que dicha etapa de permutar puntos de extremo de aristas de un mismo tipo en dichas copias de dicho grafo base comprende además utilizar un procedimiento de crecimiento

progresivo de aristas para maximizar la cintura de dicho grafo base.

3.- El procedimiento según la realización 1, en el que dicha etapa de permutar puntos de extremo de aristas de un mismo tipo en dichas copias de dicho grafo base comprende además establecer aristas o conexiones entre nodos de símbolo y de comprobación por cada arista.

5 4.- El procedimiento según la realización 1, en el que dicha etapa de podar nodos de entrada sistemáticos comprende además: podar $(N-C)(m-2)$ nodos de entrada sistemáticos, donde N representa un número de nodos variables, C representa un número de nodos de comprobación y $1/m$ representa una tasa base deseada.

10 5.- El procedimiento según la realización 1, en el que dicha etapa de podar nodos de entrada sistemáticos comprende además: podar $pK \frac{R_B - R_L}{R_B(1 - R_L)}$ nodos de entrada sistemáticos y las aristas conectadas a los mismos, donde p es el número de copias del grafo base, K es el número de nodos variables menos el número de nodos de comprobación, R_B es la tasa base deseada y R_L es la tasa de diseño.

6.- El procedimiento según la realización 1, que comprende además optimizar dichos nodos de entrada podados utilizando una combinación de añadir, quitar y permutar dichas aristas de dicho grafo permutado podado.

15 7.- El procedimiento según la realización 2, en el que dicha etapa de permutar puntos de extremo de aristas de un mismo tipo en dichas copias de dicho grafo base comprende además repetir dicha etapa de permutación.

8.- El procedimiento según la realización 2, en el que dicho procedimiento de crecimiento progresivo de aristas es un procedimiento PEG cíclico.

9.- El procedimiento según la realización 4, en el que dicha etapa de copiar un grafo base comprende copiar dicho grafo base m-1 veces.

20 10.- El procedimiento según la realización 9, en el que dicha etapa de permutar puntos de extremo de aristas de un mismo tipo en dichas m-1 copias de dicho grafo base comprende además:

utilizar un procedimiento de crecimiento progresivo de aristas para maximizar la cintura de dicho grafo base; repetir dicha etapa de permutación; y

25 elegir el grafo permutado que tenga el menor número de ciclos más cortos si el procedimiento de crecimiento progresivo de aristas requiere una semilla aleatoria.

11.- Un procedimiento de generación de códigos de tasa alta a partir de códigos de tasa baja, que comprende:

perforar un subconjunto de bits de palabra de código.

12.- El procedimiento de generación de códigos de tasa alta a partir de códigos de tasa baja según la realización 11, en el que dichos códigos de tasa baja son códigos padre.

30 13.- El procedimiento de generación de códigos de tasa alta a partir de códigos de tasa baja según la realización 11, en el que dicha etapa de perforación de un subconjunto de bits de palabra de código comprende una perforación regular-irregular de dicho subconjunto de bits de palabra de código, que comprende:

elegir un código padre con una tasa base deseada;

35 hallar un conjunto de nodos variables no perforados en un código anterior para cada tasa en un conjunto de tasas deseadas;

calcular un número de nodos variables a perforar para pasar de una tasa anterior a una tasa deseada;

calcular alfa y beta de manera que un patrón de perforación en un conjunto no perforado sea regular; y

perforar un patrón que se obtiene en función de dicho alfa y dicho beta en dicho conjunto no perforado.

40 14.- El procedimiento de generación de códigos de tasa alta a partir de códigos de tasa baja según la realización 11, en el que dicha etapa de perforar un subconjunto de bits de palabra de código comprende perforar de manera aleatoria nodos variables.

15.- El procedimiento de generación de códigos de tasa alta a partir de códigos de tasa baja según la realización 11, en el que dicha etapa de perforar un subconjunto de bits de palabra de código comprende:

perforar de manera progresiva nodos variables para obtener un código deseado a partir de un código anterior.

16.- El procedimiento de generación de códigos de tasa alta a partir de códigos de tasa baja según la realización 12, en el que dichos códigos padre se generan:

- 5 copiando un grafo base;
- permutando puntos de extremo de aristas de un mismo tipo en copias de dicho grafo base para producir un grafo permutado; y
- podar nodos de entrada sistemáticos en dicho grafo permutado y dichas aristas conectadas a los mismos.

10 17.- El procedimiento de generación de códigos de tasa alta a partir de códigos de tasa baja según el ejemplo 13, en el que dicha etapa de calcular una pluralidad de nodos variables a perforar para pasar de una tasa anterior a una tasa deseada utiliza la siguiente fórmula:

$$p_m = \text{Suelo} (N - (N-C)/R_m - N_{t_{m-1}}),$$

15 donde p_m es el número de nodos variables a perforar, R_m es la tasa deseada, R_{m-1} es una tasa anterior, N y C son un número de nodos variables y nodos de comprobación en un grafo elevado respectivamente, $N_{t_{m-1}}$ es el número total de dichos nodos variables perforados en el grafo elevado correspondiente a un código anterior $C\{R_{m-1}\}$, y $\text{Suelo}()$ representa una función de redondeo.

18.- El procedimiento de generación de códigos de tasa alta a partir de códigos de tasa baja según el ejemplo 13, en el que dicha etapa de calcular alfa y beta de manera que un patrón de perforación en el conjunto no perforado sea regular, utiliza las siguientes fórmulas:

20
$$\text{alfa} = \text{Suelo} [(r_{m-1} - p_m) / (p_m - 1)]$$

y

$$\text{beta} = r_{m-1} - \text{alfa} (p_m - 1) - p_m,$$

25 donde p_m es el número de nodos variables a perforar, r_{m-1} es la cardinalidad del conjunto de nodos variables redundantes no perforados en un código anterior $C\{R_{m-1}\}$, R_{m-1} es una tasa anterior y $\text{Suelo}()$ representa una función de redondeo.

30 19.- El procedimiento de generación de códigos de tasa alta a partir de códigos de tasa baja según el ejemplo 13, en el que dicha etapa de perforar un patrón que se obtiene en función de dicho alfa y dicho beta en dicho conjunto no perforado comprende utilizar un patrón [N_1 {alfa nodos} N_2 {alfa nodos} ... N_m {beta nodos}], donde N_1, N_2, \dots, N_p son los índices de los nodos variables perforados del conjunto de nodos redundantes no perforados de cardinalidad r_{m-1} .

20.- El procedimiento de generación de códigos de tasa alta a partir de códigos de tasa baja según el ejemplo 14, que comprende además buscar de manera aleatoria dicho subconjunto de bits de palabra de código, lo que comprende:

- 35 i) elegir un código padre con una tasa base deseada;
- ii) inicializar una relación de señal a ruido para cada tasa deseada en el conjunto de tasas deseadas;
- iii) hallar un conjunto de nodos variables no perforados en un código anterior con una tasa anterior;
- iv) calcular un número de nodos variables a perforar para pasar de dicha tasa anterior a una tasa deseada;
- v) elegir de manera aleatoria dicho número de nodos variables a perforar a partir de dicho conjunto de nodos variables no perforados;
- 40 vi) construir un código a partir de la tasa anterior perforando dicho número elegido de nodos variables;
- vii) modificar la densidad para probar si dicho código construido consigue un error despreciable con la relación de señal a ruido inicializada;
- viii) buscar de manera iterativa un nuevo umbral de señal a ruido de dicho código construido, estableciendo dicho código construido como un código ganador y estableciendo dicha relación de señal a ruido igual a un nuevo umbral si dicho código construido consigue un error despreciable con dicha relación de señal a ruido;
- 45

y

ix) repetir dicha búsqueda aleatoria repitiendo las etapas v a vii si dicho código construido no consigue un error despreciable con dicha relación de señal a ruido.

5 21.- El procedimiento de generación de códigos de tasa alta a partir de códigos de tasa baja según la realización 15, en el que dicha etapa de perforar de manera progresiva nodos variables para obtener un código deseado a partir de un código anterior comprende:

10 perforar de manera progresiva dichos nodos variables de un conjunto de dichos nodos variables no perforados de cardinalidad r_{m-1} , donde r_{m-1} es el número de elementos del conjunto de nodos variables redundantes no perforados en un código anterior y donde los nodos variables redundantes no perforados del código anterior pueden perforarse para proporcionar el código de una tasa deseada.

22.- El procedimiento de generación de códigos de tasa alta a partir de códigos de tasa baja según el ejemplo 20, en el que dicha etapa de buscar de manera iterativa un nuevo umbral de dicho código construido comprende:

- 15 a) bisecar de manera iterativa dicho intervalo para seleccionar una relación de señal a ruido de prueba;
- b) determinar si dicha relación de señal a ruido de prueba consigue un error cero; y
- c) repetir las etapas a y b hasta que se obtenga una precisión deseada en dicha relación de señal a ruido.

23.- El procedimiento de generación de códigos de tasa alta a partir de códigos de tasa baja según la realización 21, en el que dicha etapa de perforar de manera progresiva nodos variables de un conjunto de dichos nodos variables no perforados comprende una etapa de inicialización que comprende:

20 calcular para cada nodo de comprobación conectado a al menos un dicho nodo variable una pluralidad de dichos nodos variables perforados conectados a dichos nodos de comprobación, donde cada uno de dichos nodos de comprobación está conectado solamente a un dicho nodo variable perforado;

hallar dicho conjunto de dichos nodos variables no perforados;

25 calcular para cada dicho nodo variable no perforado una pluralidad de dichos nodos de comprobación conectados a cada dicho nodo variable no perforado, en el que cada uno de dichos nodos de comprobación está conectado solamente a un dicho nodo variable perforado; y

calcular para cada dicho nodo variable no perforado una suma de dichos nodos variables perforados conectados a dichos nodos de comprobación, donde dichos nodos de comprobación están conectados a dichos nodos variables no perforados.

30 24.- El procedimiento de generación de códigos de tasa alta a partir de códigos de tasa baja según la realización 23, en el que dicha etapa de perforar de manera progresiva nodos variables de un conjunto de dichos nodos variables no perforados comprende además una etapa de perforación para hallar dicho nodo variable perforado mientras el número de dichos nodos variables perforados en esta etapa es inferior al número de dichos nodos variables perforados para dicha tasa deseada.

35 25.- El procedimiento de generación de códigos de tasa alta a partir de códigos de tasa baja según la realización 24, en el que dicha etapa de hallar dicho nodo variable perforado cuando un número de dichos nodos variables perforados en esta etapa es inferior a un número de dichos nodos variables perforados para dicha tasa deseada comprende:

40 a) hallar un segundo conjunto de dichos nodos variables no perforados en dicho conjunto de dichos nodos variables no perforados para los que un número de nodos adyacentes de dichos nodos de comprobación es un valor mínimo;

b) fijar el nodo variable perforado como el nodo del segundo conjunto si dicho número de dichos nodos variables no perforados en dicho segundo conjunto de la etapa a es 1;

c) ejecutar las etapas ci a civ si el número de dichos nodos variables no perforados de dicho segundo conjunto de la etapas a es > 1 ;

45 ci) hallar un subconjunto de dicho segundo conjunto de dichos nodos variables no perforados de la etapa a, donde la suma de dicho número de nodos variables perforados conectados a dichos nodos de comprobación que están conectados a dichos nodos variables no perforados es un valor mínimo;

cii) fijar dichos nodos variables perforados al único nodo del subconjunto de la etapa ci si dicho número de dichos nodos variables no perforados de dicho subconjunto de la etapa ci es 1;

ciii) elegir dichos nodos variables a partir de dichos nodos variables no perforados del subconjunto de la etapa ci de manera aleatoria si el número de nodos variables no perforados hallados en el subconjunto de la etapa ci es mayor que 1; y

civ) perforar dicho nodo variable elegido y eliminar dicho nodo variable elegido y perforado del conjunto de nodos variables no perforados.

26.- El procedimiento de generación de códigos de tasa alta a partir de códigos de tasa baja según la realización 25, en el que dicha etapa de perforar de manera progresiva nodos variables de un conjunto de dichos nodos variables no perforados comprende además una etapa de actualización, que comprende:

actualizar dicho número de nodos variables perforados conectados a los nodos de comprobación para cada uno de dichos nodos de comprobación adyacentes a dichos nodos variables perforados;

actualizar el número de dichos nodos de comprobación conectados a dichos nodos variables no perforados, donde dicho nodo de comprobación está conectado solamente a un dicho nodo variable perforado para cada dicho nodo variable adyacente a dicho nodo de comprobación; y

actualizar la suma de dichos nodos variables perforados conectados a dicho nodo de comprobación, donde dicho nodo de comprobación está conectado a dichos nodos variables no perforados para cada dicho nodo variable adyacente a dicho nodo de comprobación.

27.- El procedimiento de generación de códigos de tasa alta a partir de códigos de tasa baja según la realización 26, en el que dicha etapa de perforar de manera progresiva nodos variables de un conjunto de dichos nodos variables no perforados comprende además la etapa de maximizar la conectividad entre los nodos de comprobación con solamente un nodo variable perforado y los otros nodos variables perforados.

28.- El procedimiento de generación de códigos de tasa alta a partir de códigos de tasa baja según la realización 27, en el que dicha etapa de maximizar la conectividad entre los nodos de comprobación con solamente un nodo variable perforado y los otros nodos variables perforados, comprende:

inicializar un resultado de perforación a cero;

modificar una semilla aleatoria;

ejecutar dichas etapas de las realizaciones 15, 21, 23, 24 y 25 en dicho código anterior para obtener dicho código perforado deseado;

calcular el resultado de perforación del código perforado deseado; y

elegir el patrón de perforación con el mayor resultado de perforación si el resultado de perforación del código perforado deseado es mayor que el resultado de perforación del código anterior.

29.- Un medio de generación de protografos de tasa baja a partir de protografos de tasa alta, que comprende:

un medio para copiar un grafo base;

un medio para permutar puntos de extremo de aristas de un mismo tipo en copias de dicho grafo base para producir un grafo permutado; y

un medio para podar nodos de entrada sistemáticos en dicho grafo permutado y dichas aristas conectadas a los mismos.

30.- El medio según la realización 29, en el que dicho medio para permutar puntos de extremo de aristas de un mismo tipo en dichas copias de dicho grafo base comprende además un medio para utilizar un procedimiento de crecimiento progresivo de aristas para maximizar la cintura de dicho grafo base.

31.- El medio según la realización 29, en el que dicho medio para permutar puntos de extremo de aristas de un mismo tipo en dichas copias de dicho grafo base comprende además un medio para establecer aristas o conexiones entre nodos de símbolo y de comprobación por cada arista.

32.- El medio según la realización 29, en el que dicho medio para podar nodos de entrada sistemáticos comprende además un medio para podar $(N-C)(m-2)$ nodos de entrada sistemáticos, donde N representa un número de nodos variables, C representa un número de nodos de comprobación y $1/m$ representa una tasa base deseada.

- 33.- El medio según la realización 29, en el que dicha etapa de podar nodos de entrada sistemáticos comprende además: podar $pK \frac{R_B - R_L}{R_B(1 - R_L)}$ nodos de entrada sistemáticos y las aristas conectadas a los mismos, donde p es el número de copias del grafo base, K es el número de nodos variables menos el número de nodos de comprobación, R_B es la tasa base deseada y R_L es la tasa de diseño.
- 5 34.- El medio según la realización 29, que comprende además optimizar dichos nodos de entrada podados utilizando una combinación de añadir, quitar y permutar dichas aristas de dicho grafo permutado podado.
- 35.- El medio según la realización 30, en el que dicho medio para permutar puntos de extremo de aristas de un mismo tipo en dichas copias de dicho grafo base comprende además un medio para repetir dicha permutación.
- 10 36.- El medio según la realización 30, en el que dicho procedimiento de crecimiento progresivo de aristas es un procedimiento PEG cíclico.
- 37.- El medio según la realización 32, en el que dicho medio para copiar un grafo base comprende un medio para copiar dicho grafo base m-1 veces.
- 38.- El medio según la realización 37, en el que dicho medio para permutar puntos de extremo de aristas de un mismo tipo en dichas m-1 copias de dicho grafo base comprende además:
- 15 un medio para utilizar un procedimiento de crecimiento progresivo de aristas para maximizar la cintura de dicho grafo base;
- un medio para repetir dicha permutación; y
- un medio para elegir el grafo permutado que tenga el menor número de ciclos más cortos si el procedimiento de crecimiento progresivo de aristas requiere una semilla aleatoria.
- 20 39.- Un medio de generación de códigos de tasa alta a partir de códigos de tasa baja, que comprende:
- un medio para perforar un subconjunto de bits de palabra de código.
- 40.- El medio de generación de códigos de tasa alta a partir de códigos de tasa baja según la realización 39, en el que dichos códigos de tasa baja son códigos padre.
- 25 41.- El medio de generación de códigos de tasa alta a partir de códigos de tasa baja según la realización 39, en el que dicho medio para perforar un subconjunto de bits de palabra de código comprende un medio para una perforación regular-irregular de dicho subconjunto de bits de palabra de código, que comprende:
- un medio para elegir un código padre con una tasa base deseada;
- un medio para hallar un conjunto de nodos variables no perforados en un código anterior para cada tasa en un conjunto de tasas deseadas;
- 30 un medio para calcular un número de nodos variables a perforar para pasar de una tasa anterior a una tasa deseada;
- un medio para calcular alfa y beta de manera que un patrón de perforación en un conjunto no perforado sea regular; y
- 35 un medio para perforar un patrón que se obtiene en función de dicho alfa y dicho beta en dicho conjunto no perforado.
- 42.- El medio de generación de códigos de tasa alta a partir de códigos de tasa baja según la realización 39, en el que dicho medio para perforar un subconjunto de bits de palabra de código comprende un medio para perforar de manera aleatoria nodos variables.
- 40 43.- El medio de generación de códigos de tasa alta a partir de códigos de tasa baja según la realización 39, en el que dicho medio para perforar un subconjunto de bits de palabra de código comprende:
- un medio para perforar de manera progresiva nodos variables para obtener un código deseado a partir de un código anterior.
- 44.- El medio de generación de códigos de tasa alta a partir de códigos de tasa baja según la realización 40, en el que dichos códigos padre se generan mediante:

un medio para copiar un grafo base;

un medio para permutar puntos de extremo de aristas de un mismo tipo en copias de dicho grafo base para producir un grafo permutado; y

5 un medio para podar nodos de entrada sistemáticos en dicho grafo permutado y dichas aristas conectadas a los mismos.

45.- El medio de generación de códigos de tasa alta a partir de códigos de tasa baja según el ejemplo 41, en el que dicho medio para calcular una pluralidad de nodos variables a perforar para pasar de una tasa anterior a una tasa deseada utiliza la siguiente fórmula:

$$p_m = \text{Suelo} \left(\frac{N - (N - C)/R_m - N_{t_{m-1}}}{R_m} \right),$$

10 donde p_m es el número de nodos variables a perforar, R_m es la tasa deseada, R_{m-1} es una tasa anterior, N y C son un número de nodos variables y nodos de comprobación en un grafo elevado respectivamente, $N_{t_{m-1}}$ es el número total de dichos nodos variables perforados en el grafo elevado correspondiente a un código anterior $C\{R_{m-1}\}$, y $\text{Suelo}()$ representa una función de redondeo.

15 46.- El medio de generación de códigos de tasa alta a partir de códigos de tasa baja según el ejemplo 41, en el que dicho medio para calcular alfa y beta de manera que un patrón de perforación en el conjunto no perforado sea regular, utiliza las siguientes fórmulas:

$$\text{alfa} = \text{Suelo} \left[\frac{(r_{m-1} - p_m)}{(p_m - 1)} \right]$$

y

$$\text{beta} = r_{m-1} - \text{alfa} (p_m - 1) - p_m,$$

20 donde p_m es el número de nodos variables a perforar, r_{m-1} es la cardinalidad del conjunto de nodos variables redundantes no perforados en un código anterior $C\{R_{m-1}\}$, R_{m-1} es una tasa anterior y $\text{Suelo}()$ representa una función de redondeo.

25 47.- El medio de generación de códigos de tasa alta a partir de códigos de tasa baja según el ejemplo 41, en el que dicho medio para perforar un patrón que se obtiene en función de dicho alfa y dicho beta en dicho conjunto no perforado comprende un medio para utilizar un patrón $[N_1 \{\text{alfa nodos}\} N_2 \{\text{alfa nodos}\} \dots N_m \{\text{beta nodos}\}]$, donde N_1, N_2, \dots, N_{p-m} son los índices de los nodos variables perforados del conjunto de nodos redundantes no perforados de cardinalidad r_{m-1} .

30 48.- El medio de generación de códigos de tasa alta a partir de códigos de tasa baja según el ejemplo 42, que comprende además un medio para buscar de manera aleatoria dicho subconjunto de bits de palabra de código, que comprende:

i) un medio para elegir un código padre con una tasa base deseada;

ii) un medio para inicializar una relación de señal a ruido para cada tasa deseada en el conjunto de tasas deseadas;

35 iii) un medio para hallar un conjunto de nodos variables no perforados en un código anterior con una tasa anterior;

iv) un medio para calcular un número de nodos variables a perforar para pasar de dicha tasa anterior a una tasa deseada;

v) un medio para elegir de manera aleatoria dicho número de nodos variables a perforar a partir de dicho conjunto de nodos variables no perforados;

40 vi) un medio para construir un código a partir de la tasa anterior perforando dicho número elegido de nodos variables;

vii) un medio para modificar la densidad para probar si dicho código construido consigue un error despreciable con la relación de señal a ruido inicializada;

45 viii) un medio para buscar de manera iterativa un nuevo umbral de señal a ruido de dicho código construido, estableciendo dicho código construido como un código ganador y estableciendo dicha relación de señal a ruido igual a un nuevo umbral si dicho código construido consigue un error despreciable con dicha relación

de señal a ruido; y

ix) un medio para repetir dicha búsqueda aleatoria repitiendo las etapas v a vii si dicho código construido no consigue un error despreciable con dicha relación de señal a ruido.

5 49.- El medio de generación de códigos de tasa alta a partir de códigos de tasa baja según la realización 43, en el que dicho medio para perforar de manera progresiva nodos variables para obtener un código deseado a partir de un código anterior comprende:

10 un medio para perforar de manera progresiva dichos nodos variables de un conjunto de dichos nodos variables no perforados de cardinalidad r_{m-1} , donde r_{m-1} es el número de elementos del conjunto de nodos variables redundantes no perforados en un código anterior y donde los nodos variables redundantes no perforados del código anterior pueden perforarse para proporcionar el código de una tasa deseada.

50.- El medio de generación de códigos de tasa alta a partir de códigos de tasa baja según la realización 48, en el que dicho medio para buscar de manera iterativa un nuevo umbral de dicho código construido comprende:

- 15 a) un medio para biseccionar de manera iterativa dicho intervalo para seleccionar una relación de señal a ruido de prueba;
- b) un medio para determinar si dicha relación de señal a ruido de prueba consigue un error cero; y
- c) un medio para repetir los procesos de los medios a y b hasta que se obtenga una precisión deseada en dicha relación de señal a ruido.

20 51.- El medio de generación de códigos de tasa alta a partir de códigos de tasa baja según la realización 49, en el que dicho medio para perforar de manera progresiva nodos variables de un conjunto de dichos nodos variables no perforados comprende un medio de inicialización que comprende:

- un medio para calcular para cada nodo de comprobación conectado a al menos un dicho nodo variable una pluralidad de dichos nodos variables perforados conectados a dichos nodos de comprobación, donde cada uno de dichos nodos de comprobación está conectado solamente a un dicho nodo variable perforado;
- un medio para hallar dicho conjunto de dichos nodos variables no perforados;
- 25 un medio para calcular para cada dicho nodo variable no perforado una pluralidad de dichos nodos de comprobación conectados a cada dicho nodo variable no perforado, donde cada uno de dichos nodos de comprobación está conectado solamente a un dicho nodo variable perforado; y
- 30 un medio para calcular para cada dicho nodo variable no perforado una suma de dichos nodos variables perforados conectados a dichos nodos de comprobación, donde dichos nodos de comprobación están conectados a dichos nodos variables no perforados.

35 52.- El medio de generación de códigos de tasa alta a partir de códigos de tasa baja según la realización 51, en el que dicho medio para perforar de manera progresiva nodos variables de un conjunto de dichos nodos variables no perforados comprende además un medio de perforación para hallar dicho nodo variable perforado mientras el número de dichos nodos variables perforados en esta etapa es inferior al número de dichos nodos variables perforados para dicha tasa deseada.

53.- El medio de generación de códigos de tasa alta a partir de códigos de tasa baja según la realización 52, en el que dicho medio para hallar dicho nodo variable perforado mientras el número de dichos nodos variables perforados en esta etapa es inferior al número de dichos nodos variables perforados para dicha tasa deseada comprende:

- 40 a) un medio para hallar un segundo conjunto de dicho número de nodos variables no perforados en dicho conjunto de dichos nodos variables no perforados para los que un número de nodos adyacentes de dichos nodos de comprobación es un valor mínimo;
- b) un medio para fijar el nodo variable perforado como el nodo del segundo conjunto si dicho número de dichos nodos variables no perforados en dicho segundo conjunto del medio a es 1;
- 45 c) un medio para ejecutar los procesos de los medios ci a civ si el número de dichos nodos variables no perforados de dicho segundo conjunto del medio a es > 1 ;
- ci) un medio para hallar un subconjunto de dicho segundo conjunto de dichos nodos variables no perforados del medio a, donde la suma de dicho número de nodos variables perforados conectados a dichos nodos de comprobación que están conectados a dichos nodos variables no perforados es un

valor mínimo;

cii) un medio para fijar dichos nodos variables perforados al único nodo del subconjunto del medio ci si el número de dichos nodos variables no perforados de dicho subconjunto del medio ci es 1;

5 ciii) un medio para elegir dichos nodos variables a partir de dichos nodos variables no perforados del medio ci de manera aleatoria si el número de los nodos variables no perforados del subconjunto del medio ci es mayor que 1; y

civ) un medio para perforar dicho nodo variable elegido y un medio para eliminar dicho nodo variable elegido y perforado del conjunto de nodos variables no perforados.

10 54.- El medio de generación de códigos de tasa alta a partir de códigos de tasa baja según la realización 53, en el que dicho medio para perforar de manera progresiva nodos variables de un conjunto de dichos nodos variables no perforados comprende además un medio de actualización, que comprende:

un medio para actualizar dicho número de nodos variables perforados conectados a los nodos de comprobación para cada uno de dichos nodos de comprobación adyacentes a dichos nodos variables perforados;

15 un medio para actualizar el número de dichos nodos de comprobación conectados a dichos nodos variables no perforados, donde dicho nodo de comprobación está conectado solamente a un dicho nodo variable perforado para cada dicho nodo variable adyacente a dicho nodo de comprobación; y

20 un medio para actualizar la suma de dichos nodos variables perforados conectados a dicho nodo de comprobación, donde dicho nodo de comprobación está conectado a dichos nodos variables no perforados para cada dicho nodo variable adyacente a dicho nodo de comprobación.

55.- El medio de generación de códigos de tasa alta a partir de códigos de tasa baja según la realización 54, en el que dicho medio para perforar de manera progresiva nodos variables de un conjunto de dichos nodos variables no perforados comprende además un medio para maximizar la conectividad entre los nodos de comprobación con solamente un nodo variable perforado y los otros nodos variables perforados.

25 56.- El medio de generación de códigos de tasa alta a partir de códigos de tasa baja según la realización 55, en el que dicho medio para maximizar la conectividad entre los nodos de comprobación con solamente un nodo variable perforado y los otros nodos variables perforados comprende:

un medio para inicializar un resultado de perforación a cero;

un medio para modificar una semilla aleatoria;

30 un medio para ejecutar dichos medios de las realizaciones 43, 49, 51, 52 y 53 en dicho código anterior para obtener dicho código perforado deseado;

un medio para calcular el resultado de perforación del código perforado deseado; y

un medio para elegir el patrón de perforación con el mayor resultado de perforación si el resultado de perforación del código perforado deseado es mayor que el resultado de perforación del código anterior.

35 57.- Un codificador que genera protografos de tasa baja a partir de protografos de tasa alta, que comprende:

una memoria, en la que dicho codificador está adaptado para ejecutar instrucciones de software almacenadas en dicha memoria, que comprenden:

copiar un grafo base;

40 permutar puntos de extremo de aristas de un mismo tipo en copias de dicho grafo base para producir un grafo permutado; y

podar nodos de entrada sistemáticos en dicho grafo permutado y dichas aristas conectadas a los mismos.

45 58.- El codificador según la realización 57, en el que dicha instrucción para permutar puntos de extremo de aristas de un mismo tipo en dichas copias de dicho grafo base comprende además utilizar un procedimiento de crecimiento progresivo de aristas para maximizar la cintura de dicho grafo base.

59.- El codificador según la realización 57, en el que dicha instrucción para permutar puntos de extremo de aristas

de un mismo tipo en dichas copias de dicho grafo base comprende además establecer aristas o conexiones entre nodos de símbolo y de comprobación por cada arista.

- 5 60.- El codificador según la realización 57, en el que dicha instrucción para podar nodos de entrada sistemáticos comprende además podar $(N-C)(m-2)$ nodos de entrada sistemáticos, donde N representa un número de nodos variables, C representa un número de nodos de comprobación y $1/m$ representa una tasa deseada.
- 61.- El codificador según la realización 57, en el que dicha instrucción para podar nodos de entrada sistemáticos comprende además: podar $pK \frac{R_B - R_L}{R_B(1 - R_L)}$ nodos de entrada sistemáticos y las aristas conectadas a los mismos, donde p es el número de copias del grafo base, K es el número de nodos variables menos el número de nodos de comprobación, R_B es la tasa base deseada y R_L es la tasa de diseño.
- 10 62.- El codificador según la realización 57, que comprende además una instrucción para optimizar dichos nodos de entrada podados utilizando una combinación de añadir, quitar y permutar dichas aristas de dicho grafo permutado podado.
- 15 63.- El codificador según la realización 58, en el que dicha instrucción para permutar puntos de extremo de aristas de un mismo tipo en dichas copias de dicho grafo base comprende además repetir dicha instrucción para permutar puntos de extremo de aristas de un mismo tipo en las copias de dicho grafo base para producir un grafo permutado.
- 64.- El codificador según la realización 58, en el que dicho procedimiento de crecimiento progresivo de aristas es un procedimiento PEG cíclico.
- 65.- El codificador según la realización 60, en el que dicha instrucción para copiar un grafo base comprende copiar dicho grafo base m-1 veces.
- 20 66.- El codificador según la realización 65, en el que dicha instrucción para permutar puntos de extremo de aristas de un mismo tipo en dichas m-1 copias de dicho grafo base comprende además:
- utilizar un procedimiento de crecimiento progresivo de aristas para maximizar la cintura de dicho grafo base;
 - repetir dicha instrucción para permutar puntos de extremo de aristas de un mismo tipo en las copias de dicho grafo base para producir un grafo permutado; y
 - 25 elegir el grafo permutado que tenga el menor número de ciclos más cortos si el procedimiento de crecimiento progresivo de aristas requiere una semilla aleatoria.
- 67.- Un codificador que genera códigos de tasa alta a partir de códigos de tasa baja, que comprende:
- una memoria, en la que dicho codificador está adaptado para ejecutar instrucciones almacenadas en dicha memoria, que comprenden:
 - 30 perforar un subconjunto de bits de palabra de código.
- 68.- El codificador que genera códigos de tasa alta a partir de códigos de tasa baja según la realización 67, en el que dichos códigos de tasa baja son códigos padre.
- 69.- El codificador que genera códigos de tasa alta a partir de códigos de tasa baja según la realización 67, en el que dicha instrucción para perforar un subconjunto de bits de palabra de código comprende una perforación regular-irregular de dicho subconjunto de bits de palabra de código, que comprende:
- elegir un código padre con una tasa base deseada;
 - hallar un conjunto de nodos variables no perforados en un código anterior para cada tasa en un conjunto de tasas deseadas;
 - calcular un número de nodos variables a perforar para pasar de una tasa anterior a una tasa deseada;
 - 40 calcular alfa y beta de manera que un patrón de perforación en un conjunto no perforado sea regular; y
 - perforar un patrón que se obtiene en función de dicho alfa y dicho beta en dicho conjunto no perforado.
- 70.- El codificador que genera códigos de tasa alta a partir de códigos de tasa baja según la realización 67, en el que dicha instrucción para perforar un subconjunto de bits de palabra de código comprende perforar de manera aleatoria nodos variables.

71.- El codificador que genera códigos de tasa alta a partir de códigos de tasa baja según la realización 67, en el que dicha instrucción para perforar un subconjunto de bits de palabra de código comprende:

perforar de manera progresiva nodos variables para obtener un código deseado a partir de un código anterior.

5 72.- El codificador que genera códigos de tasa alta a partir de códigos de tasa baja según la realización 68, en el que dichos códigos padre se generan ejecutando las siguientes instrucciones:

copiar un grafo base;

permutar puntos de extremo de aristas de un mismo tipo en copias de dicho grafo base para producir un grafo permutado; y

10 podar nodos de entrada sistemáticos en dicho grafo permutado y dichas aristas conectadas a los mismos.

73.- El codificador que genera códigos de tasa alta a partir de códigos de tasa baja según el ejemplo 69, en el que dicha instrucción para calcular una pluralidad de nodos variables a perforar para pasar de una tasa anterior a una tasa deseada utiliza la siguiente fórmula:

$$p_m = \text{Suelo} (N - (N-C)/R_m - N_{t_{m-1}}),$$

15 donde p_m es el número de nodos variables a perforar, R_m es la tasa deseada, R_{m-1} es una tasa anterior, N y C son un número de nodos variables y nodos de comprobación en un grafo elevado respectivamente, $N_{t_{m-1}}$ es el número total de dichos nodos variables perforados en el grafo elevado correspondiente a un código anterior $C\{R_{m-1}\}$, y $\text{Suelo}()$ representa una función de redondeo.

20 74.- El codificador que genera códigos de tasa alta a partir de códigos de tasa baja según el ejemplo 69, en el que dicha instrucción para calcular alfa y beta de manera que un patrón de perforación en el conjunto no perforado sea regular utiliza las siguientes fórmulas:

$$\text{alfa} = \text{Suelo} [(r_{m-1} - p_m) / (p_m - 1)]$$

y

$$\text{beta} = r_{m-1} - \text{alfa} (p_m - 1) - p_m,$$

25 donde p_m es el número de nodos variables a perforar, r_{m-1} es la cardinalidad del conjunto de nodos variables redundantes no perforados en un código anterior $C\{R_{m-1}\}$, R_{m-1} es una tasa anterior y $\text{Suelo}()$ representa una función de redondeo.

30 75.- El codificador que genera códigos de tasa alta a partir de códigos de tasa baja según la realización 69, en el que dicha instrucción para perforar un patrón que se obtiene en función de dicho alfa y dicho beta en dicho conjunto no perforado comprende utilizar un patrón $[N_1 \{ \text{alfa nodos} \} N_2 \{ \text{alfa nodos} \} \dots N_m \{ \text{beta nodos} \}]$, donde N_1, N_2, \dots, N_{p_m} son los índices de los nodos variables perforados del conjunto de nodos redundantes no perforados de cardinalidad r_{m-1} .

35 76.- El codificador que genera códigos de tasa alta a partir de códigos de tasa baja según el ejemplo 70, que comprende además una instrucción para buscar de manera aleatoria dicho subconjunto de bits de palabra de código, que comprende:

i) elegir un código padre con una tasa base deseada;

ii) inicializar una relación de señal a ruido para cada tasa deseada en el conjunto de tasas deseadas;

iii) hallar un conjunto de nodos variables no perforados en un código anterior con una tasa anterior;

iv) calcular un número de nodos variables a perforar para pasar de dicha tasa anterior a una tasa deseada;

40 v) elegir de manera aleatoria dicho número de nodos variables a perforar a partir de dicho conjunto de nodos variables no perforados;

vi) construir un código a partir de la tasa anterior perforando dicho número elegido de nodos variables;

vii) modificar la densidad para probar si dicho código construido consigue un error despreciable con la relación de señal a ruido inicializada;

viii) buscar de manera iterativa un nuevo umbral de señal a ruido de dicho código construido, estableciendo dicho código construido como un código ganador y estableciendo dicha relación de señal a ruido igual a un nuevo umbral si dicho código construido consigue un error despreciable con dicha relación de señal a ruido; y

5 ix) repetir dicha búsqueda aleatoria repitiendo las etapas v a vii si dicho código construido no consigue un error despreciable con dicha relación de señal a ruido.

77.- El codificador que genera códigos de tasa alta a partir de códigos de tasa baja según la realización 71, en el que dicha instrucción para perforar de manera progresiva nodos variables para obtener un código deseado a partir de un código anterior comprende:

10 perforar de manera progresiva dichos nodos variables de un conjunto de dichos nodos variables no perforados de cardinalidad r_{m-1} , donde r_{m-1} es el número de elementos del conjunto de nodos variables redundantes no perforados en un código anterior y donde los nodos variables redundantes no perforados del código anterior pueden perforarse para proporcionar el código de una tasa deseada.

15 78.- El codificador que genera códigos de tasa alta a partir de códigos de tasa baja según el ejemplo 76, en el que dicha instrucción para buscar de manera iterativa un nuevo umbral de dicho código construido comprende:

a) bisecar de manera iterativa dicho intervalo para seleccionar una relación de señal a ruido de prueba;

b) determinar si dicha relación de señal a ruido de prueba consigue un error cero; y

c) repetir las etapas a y b hasta que se obtenga una precisión deseada en dicha relación de señal a ruido.

20 79.- El codificador que genera códigos de tasa alta a partir de códigos de tasa baja según la realización 77, en el que dicha instrucción para perforar de manera progresiva nodos variables de un conjunto de dichos nodos variables no perforados comprende una instrucción de inicialización que comprende:

calcular para cada nodo de comprobación conectado a al menos un dicho nodo variable una pluralidad de dichos nodos variables perforados conectados a dichos nodos de comprobación, donde cada uno de dichos nodos de comprobación está conectado solamente a un dicho nodo variable perforado;

25 hallar dicho conjunto de dichos nodos variables no perforados;

calcular para cada dicho nodo variable no perforado una pluralidad de dichos nodos de comprobación conectados a cada dicho nodo variable no perforado, donde cada uno de dichos nodos de comprobación está conectado solamente a un dicho nodo variable perforado; y

30 calcular para cada dicho nodo variable no perforado una suma de dichos nodos variables perforados conectados a dichos nodos de comprobación, donde dichos nodos de comprobación están conectados a dichos nodos variables no perforados.

80.- El codificador que genera códigos de tasa alta a partir de códigos de tasa baja según la realización 79, en el que dicha instrucción para perforar de manera progresiva nodos variables de un conjunto de dichos nodos variables no perforados comprende además una instrucción de perforación para hallar dicho nodo variable perforado mientras el número de dichos nodos variables perforados en esta instrucción es inferior al número de dichos nodos variables perforados para dicha tasa deseada.

81.- El codificador que genera códigos de tasa alta a partir de códigos de tasa baja según la realización 80, en el que dicha instrucción para hallar dicho nodo variable perforado mientras el número de dichos nodos variables perforados en esta instrucción es inferior al número de dichos nodos variables perforados para dicha tasa deseada comprende:

a) hallar un segundo conjunto de dichos nodos variables no perforados en dicho conjunto de dichos nodos variables no perforados para los que un número de nodos adyacentes de dichos nodos de comprobación es un valor mínimo;

45 b) fijar el nodo variable perforado como el nodo del segundo conjunto si dicho número de dichos nodos variables no perforados en dicho segundo conjunto de la etapa a es 1;

c) ejecutar las instrucciones ci a civ si el número de dichos nodos variables no perforados de dicho segundo conjunto de la etapa a es > 1 ;

ci) hallar un subconjunto de dicho segundo conjunto de dichos nodos variables no perforados de la

instrucción a, donde la suma de dicho número de nodos variables perforados conectados a dichos nodos de comprobación que están conectados a dichos nodos variables no perforados es un valor mínimo;

5 cii) fijar dichos nodos variables perforados al único nodo del subconjunto de la etapa ci si el número de dichos nodos variables no perforados de dicho subconjunto de la instrucción ci es 1;

ciiii) elegir dichos nodos variables a partir de dichos nodos variables no perforados del subconjunto de la instrucción ci de manera aleatoria si el número de los nodos variables no perforados del subconjunto de la instrucción ci es mayor que 1; y

10 civ) perforar dicho nodo variable elegido y eliminar dicho nodo variable elegido y perforado del conjunto de nodos variables no perforados.

82.- El codificador que genera códigos de tasa alta a partir de códigos de tasa baja según la realización 81, en el que dicha instrucción para perforar de manera progresiva nodos variables de un conjunto de dichos nodos variables no perforados comprende además una instrucción de actualización, que comprende:

15 actualizar dicho número de nodos variables perforados conectados a los nodos de comprobación para cada uno de dichos nodos de comprobación adyacentes a dichos nodos variables perforados;

actualizar el número de dichos nodos de comprobación conectados a dichos nodos variables no perforados, donde dicho nodo de comprobación está conectado solamente a un dicho nodo variable perforado para cada dicho nodo variable adyacente a dicho nodo de comprobación; y

20 actualizar la suma de dichos nodos variables perforados conectados a dicho nodo de comprobación, donde dicho nodo de comprobación está conectado a dichos nodos variables no perforados para cada dicho nodo variable adyacente a dicho nodo de comprobación.

83.- El codificador que genera códigos de tasa alta a partir de códigos de tasa baja según la realización 82, en el que dicha instrucción para perforar de manera progresiva nodos variables de un conjunto de dichos nodos variables no perforados comprende además una instrucción para maximizar la conectividad entre los nodos de comprobación con solamente un nodo variable perforado y los otros nodos variables perforados.

84.- El codificador que genera códigos de tasa alta a partir de códigos de tasa baja según la realización 83, en el que dicha instrucción para maximizar la conectividad entre los nodos de comprobación con solamente un nodo variable perforado y los otros nodos variables perforados comprende:

30 inicializar un resultado de perforación a cero;

modificar una semilla aleatoria;

ejecutar dichas etapas de instrucciones de las realizaciones 71, 77, 79, 80 y 81 en dicho código anterior para obtener dicho código perforado deseado;

calcular el resultado de perforación del código perforado deseado; y

35 elegir el patrón de perforación con el mayor resultado de perforación si el resultado de perforación del código perforado deseado es mayor que el resultado de perforación del código anterior.

85.- Un terminal de acceso que genera protografos de tasa baja a partir de protografos de tasa alta, que comprende:

un transmisor;

un receptor conectado de manera operativa a dicho transmisor;

40 un procesador conectado de manera operativa a dicho transmisor y a dicho receptor;

una antena conectada de manera operativa a dicho transmisor y a dicho receptor; y

una memoria conectada de manera operativa a dicho procesador, en la que dicho terminal de acceso está adaptado para ejecutar instrucciones almacenadas en dicha memoria, que comprenden:

copiar un grafo base;

45 permutar puntos de extremo de aristas de un mismo tipo en copias de dicho grafo base para producir

un grafo permutado; y

podar nodos de entrada sistemáticos en dicho grafo permutado y dichas aristas conectadas a los mismos.

- 5 86.- El terminal de acceso según la realización 85, en el que dicha instrucción para permutar puntos de extremo de aristas de un mismo tipo en dichas copias de dicho grafo base comprende además utilizar un procedimiento de crecimiento progresivo de aristas para maximizar la cintura de dicho grafo base.
- 87.- El terminal de acceso según la realización 85, en el que dicha instrucción para permutar puntos de extremo de aristas de un mismo tipo en dichas copias de dicho grafo base comprende además establecer aristas o conexiones entre nodos de símbolo y de comprobación por cada arista.
- 10 88.- El terminal de acceso según la realización 85, en el que dicha instrucción para podar nodos de entrada sistemáticos comprende además podar $(N-C)(m-2)$ nodos de entrada sistemáticos, donde N representa un número de nodos variables, C representa un número de nodos de comprobación y $1/m$ es la tasa deseada.
- 89.- El terminal de acceso según la realización 85, en el que dicha instrucción para podar nodos de entrada sistemáticos comprende además: podar $pK \frac{R_B - R_L}{R_B(1 - R_L)}$ nodos de entrada sistemáticos y las aristas conectadas a los mismos, donde p es el número de copias del grafo base, K es el número de nodos variables menos el número de nodos de comprobación, R_B es la tasa base deseada y R_L es la tasa de diseño.
- 15 90.- El terminal de acceso según la realización 85, que comprende además una instrucción para optimizar dichos nodos de entrada podados utilizando una combinación de añadir, quitar y permutar dichas aristas de dicho grafo permutado podado.
- 20 91.- El terminal de acceso según la realización 86, en el que dicha instrucción para permutar puntos de extremo de aristas de un mismo tipo en dichas copias de dicho grafo base comprende además repetir dicha instrucción para permutar puntos de extremo de aristas de un mismo tipo en las copias de dicho grafo base para producir un grafo permutado.
- 25 92.- El terminal de acceso según la realización 86, en el que dicho procedimiento de crecimiento progresivo de aristas es un procedimiento PEG cíclico.
- 93.- El terminal de acceso según la realización 88, en el que dicha instrucción para copiar un grafo base comprende copiar dicho grafo base m-1 veces.
- 94.- El terminal de acceso según la realización 93, en el que dicha instrucción para permutar puntos de extremo de aristas de un mismo tipo en dichas m-1 copias de dicho grafo base comprende además:
- 30 utilizar un procedimiento de crecimiento progresivo de aristas para maximizar la cintura de dicho grafo base; repetir dicha instrucción para permutar puntos de extremo de aristas de un mismo tipo en las copias de dicho grafo base para producir un grafo permutado; y elegir el grafo permutado que tenga el menor número de ciclos más cortos si el procedimiento de crecimiento progresivo de aristas requiere una semilla aleatoria.
- 35 95.- Un terminal de acceso que genera códigos de tasa alta a partir de códigos de tasa baja, que comprende:
- un transmisor;
- un receptor conectado de manera operativa a dicho transmisor;
- un procesador conectado de manera operativa a dicho transmisor y a dicho receptor;
- una antena conectada de manera operativa a dicho transmisor y a dicho receptor; y
- 40 una memoria conectada de manera operativa a dicho procesador, en la que dicho terminal de acceso está adaptado para ejecutar instrucciones de software almacenadas en dicha memoria, que comprenden:
- perforar un subconjunto de bits de palabra de código.
- 96.- El terminal de acceso que genera códigos de tasa alta a partir de códigos de tasa baja según la realización 95, en el que dichos códigos de tasa baja son códigos padre.

97.- El terminal de acceso que genera códigos de tasa alta a partir de códigos de tasa baja según la realización 95, en el que dicha instrucción para perforar un subconjunto de bits de palabra de código comprende una perforación regular-irregular de dicho subconjunto de bits de palabra de código, que comprende:

elegir un código padre con una tasa base deseada;

5 hallar un conjunto de nodos variables no perforados en un código anterior para cada tasa en un conjunto de tasas deseadas;

calcular un número de nodos variables a perforar para pasar de una tasa anterior a una tasa deseada;

calcular alfa y beta de manera que un patrón de perforación en un conjunto no perforado sea regular; y

perforar un patrón que se obtiene en función de dicho alfa y dicho beta en dicho conjunto no perforado.

10 98.- El terminal de acceso que genera códigos de tasa alta a partir de códigos de tasa baja según la realización 95, en el que dicha instrucción para perforar un subconjunto de bits de palabra de código comprende perforar de manera aleatoria nodos variables.

99.- El terminal de acceso que genera códigos de tasa alta a partir de códigos de tasa baja según la realización 95, en el que dicha instrucción para perforar un subconjunto de bits de palabra de código comprende:

15 perforar de manera progresiva nodos variables para obtener un código deseado a partir de un código anterior.

100.- El terminal de acceso que genera códigos de tasa alta a partir de códigos de tasa baja según la realización 96, en el que dichos códigos padre se generan ejecutando las siguientes instrucciones:

copiar un grafo base;

20 permutar puntos de extremo de aristas de un mismo tipo en copias de dicho grafo base para producir un grafo permutado; y

podar nodos de entrada sistemáticos en dicho grafo permutado y dichas aristas conectadas a los mismos.

101.- El terminal de acceso que genera códigos de tasa alta a partir de códigos de tasa baja según el ejemplo 97, en el que dicha instrucción para calcular una pluralidad de nodos variables a perforar para pasar de una tasa anterior a una tasa deseada utiliza la siguiente fórmula:

25

$$p_m = \text{Suelo} (N - (N-C)/R_m - N_{t_{m-1}}),$$

donde p_m es el número de nodos variables a perforar, R_m es la tasa deseada, R_{m-1} es una tasa anterior, N y C son un número de nodos variables y de nodos de comprobación en un grafo elevado respectivamente, $N_{t_{m-1}}$ es el número total de dichos nodos variables perforados en el grafo elevado correspondiente a un código anterior $C\{R_{m-1}\}$, y $\text{Suelo}()$ representa una función de redondeo.

30

102.- El terminal de acceso que genera códigos de tasa alta a partir de códigos de tasa baja según el ejemplo 97, en el que dicha instrucción para calcular alfa y beta de manera que un patrón de perforación en el conjunto no perforado sea regular, utiliza las siguientes fórmulas:

$$\text{alfa} = \text{Suelo} [(r_{m-1} - p_m) / (p_m - 1)]$$

35 y

$$\text{beta} = r_{m-1} - \text{alfa} (p_m - 1) - p_m,$$

donde p_m es el número de nodos variables a perforar, r_{m-1} es la cardinalidad del conjunto de nodos variables redundantes no perforados en un código anterior $C\{R_{m-1}\}$, R_{m-1} es una tasa anterior y $\text{Suelo}()$ representa una función de redondeo.

40 103.- El terminal de acceso que genera códigos de tasa alta a partir de códigos de tasa baja según la realización 97, en el que dicha instrucción para perforar un patrón que se obtiene en función de dicho alfa y dicho beta en dicho conjunto no perforado comprende utilizar un patrón $[N_1 \{ \text{alfa nodos} \} N_2 \{ \text{alfa nodos} \} \dots N_m \{ \text{beta nodos} \}]$, donde N_1 , N_2 , ..., N_{p_m} son los índices de los nodos variables perforados del conjunto de nodos redundantes no perforados de cardinalidad r_{m-1} .

45 104.- El terminal de acceso que genera códigos de tasa alta a partir de códigos de tasa baja según el ejemplo 98,

que comprende además una instrucción para buscar de manera aleatoria dicho subconjunto de bits de palabra de código, que comprende:

- i) elegir un código padre con una tasa base deseada;
- ii) inicializar una relación de señal a ruido para cada tasa deseada en el conjunto de tasas deseadas;
- 5 iii) hallar un conjunto de nodos variables no perforados en un código anterior con una tasa anterior;
- iv) calcular un número de nodos variables a perforar para pasar de dicha tasa anterior a una tasa deseada;
- v) elegir de manera aleatoria dicho número de nodos variables a perforar a partir de dicho conjunto de nodos variables no perforados;
- vi) construir un código a partir de la tasa anterior perforando dicho número elegido de nodos variables;
- 10 vii) modificar la densidad para probar si dicho código construido consigue un error despreciable con la relación de señal a ruido inicializada;
- viii) buscar de manera iterativa un nuevo umbral de señal a ruido de dicho código construido, estableciendo dicho código construido como un código ganador y estableciendo dicha relación de señal a ruido igual a un nuevo umbral si dicho código construido consigue un error despreciable con dicha relación de señal a ruido;
- 15 y
- ix) repetir dicha búsqueda aleatoria repitiendo las etapas v a vii si dicho código construido no consigue un error despreciable con dicha relación de señal a ruido.

105.- El terminal de acceso que genera códigos de tasa alta a partir de códigos de tasa baja según la realización 99, en el que dicha instrucción para perforar de manera progresiva nodos variables para obtener un código deseado a partir de un código anterior comprende:

perforar de manera progresiva dichos nodos variables de un conjunto de dichos nodos variables no perforados de cardinalidad r_{m-1} , donde r_{m-1} es el número de elementos del conjunto de nodos variables redundantes no perforados en un código anterior y donde los nodos variables redundantes no perforados del código anterior pueden perforarse para proporcionar el código de una tasa deseada.

25 106.- El terminal de acceso que genera códigos de tasa alta a partir de códigos de tasa baja según el ejemplo 104, en el que dicha instrucción para buscar de manera iterativa un nuevo umbral de dicho código construido comprende:

- a) bisecar de manera iterativa dicho intervalo para seleccionar una relación de señal a ruido de prueba;
- b) determinar si dicha relación de señal a ruido de prueba consigue un error cero; y
- 30 c) repetir las etapas a y b hasta que se obtenga una precisión deseada en dicha relación de señal a ruido.

107.- El terminal de acceso que genera códigos de tasa alta a partir de códigos de tasa baja según la realización 105, en el que dicha instrucción para perforar de manera progresiva nodos variables de un conjunto de dichos nodos variables no perforados comprende una instrucción de inicialización que comprende:

35 calcular para cada nodo de comprobación conectado a al menos un dicho nodo variable una pluralidad de dichos nodos variables perforados conectados a dichos nodos de comprobación, donde cada uno de dichos nodos de comprobación está conectado solamente a un dicho nodo variable perforado;

hallar dicho conjunto de dichos nodos variables no perforados;

40 calcular para cada dicho nodo variable no perforado una pluralidad de dichos nodos de comprobación conectados a cada dicho nodo variable no perforado, donde cada uno de dichos nodos de comprobación está conectado solamente a un dicho nodo variable perforado; y

calcular para cada dicho nodo variable no perforado una suma de dichos nodos variables perforados conectados a dichos nodos de comprobación, donde dichos nodos de comprobación están conectados a dichos nodos variables no perforados.

45 108.- El terminal de acceso que genera códigos de tasa alta a partir de códigos de tasa baja según la realización 107, en el que dicha instrucción para perforar de manera progresiva nodos variables de un conjunto de dichos nodos variables no perforados comprende además una instrucción de perforación para hallar dicho nodo variable

perforado mientras el número de dichos nodos variables perforados en esta instrucción es inferior al número de dichos nodos variables perforados para dicha tasa deseada.

5 109.- El terminal de acceso que genera códigos de tasa alta a partir de códigos de tasa baja según la realización 108, en el que dicha instrucción para hallar dicho nodo variable perforado mientras el número de dichos nodos variables perforados en esta instrucción es inferior al número de dichos nodos variables perforados para dicha tasa deseada comprende:

a) hallar un segundo conjunto de dichos nodos variables no perforados en dicho conjunto de dichos nodos variables no perforados para los que un número de nodos adyacentes de dichos nodos de comprobación es un valor mínimo;

10 b) fijar el nodo variable perforado como el nodo del segundo conjunto si dicho número de dichos nodos variables no perforados en dicho segundo conjunto de la etapa a es 1;

c) ejecutar las instrucciones ci a civ si el número de dichos nodos variables no perforados de dicho segundo conjunto de la etapa a es > 1 ;

15 ci) hallar un subconjunto de dicho segundo conjunto de dichos nodos variables no perforados de la instrucción a, donde la suma de dicho número de nodos variables perforados conectados a dichos nodos de comprobación que están conectados a dichos nodos variables no perforados es un valor mínimo;

cii) fijar dichos nodos variables perforados al único nodo del subconjunto de la etapa ci si el número de dichos nodos variables no perforados de dicho subconjunto de la instrucción ci es 1;

20 ciii) elegir dichos nodos variables a partir de dichos nodos variables no perforados del subconjunto de la instrucción ci de manera aleatoria si el número de los nodos variables no perforados del subconjunto de la instrucción ci es mayor que 1; y

civ) perforar dicho nodo variable elegido y eliminar dicho nodo variable elegido y perforado del conjunto de nodos variables no perforados.

25 110.- El terminal de acceso que genera códigos de tasa alta a partir de códigos de tasa baja según la realización 109, en el que dicha instrucción para perforar de manera progresiva nodos variables de un conjunto de dichos nodos variables no perforados comprende además una instrucción de actualización, que comprende:

actualizar dicho número de nodos variables perforados conectados a los nodos de comprobación para cada uno de dichos nodos de comprobación adyacentes a dichos nodos variables perforados;

30 actualizar el número de dichos nodos de comprobación conectados a dichos nodos variables no perforados, donde dicho nodo de comprobación está conectado solamente a un dicho nodo variable perforado para cada dicho nodo variable adyacente a dicho nodo de comprobación; y

35 actualizar la suma de dichos nodos variables perforados conectados a dicho nodo de comprobación, donde dicho nodo de comprobación está conectado a dichos nodos variables no perforados para cada dicho nodo variable adyacente a dicho nodo de comprobación.

111.- El terminal de acceso que genera códigos de tasa alta a partir de códigos de tasa baja según la realización 110, en el que dicha instrucción para perforar de manera progresiva nodos variables de un conjunto de dichos nodos variables no perforados comprende además una instrucción para maximizar la conectividad entre los nodos de comprobación con solamente un nodo variable perforado y los otros nodos variables perforados.

40 112.- El terminal de acceso que genera códigos de tasa alta a partir de códigos de tasa baja según la realización 111, en el que dicha instrucción para maximizar la conectividad entre los nodos de comprobación con solamente un nodo variable perforado y los otros nodos variables perforados comprende:

inicializar un resultado de perforación a cero;

modificar una semilla aleatoria;

45 ejecutar dichas etapas de instrucciones de las realizaciones 99, 105, 107, 108 y 109 en dicho código anterior para obtener dicho código perforado deseado;

calcular el resultado de perforación del código perforado deseado; y

elegir el patrón de perforación con el mayor resultado de perforación si el resultado de perforación del código perforado deseado es mayor que el resultado de perforación del código anterior.

REIVINDICACIONES

1. Un procedimiento para generar códigos de tasa compatible LDPC basados en protografos de tasa alta a partir de códigos de tasa baja para codificar datos para su transmisión a través de un canal de comunicaciones:

en el que dichos códigos de tasa baja son códigos padres, comprendiendo el procedimiento:

- 5 generar dichos códigos padre mediante:
 - copiar un grafo base;
 - permutar puntos de extremo de aristas de un mismo tipo en copias de dicho grafo base para producir un grafo permutado; y
 - 10 podar nodos de entrada sistemáticos (20) en dicho grafo permutado y dichas aristas conectadas a los mismos;
 - perforar un subconjunto de bits de palabra de código (30), comprendiendo:
 - perforar de manera progresiva nodos variables (20) para obtener un código deseado a partir de un código anterior en una familia de tasa compatible del código padre respectivo;
 - 15 donde dicha etapa de perforar de manera progresiva nodos variables (20) para obtener un código deseado a partir de un código anterior comprende:
 - perforar de manera progresiva dichos nodos variables (20) de un conjunto de nodos variables redundantes no perforados (20) de cardinalidad r_{m-1} , donde r_{m-1} es el número de elementos del conjunto de nodos variables redundantes no perforados (20) del código anterior y donde los nodos variables redundantes no perforados (20) del código anterior pueden perforarse para proporcionar el código de una tasa deseada;
 - 20 donde dicha etapa de perforar de manera progresiva nodos variables (20) del conjunto de nodos variables redundantes no perforados (20) comprende una etapa de inicialización que comprende:
 - 25 calcular para cada nodo de comprobación (25) conectado a al menos un nodo variable redundante (20) el número de nodos variables perforados (20) conectados a dicho nodo de comprobación (25);
 - hallar dicho conjunto de nodos variables redundantes no perforados (20);
 - 30 calcular para cada nodo variable redundante no perforado (20) el número de tales nodos de comprobación (25) conectados al nodo variable redundante no perforado (20) que están conectados solamente a un nodo variable perforado (20); y
 - calcular para cada nodo variable redundante no perforado (20) la suma de nodos variables perforados (20) conectados a tales nodos de comprobación (25) que están conectados a dicho nodo variable redundante no perforado (20).

35 2. Un medio para generar códigos de tasa compatible LDPC basados en protografos de tasa alta a partir de códigos de tasa baja para codificar datos para su transmisión a través de un canal de comunicaciones, comprendiendo el medio:

un medio para perforar un subconjunto de bits de palabra de código (30),

en el que dichos códigos de tasa baja son códigos padre, y

40 en el que dichos códigos padre se generan mediante:

- un medio para copiar un grafo base;
- un medio para permutar puntos de extremo de aristas de un mismo tipo en copias de dicho grafo base para producir un grafo permutado; y
- 45 un medio para podar nodos de entrada sistemáticos (20) en dicho grafo permutado y dichas aristas conectadas a los mismos;

donde dicho medio para perforar un subconjunto de bits de palabra de código (30) comprende un medio para perforar de manera progresiva nodos variables (20) para obtener un código deseado a partir de un código anterior,

5 en el que dicho medio para perforar de manera progresiva nodos variables (20) para obtener un código deseado a partir de un código anterior comprende:

10 un medio para perforar de manera progresiva dichos nodos variables (20) de un conjunto de nodos variables redundantes no perforados (20) de cardinalidad r_{m-1} , donde r_{m-1} es el número de elementos del conjunto de nodos variables redundantes no perforados (20) del código anterior y donde los nodos variables redundantes no perforados (20) del código anterior pueden perforarse para proporcionar el código de una tasa deseada;

donde dicho medio para perforar de manera progresiva nodos variables (20) del conjunto de nodos variables redundantes no perforados (20) comprende un medio de inicialización que comprende:

15 un medio para calcular para cada nodo de comprobación (25) conectado a al menos un nodo variable redundante (20) el número de nodos variables perforados (20) conectados a dicho nodo de comprobación (25);

un medio para calcular para cada nodo variable redundante no perforado (20) el número de dichos nodos de comprobación (25) conectados a cada nodo variable no perforado (20), donde cada dicho nodo de comprobación (25) está conectado solamente a un dicho nodo variable perforado (20); y

20 un medio para calcular para cada dicho nodo variable redundante no perforado (20) la suma de dichos nodos variables perforados (20) conectados a tales nodos de comprobación que están conectados a dicho nodo variable redundante no perforado (20).

3. El medio según la reivindicación 2, en el que el medio comprende un codificador (552) adaptado para generar códigos de tasa alta a partir de códigos de tasa baja, que comprende:

25 una memoria (2605), en la que dicho codificador (552) está adaptado para ejecutar instrucciones almacenadas en dicha memoria (2605), que comprenden:

perforar un subconjunto de bits de palabra de código (30).

4. El medio según la reivindicación 2, en el que el medio comprende un terminal de acceso (106) que genera códigos de tasa alta a partir de códigos de tasa baja, que comprende:

30 un transmisor (2608);

un receptor (2610) conectado de manera operativa a dicho transmisor (2608);

un procesador (2602) conectado de manera operativa a dicho transmisor (2608) y a dicho receptor (2610);

una antena (2614) conectada de manera operativa a dicho transmisor (2608) y a dicho receptor (2610); y

35 una memoria (2605) conectada de manera operativa a dicho procesador (2602), en la que dicho terminal de acceso (106) está adaptado para ejecutar instrucciones de software almacenadas en dicha memoria (2605) que comprenden:

perforar un subconjunto de bits de palabra de código (30).

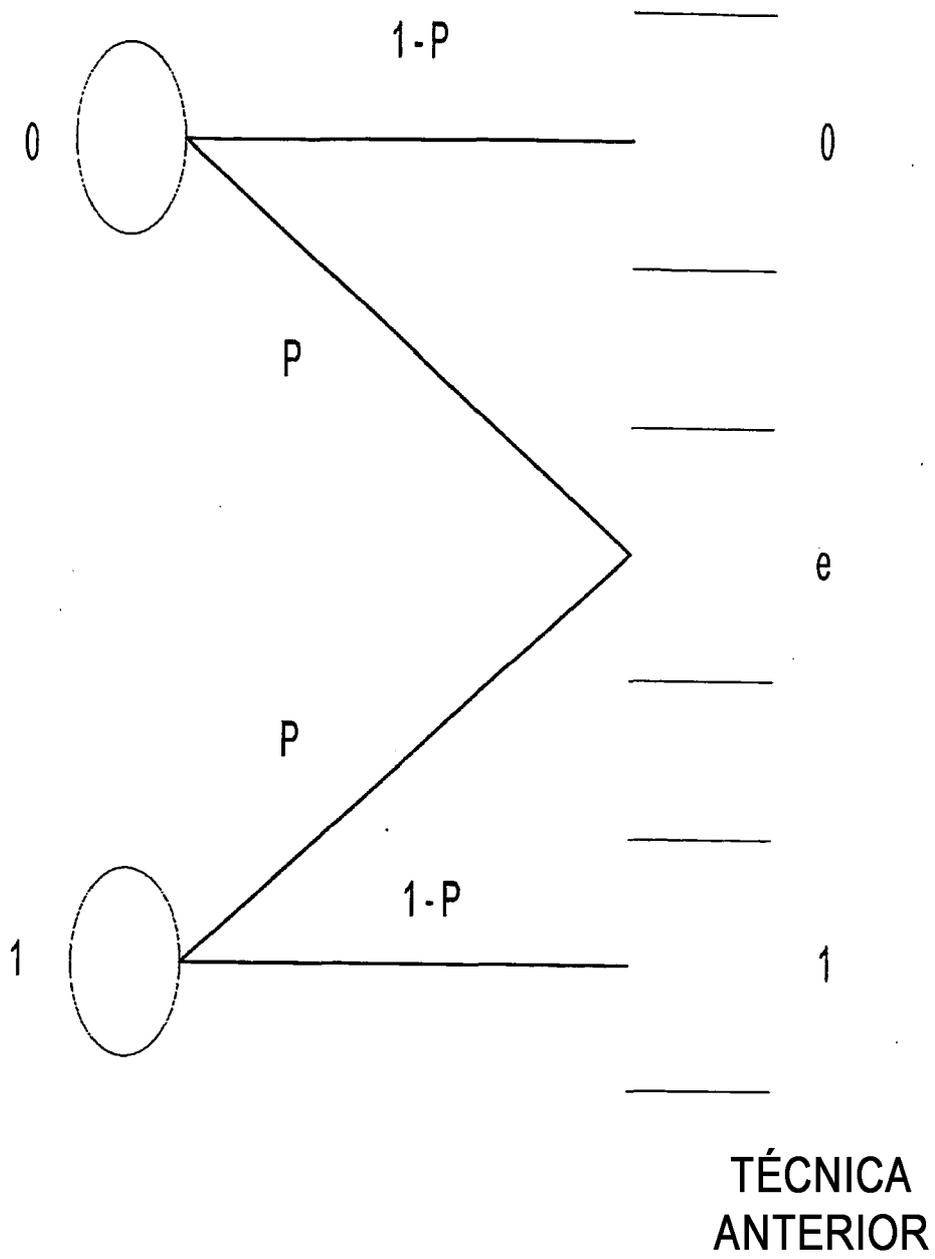
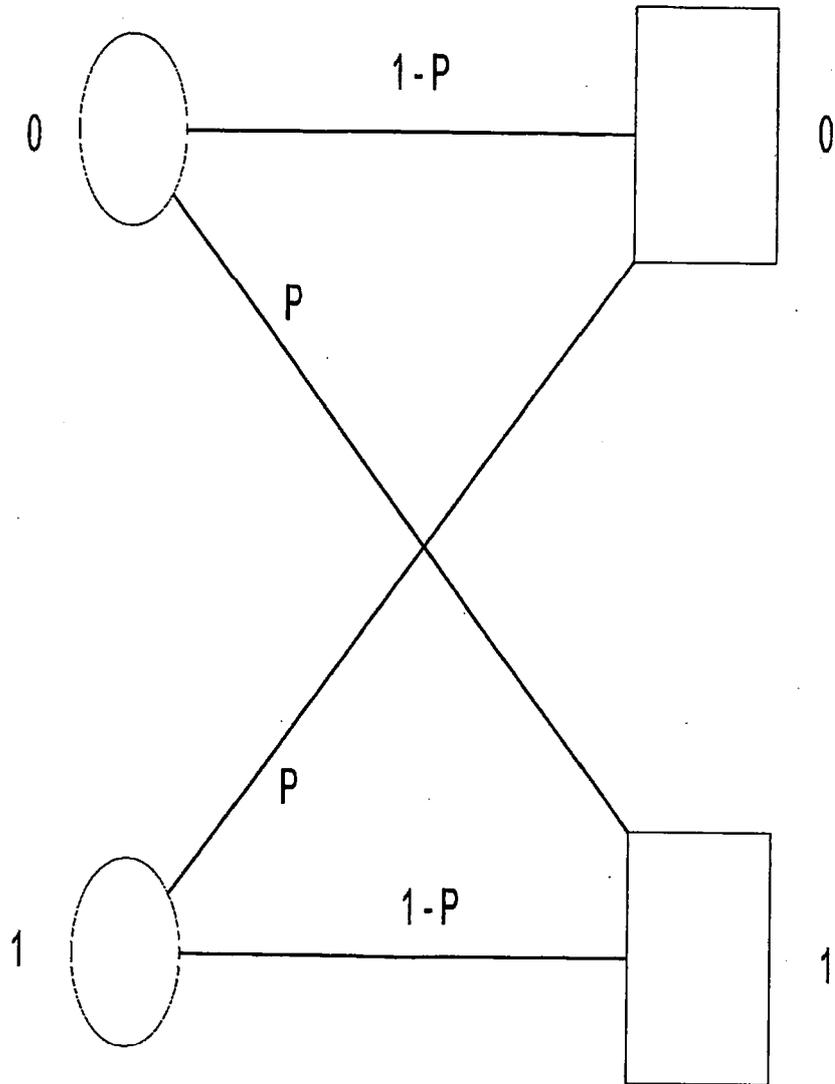


FIG. 1A



TÉCNICA
ANTERIOR

FIG. 1B

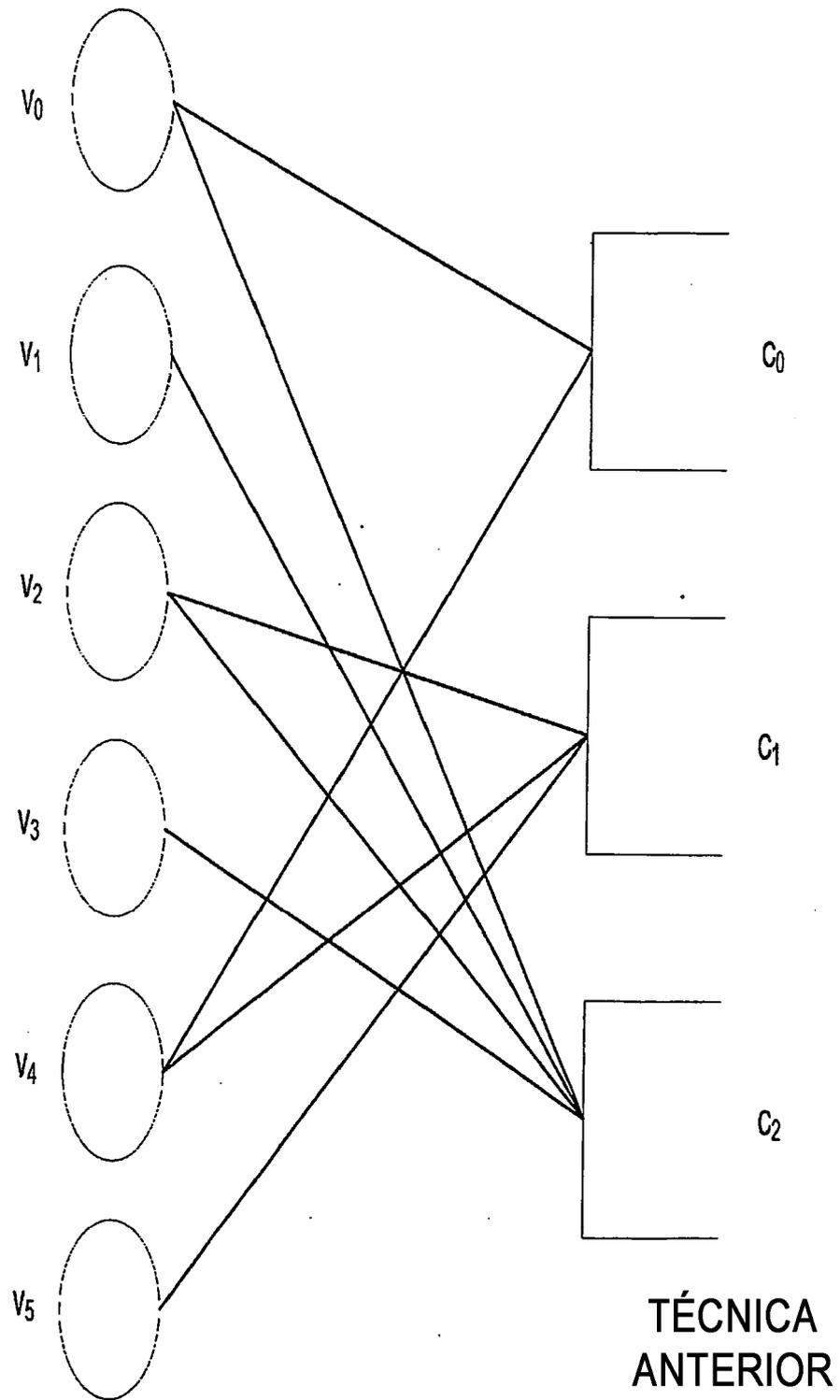


FIG. 2

V_0	V_1	V_2	V_3	V_4	V_5	
1	0	0	0	1	0	C_0
0	0	1	0	1	1	C_1
1	1	1	1	0	0	C_2

TÉCNICA
ANTERIOR

FIG. 3

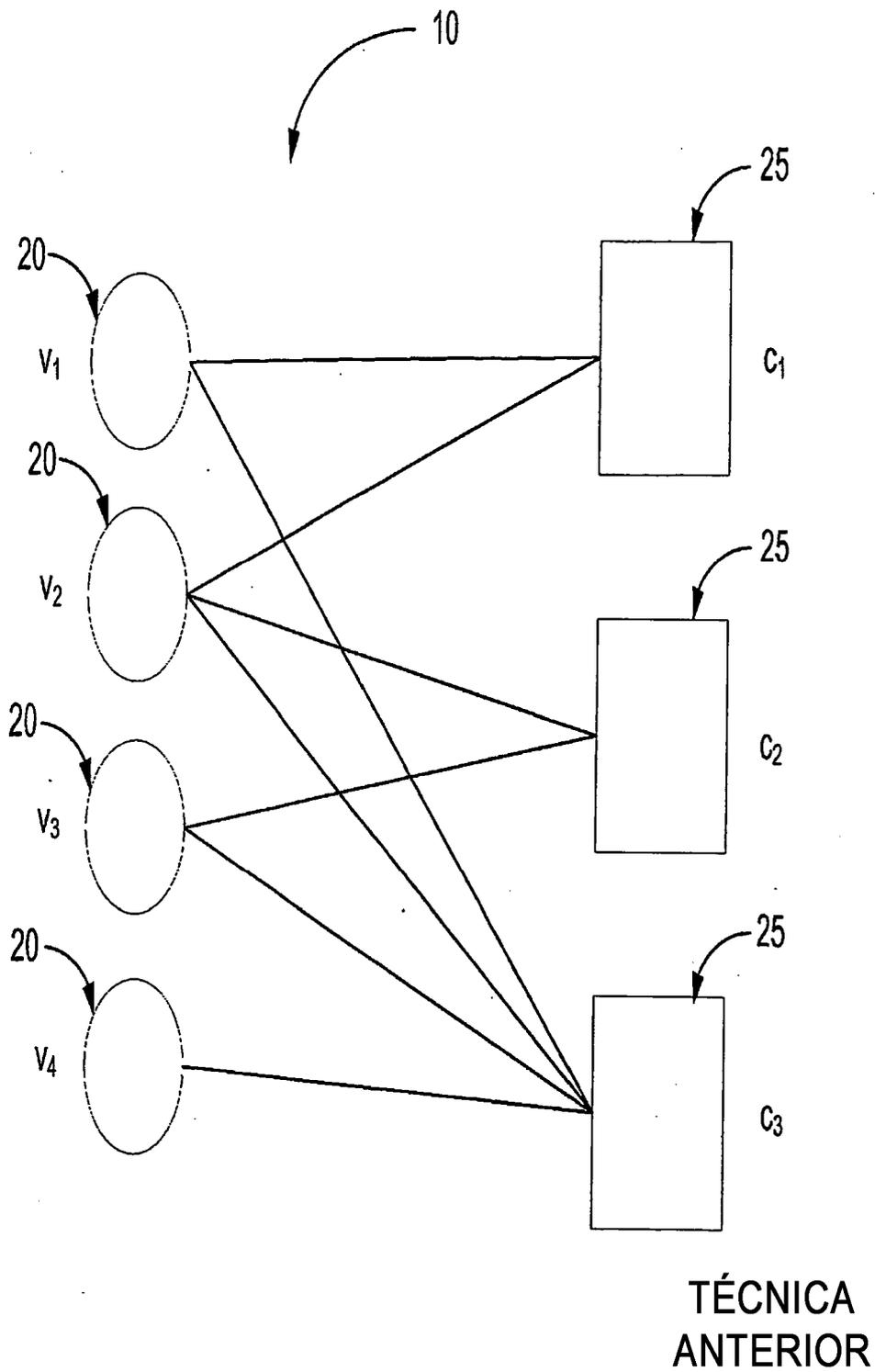


FIG. 4

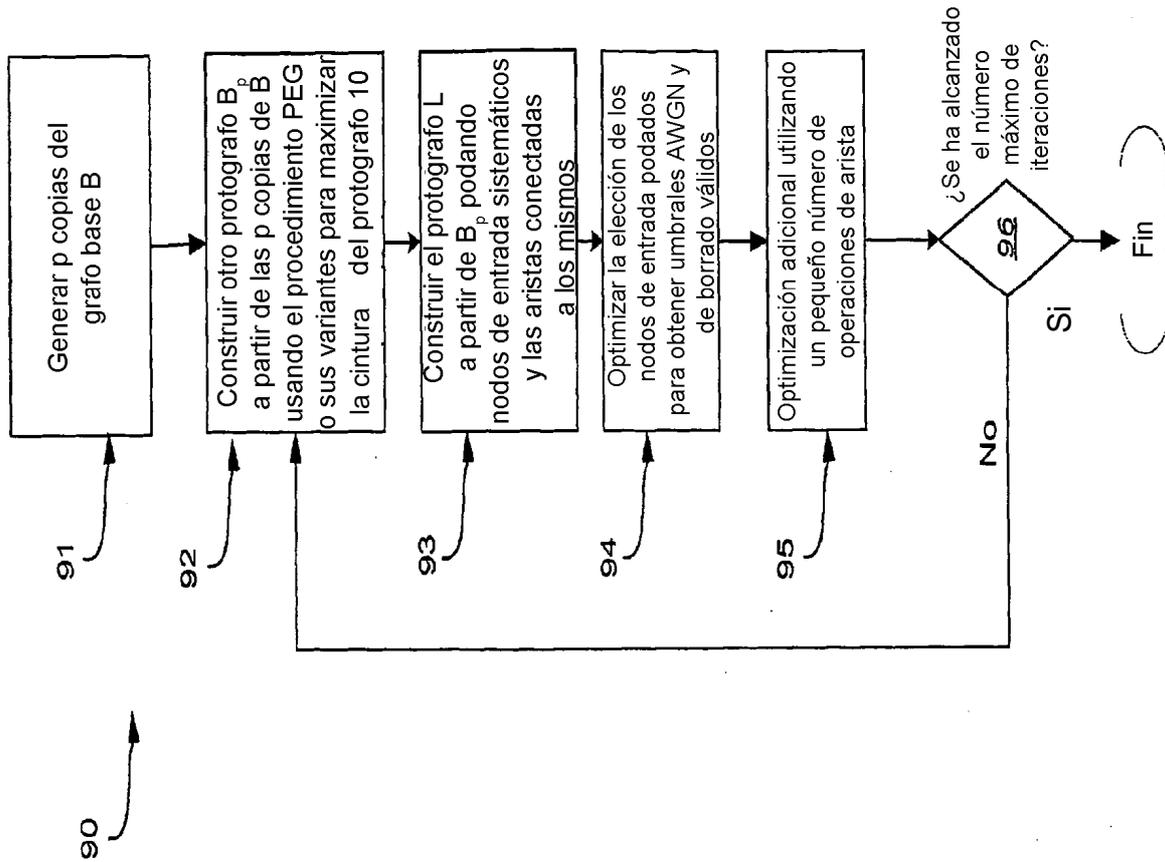


FIG. 5A

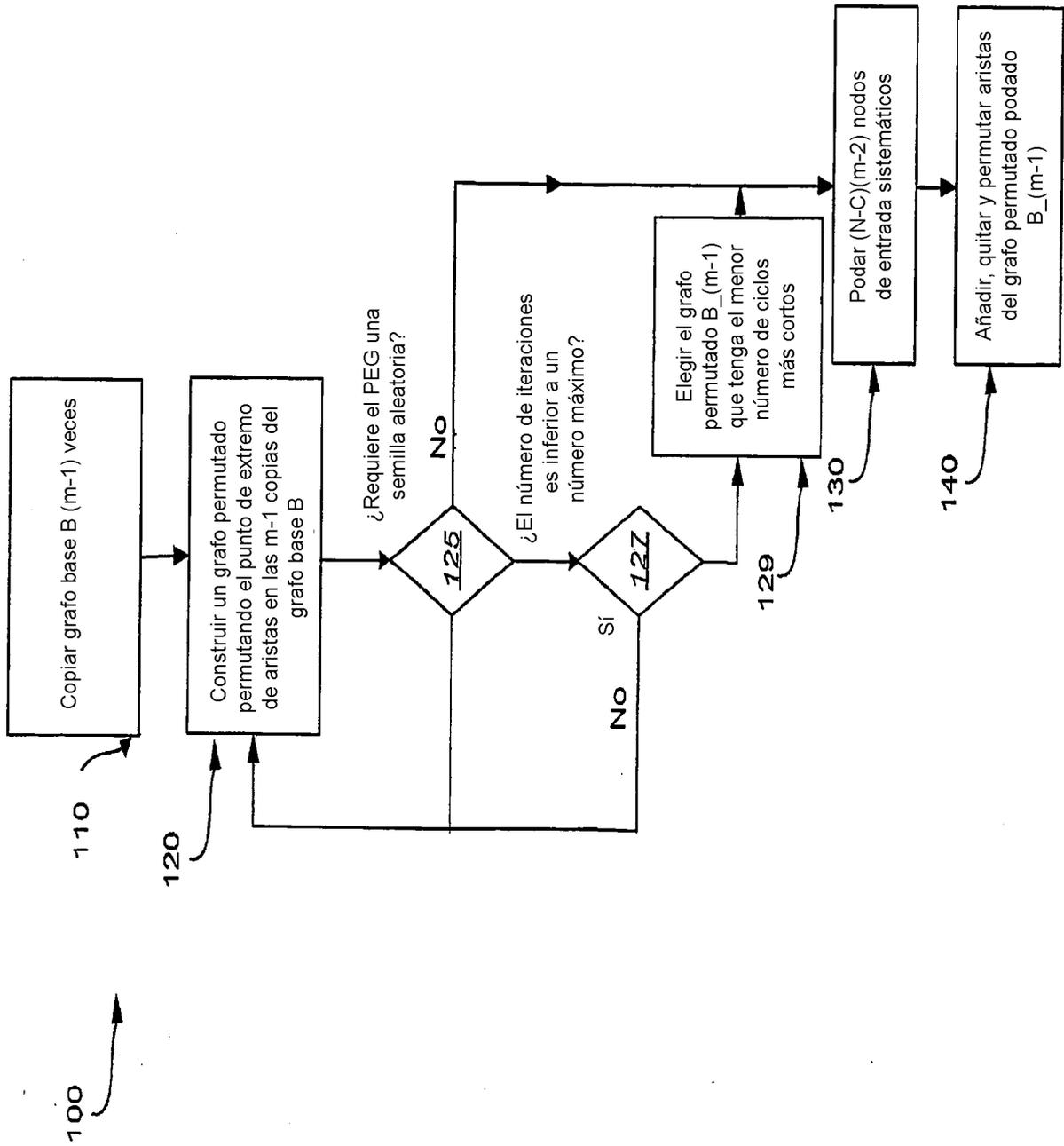


FIG. 5B

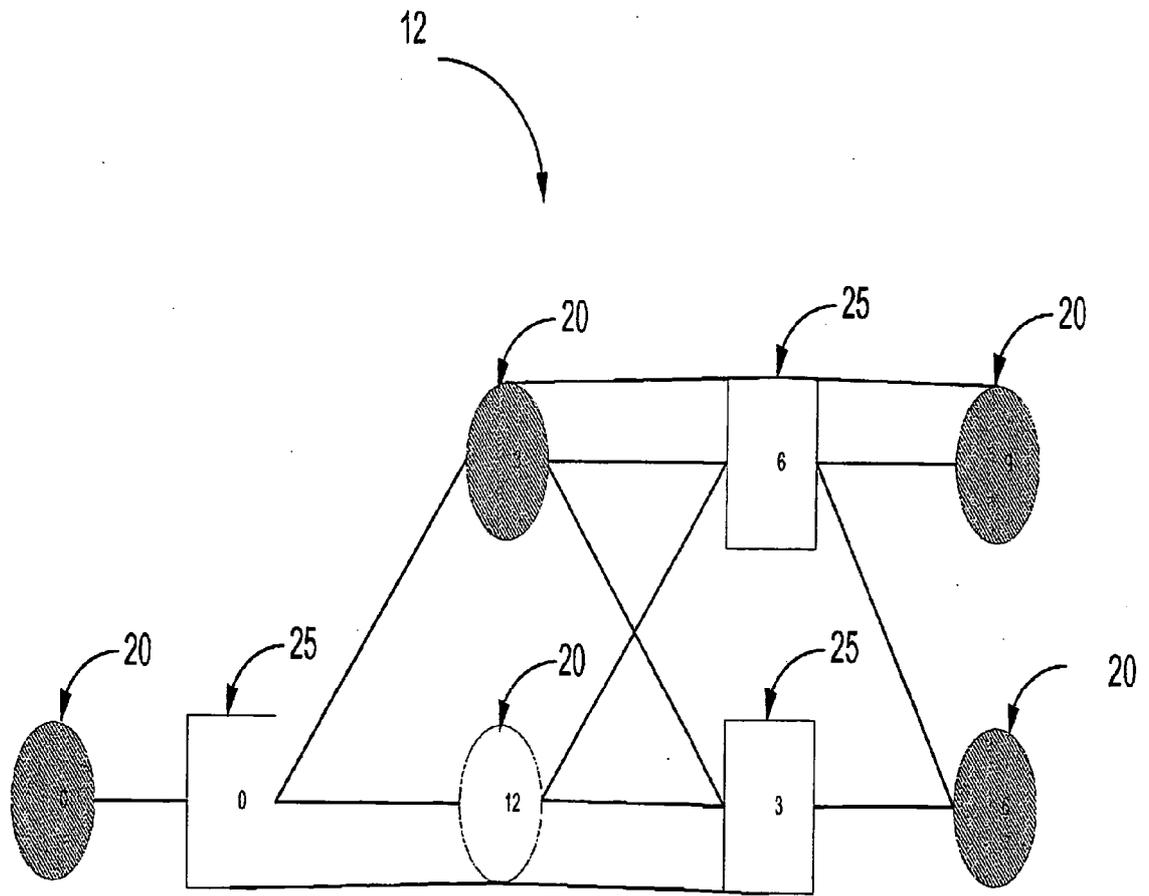


FIG. 6

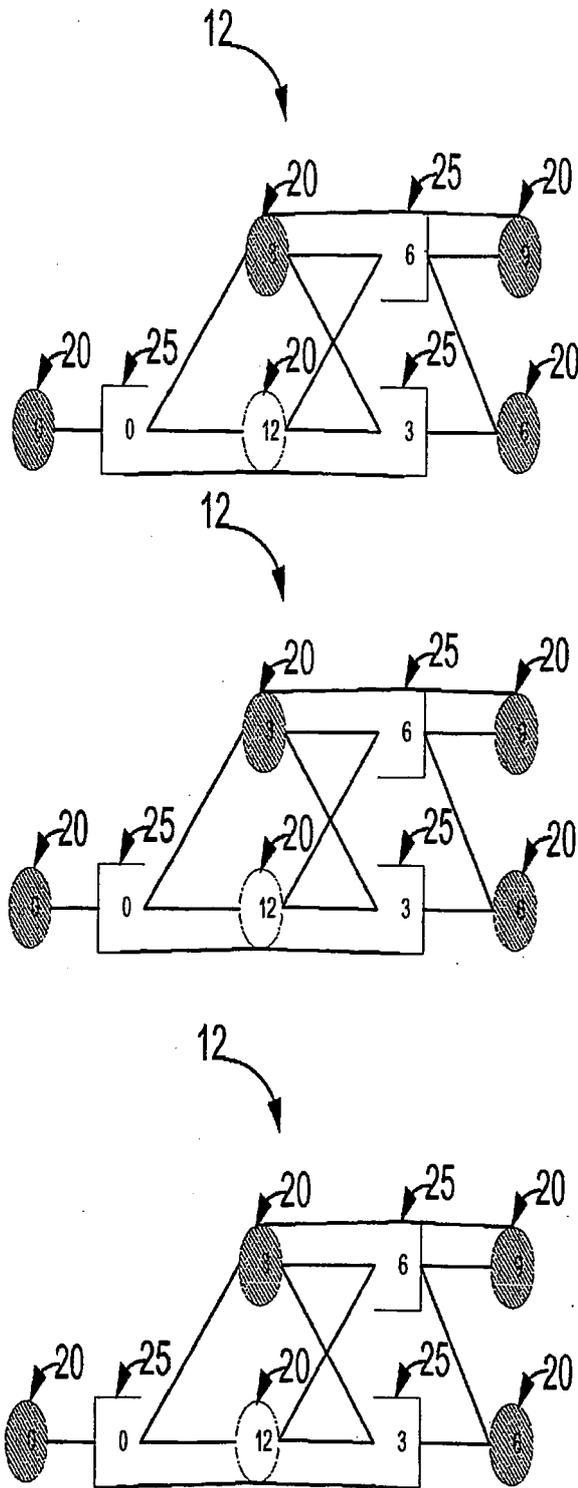


FIG. 7

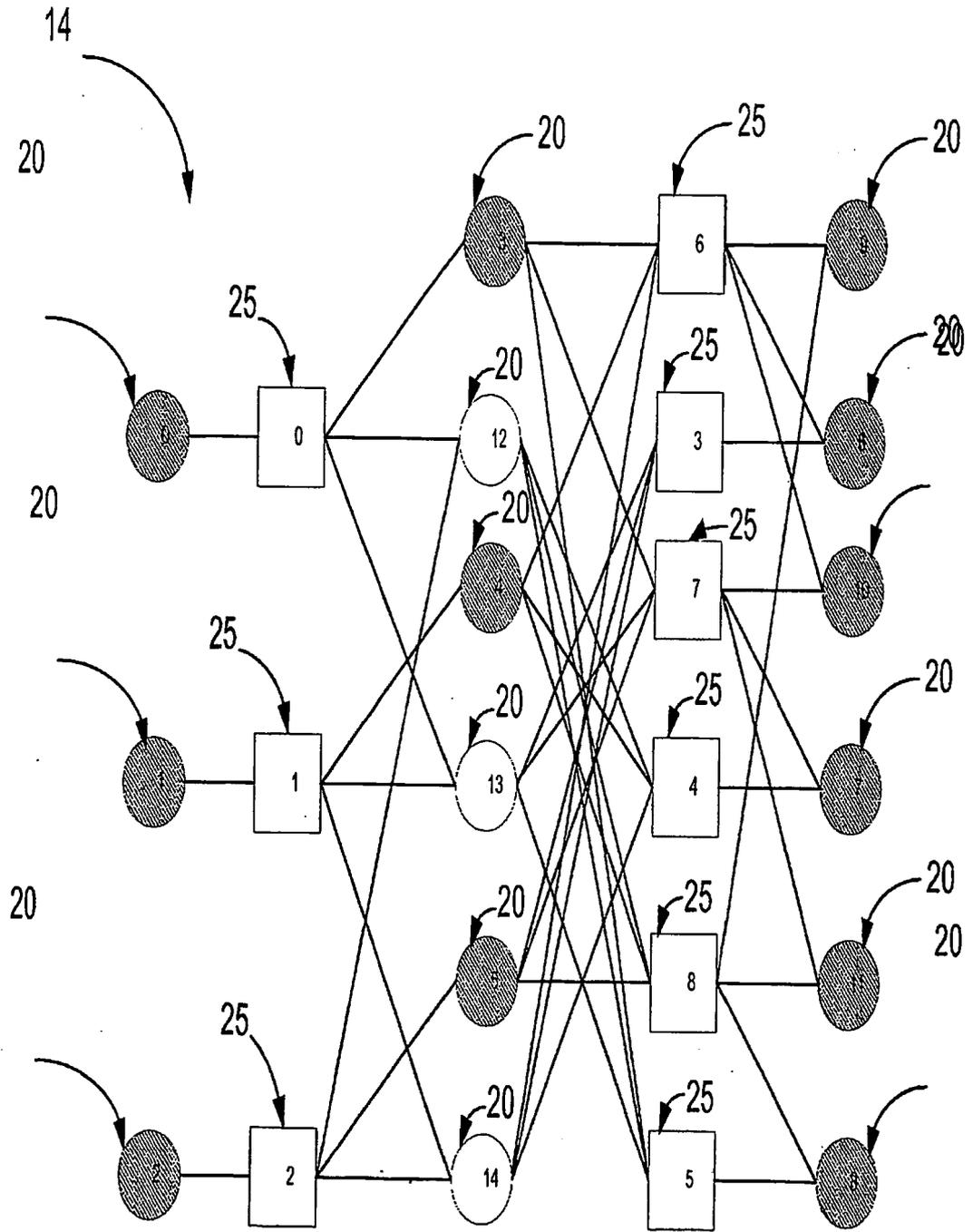


FIG. 8

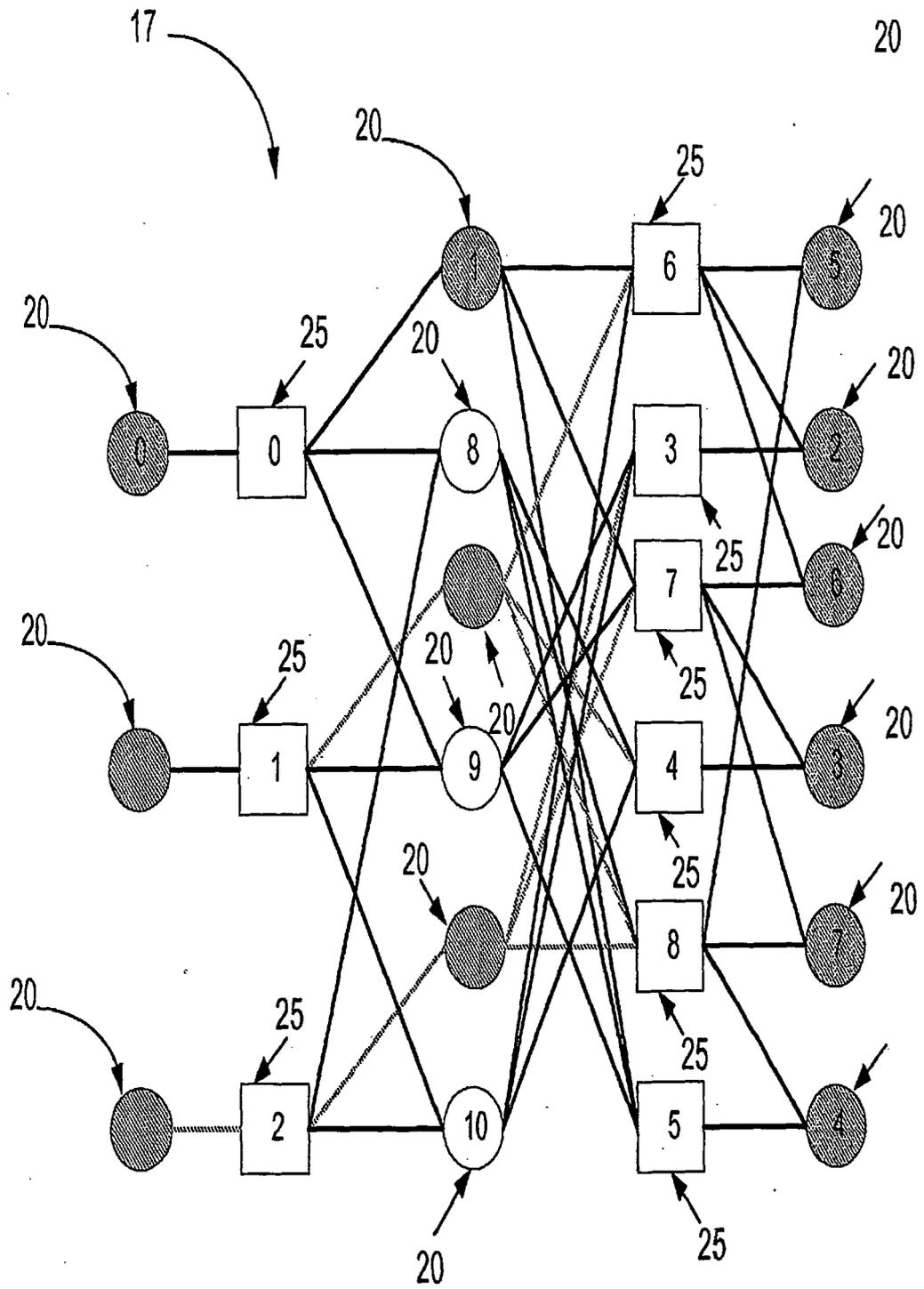


FIG. 9

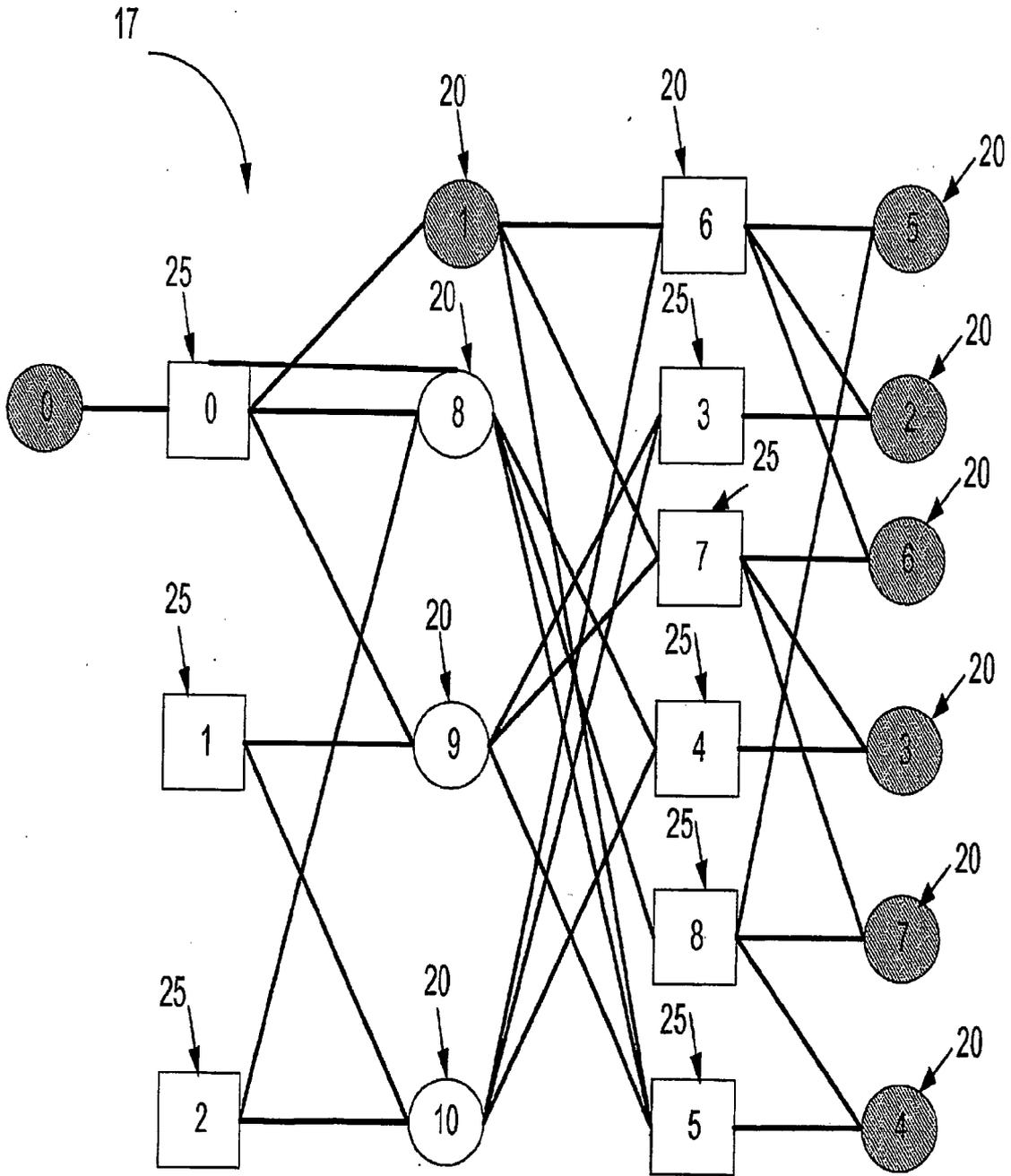
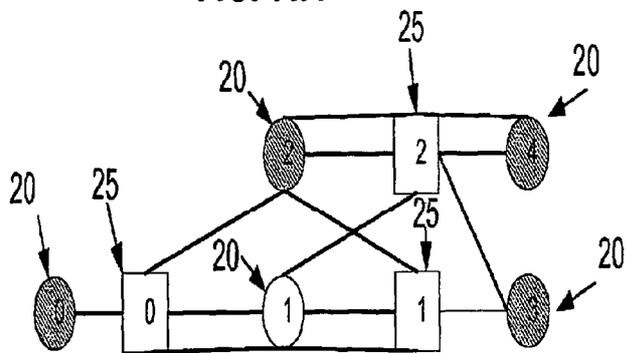


FIG. 10

Grafo base

12

FIG. 11A



tasa = 1/2

FIG. 11B

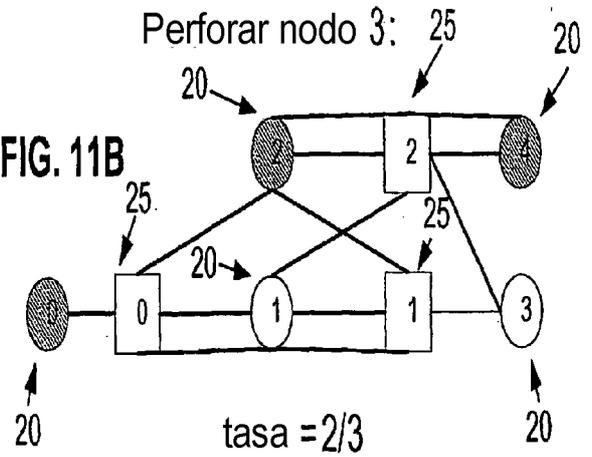
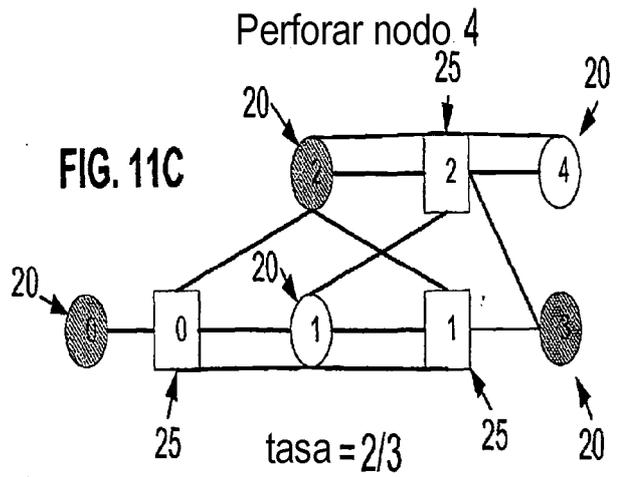


FIG. 11C



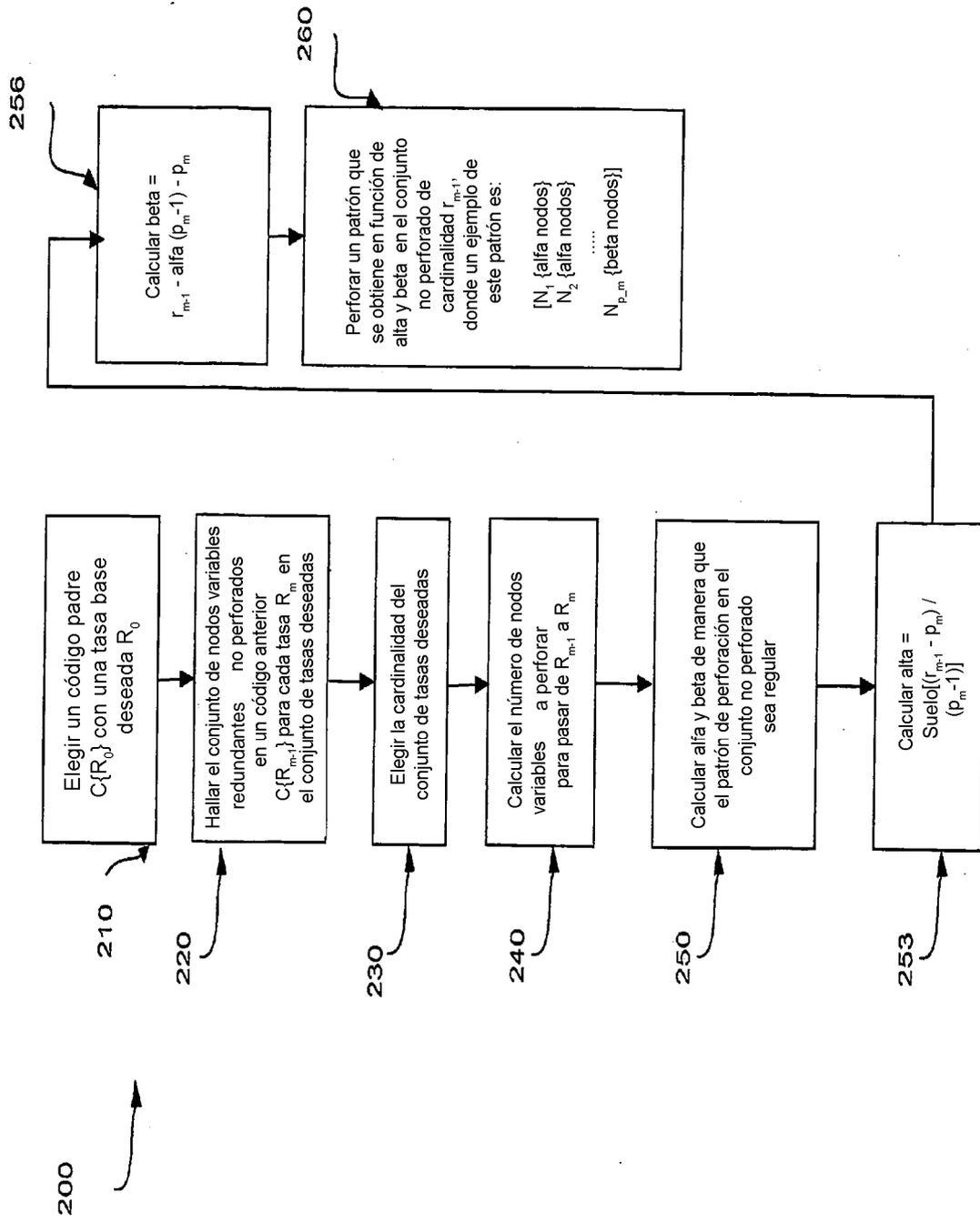


FIG. 12

$T = \frac{1}{2} Z = 6, N = 30, N_t = 6, C = 18$

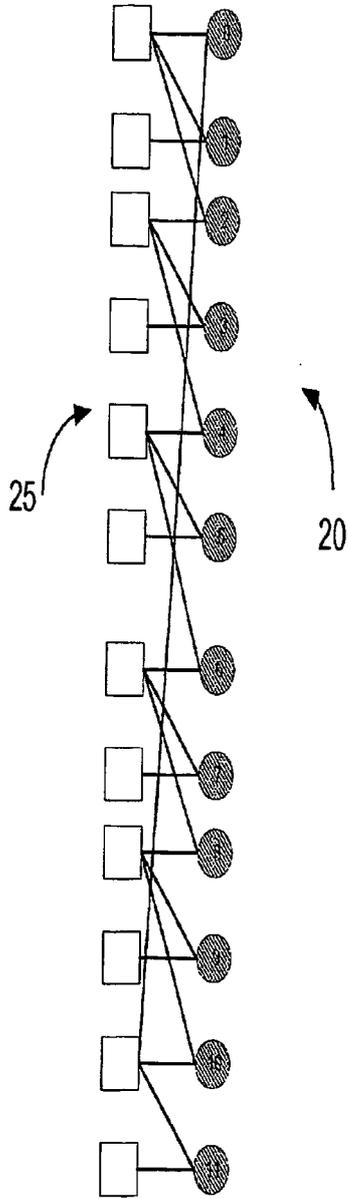
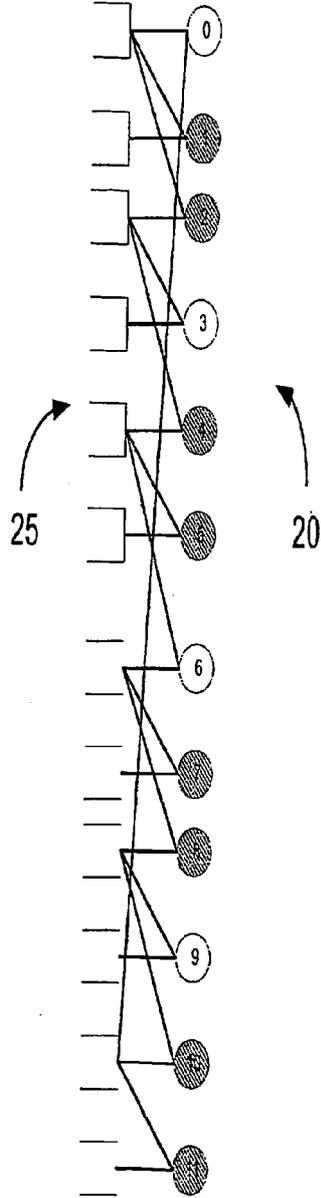


FIG. 13A

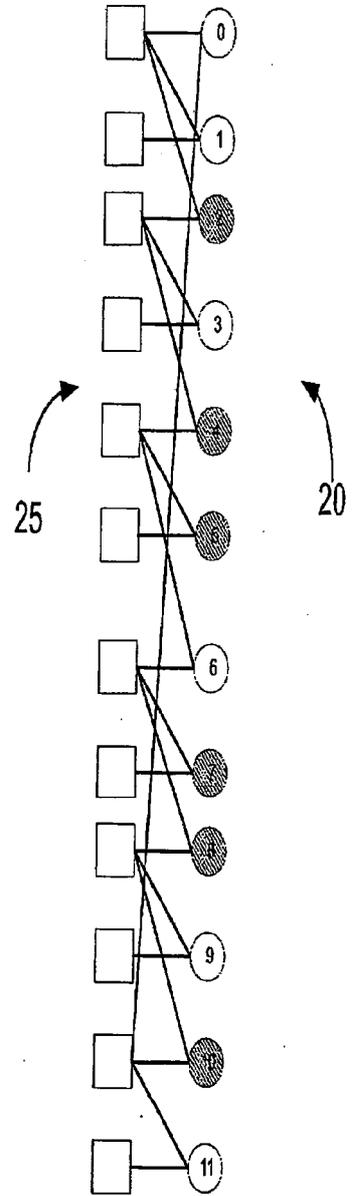
$T = 6/10,$
Perforar $10 - 6 = 4$ nodos



(Regular)

FIG. 13B

$T = 2/3,$
Perforar $12 - 10 = 2$ nodos



(Irregular)

FIG. 13C

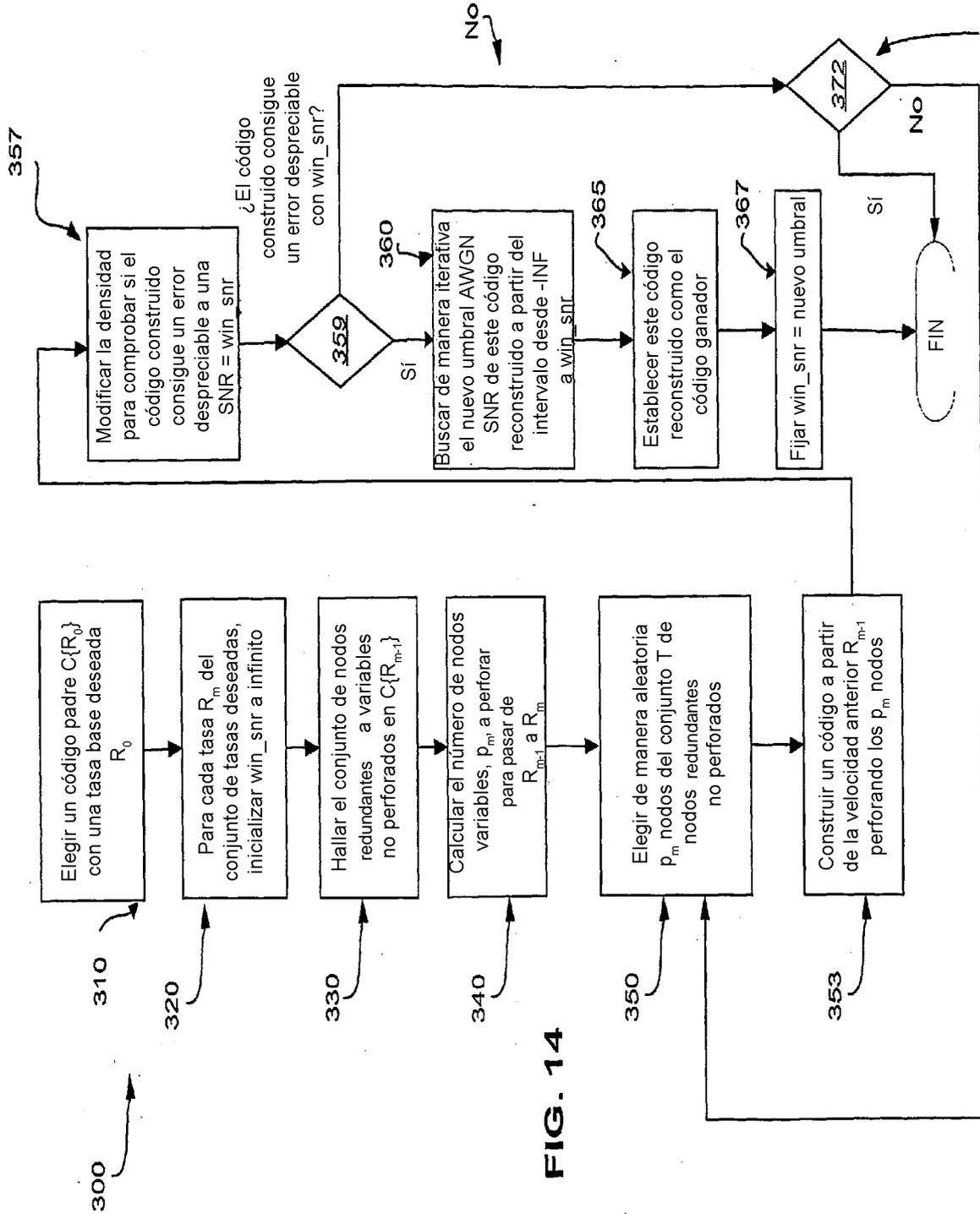


FIG. 14

Repetir el procedimiento de búsqueda aleatoria 300 a no ser que se haya alcanzado el número máximo de repeticiones

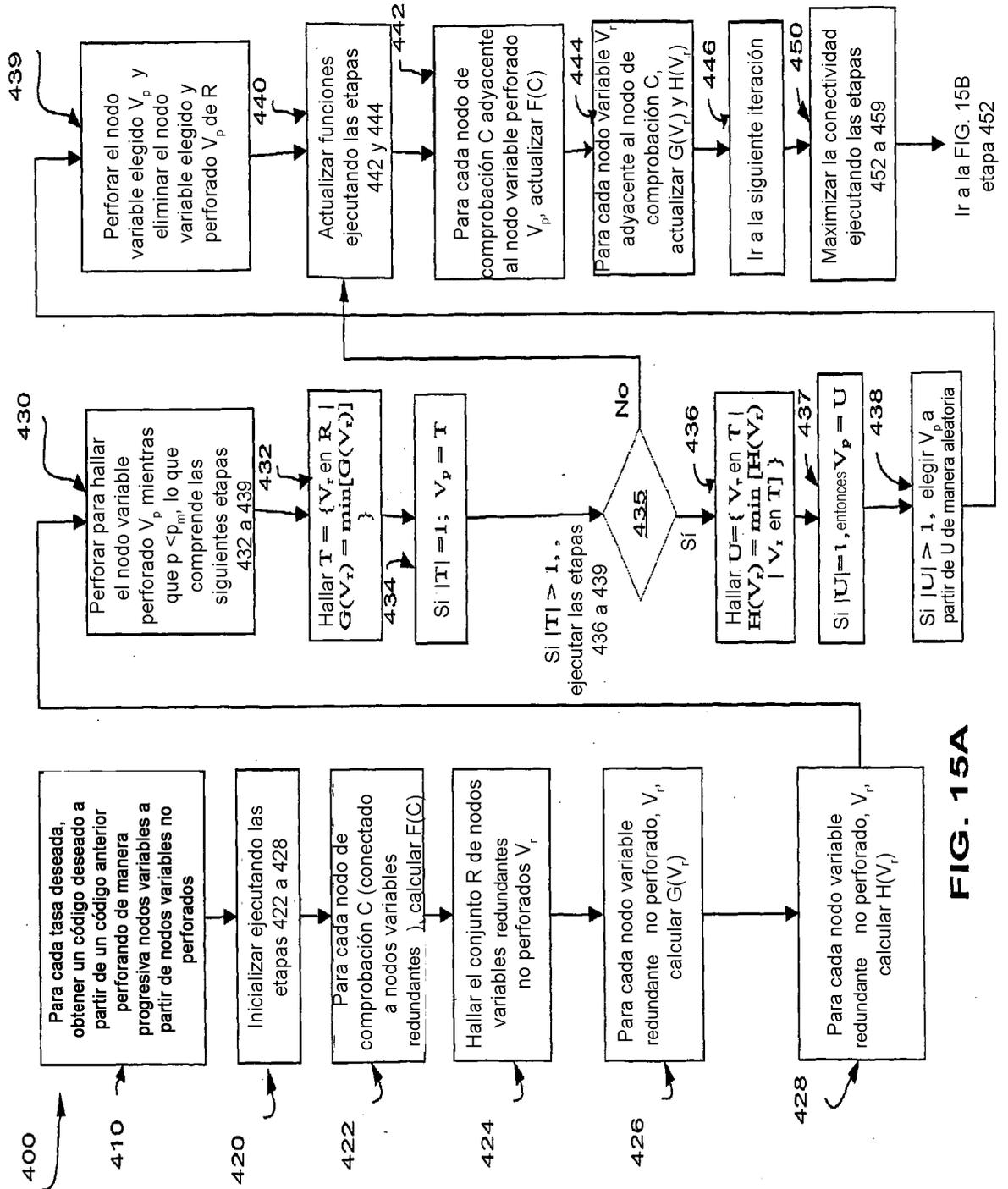


FIG. 15A

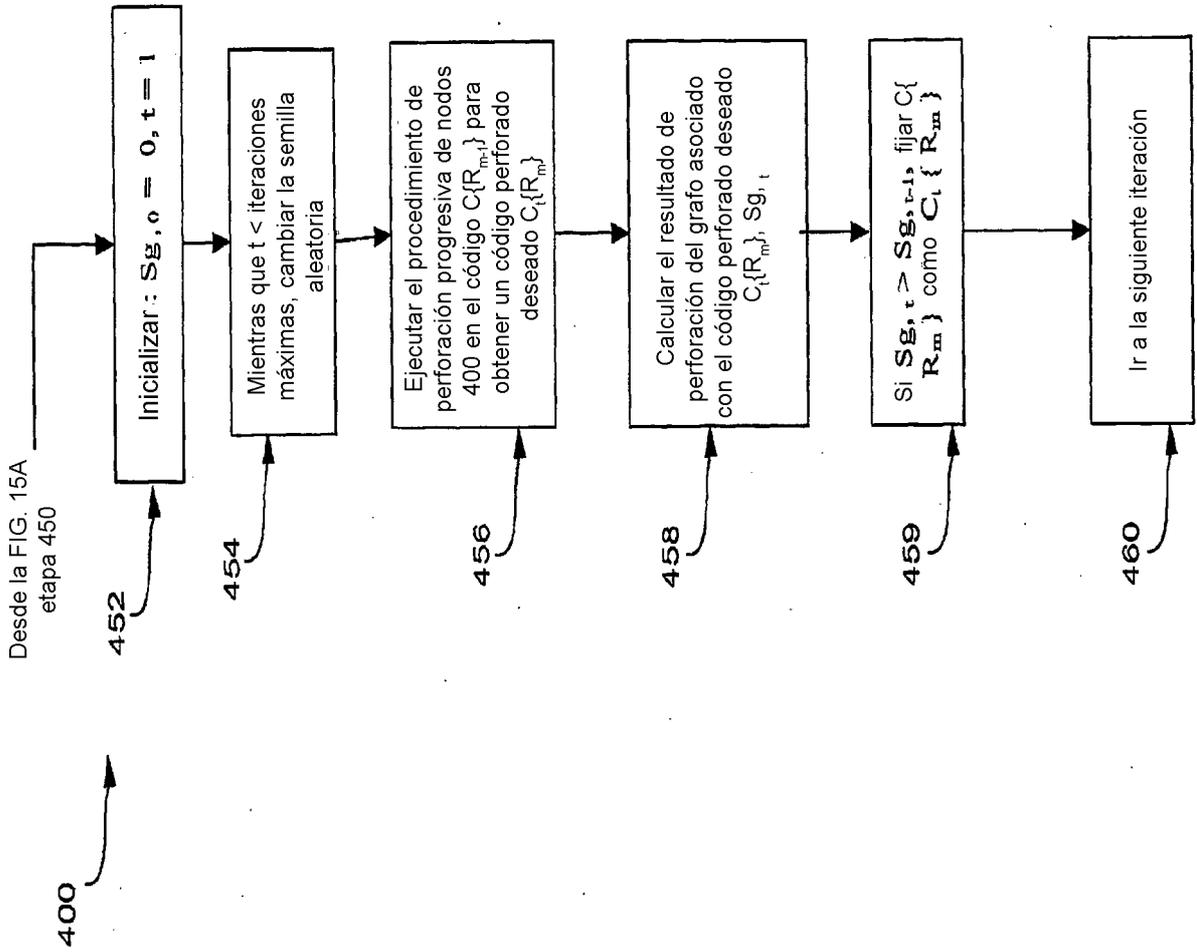


FIG. 15B

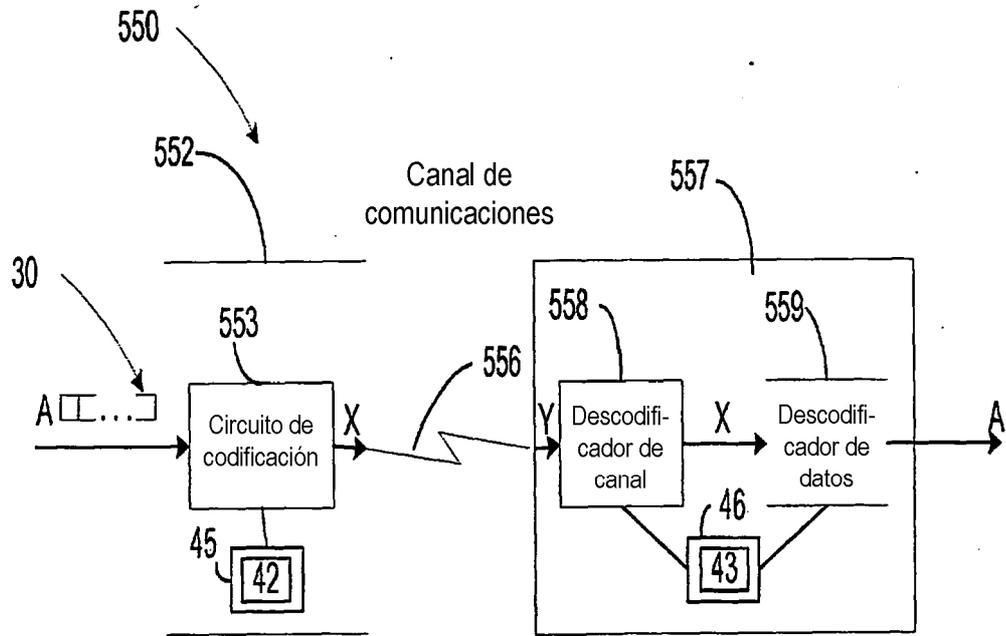


FIG. 16

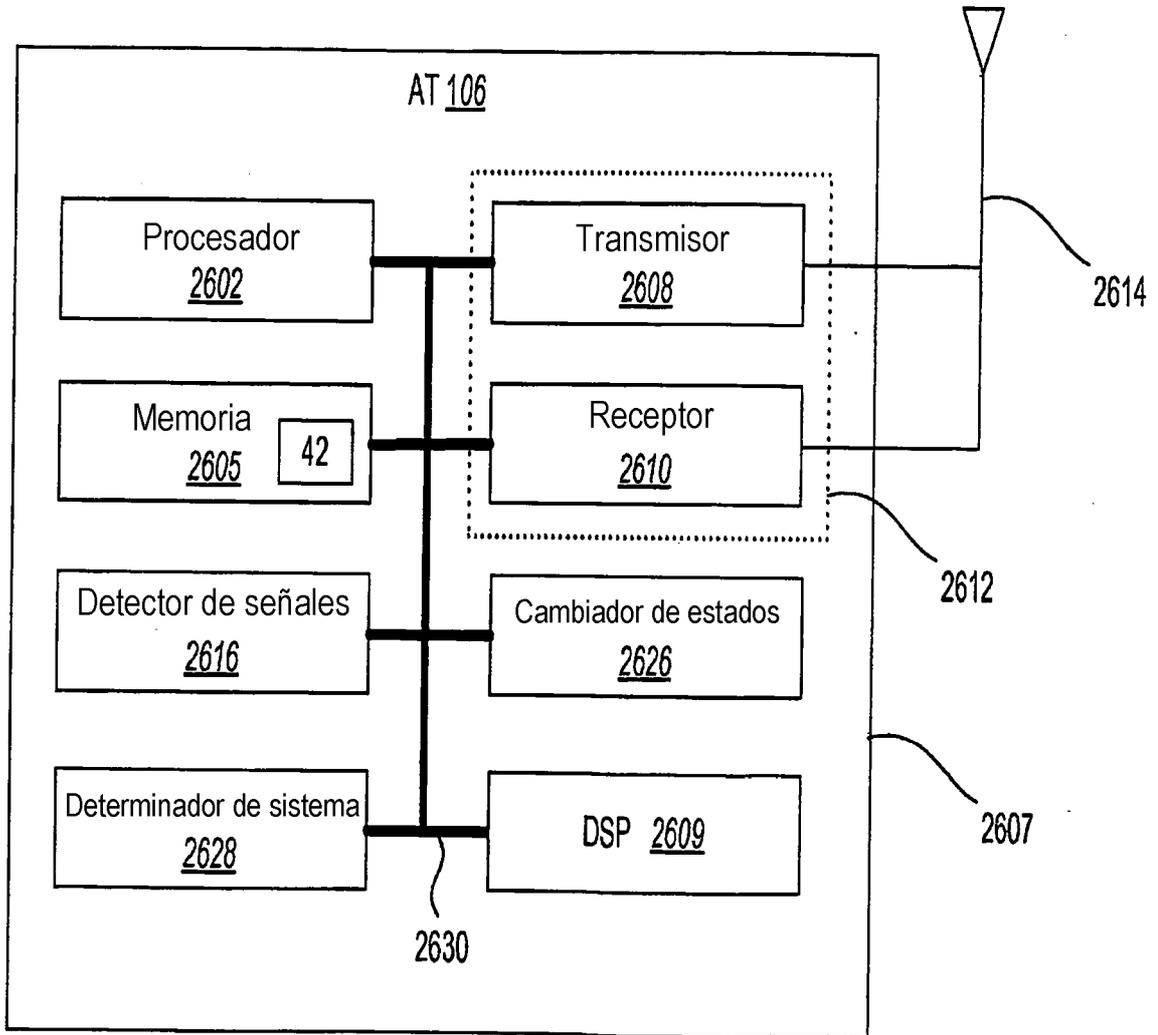


FIG. 17

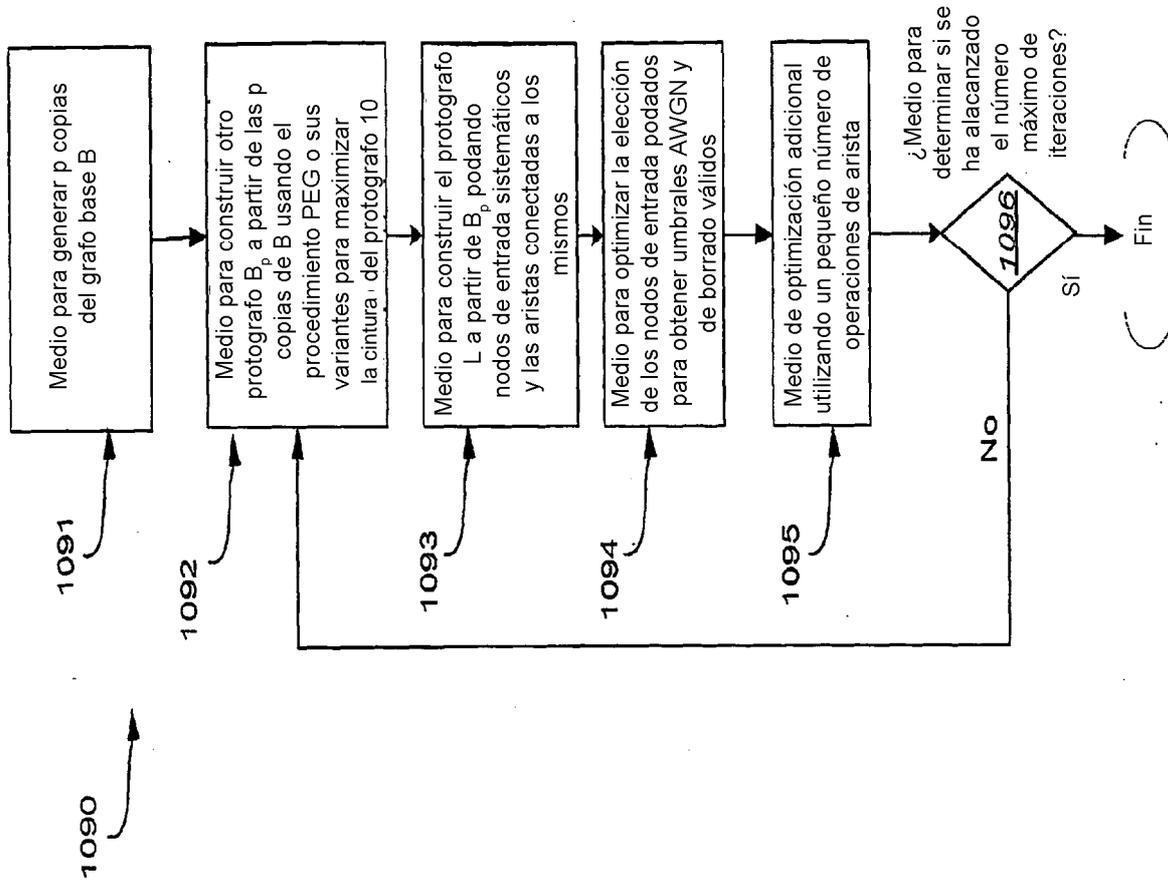


FIG. 18A

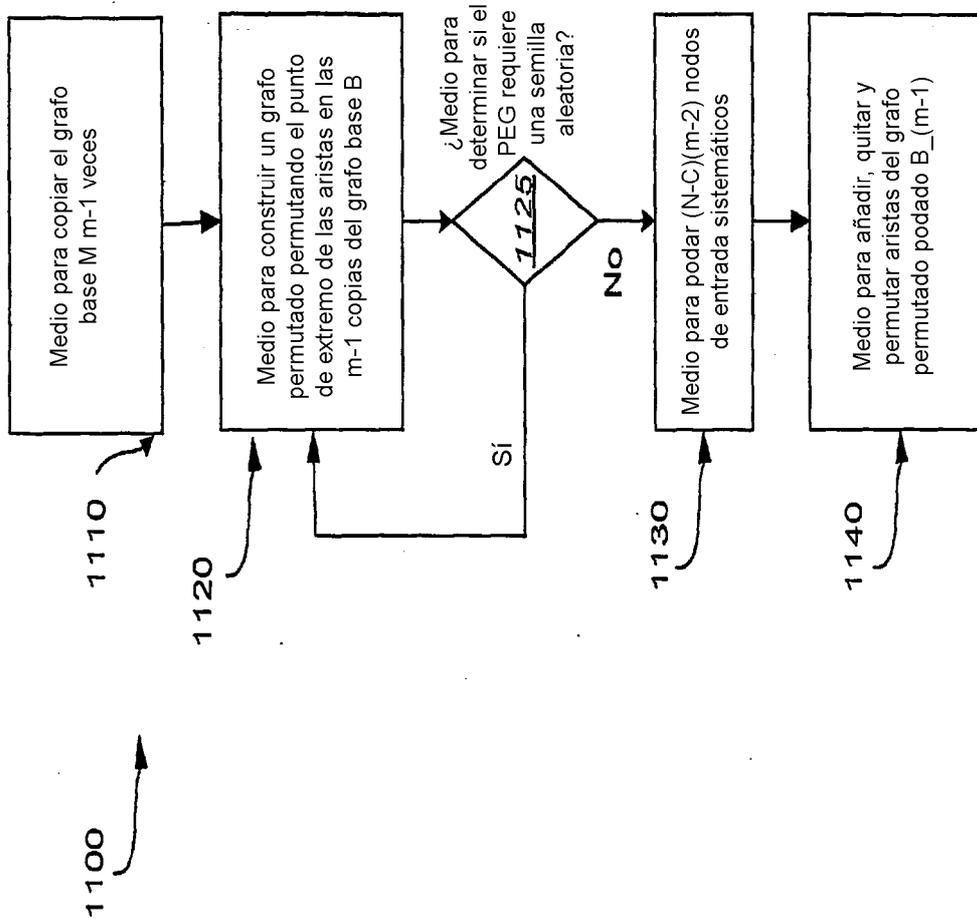


FIG. 18B

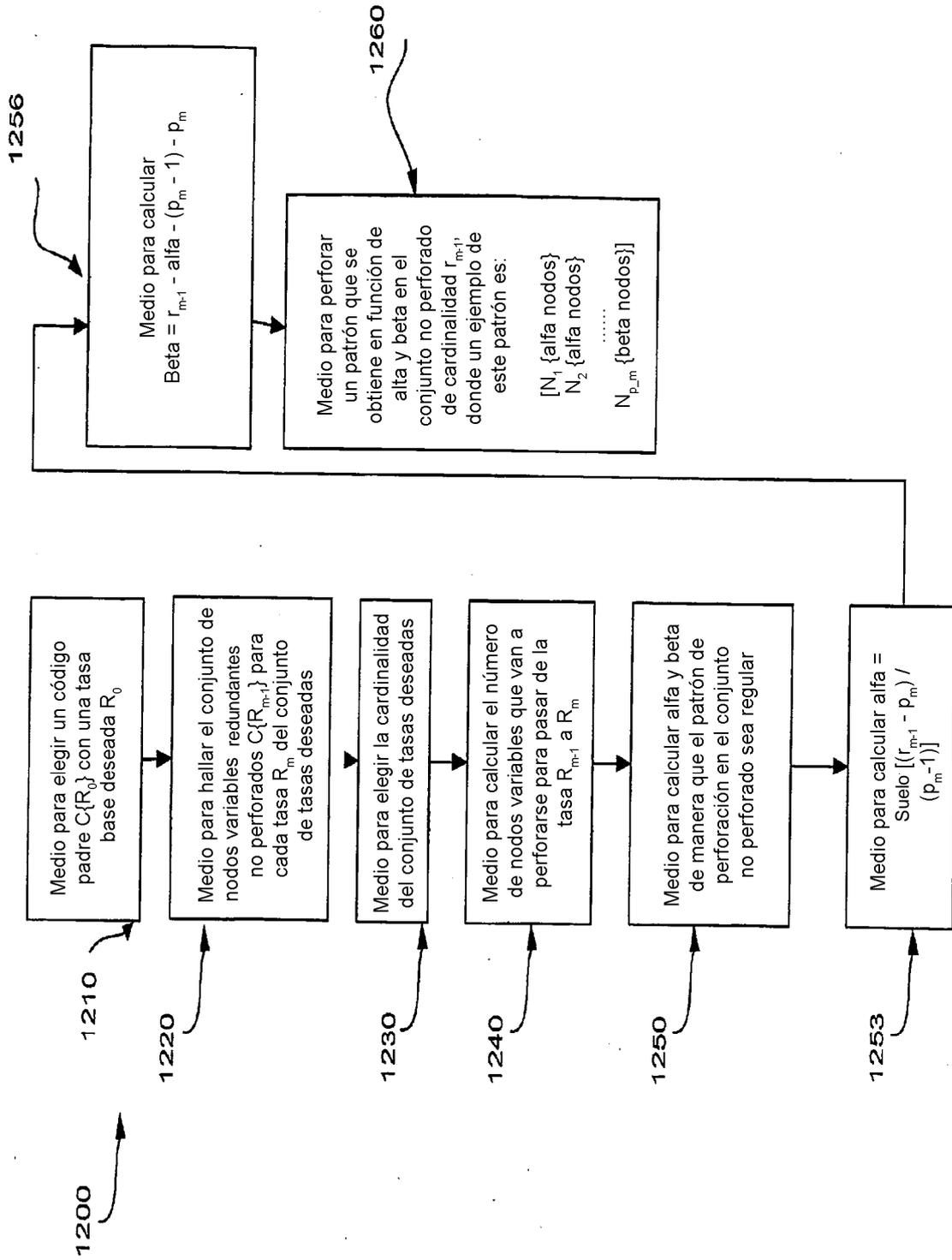


FIG. 19

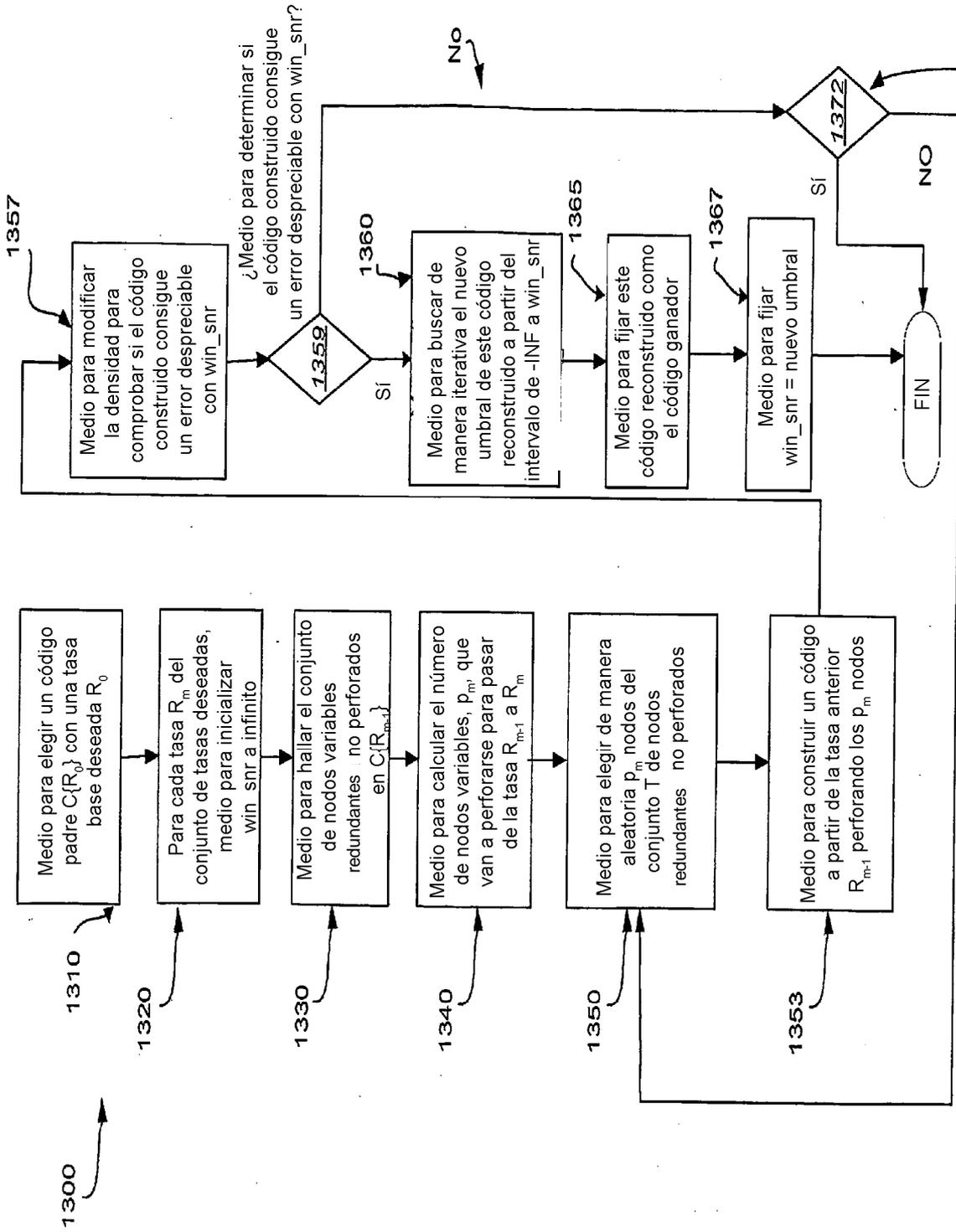


FIG. 20 Medio para repetir el procedimiento de búsqueda aleatoria 300 un número máximo de repeticiones

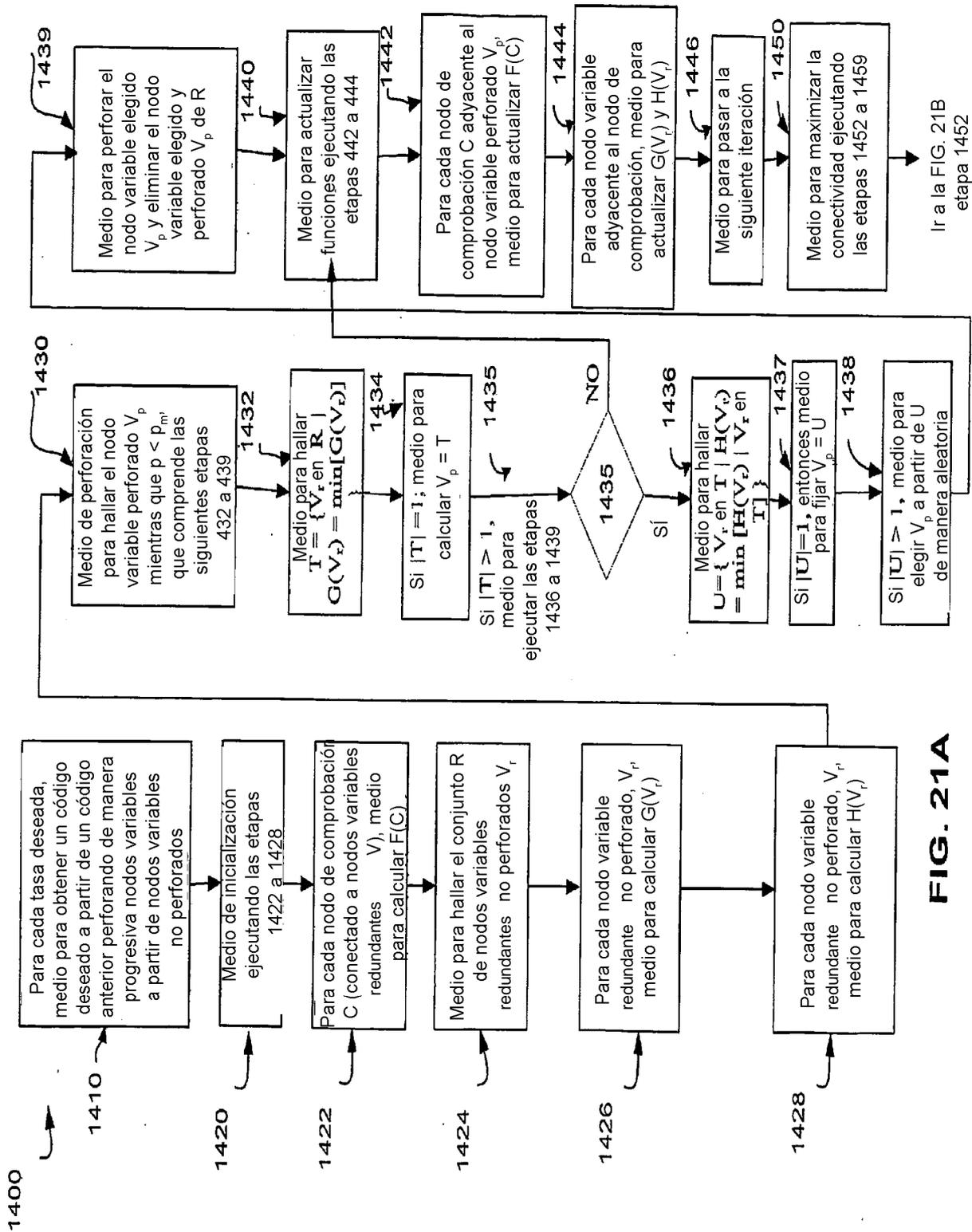


FIG. 21A

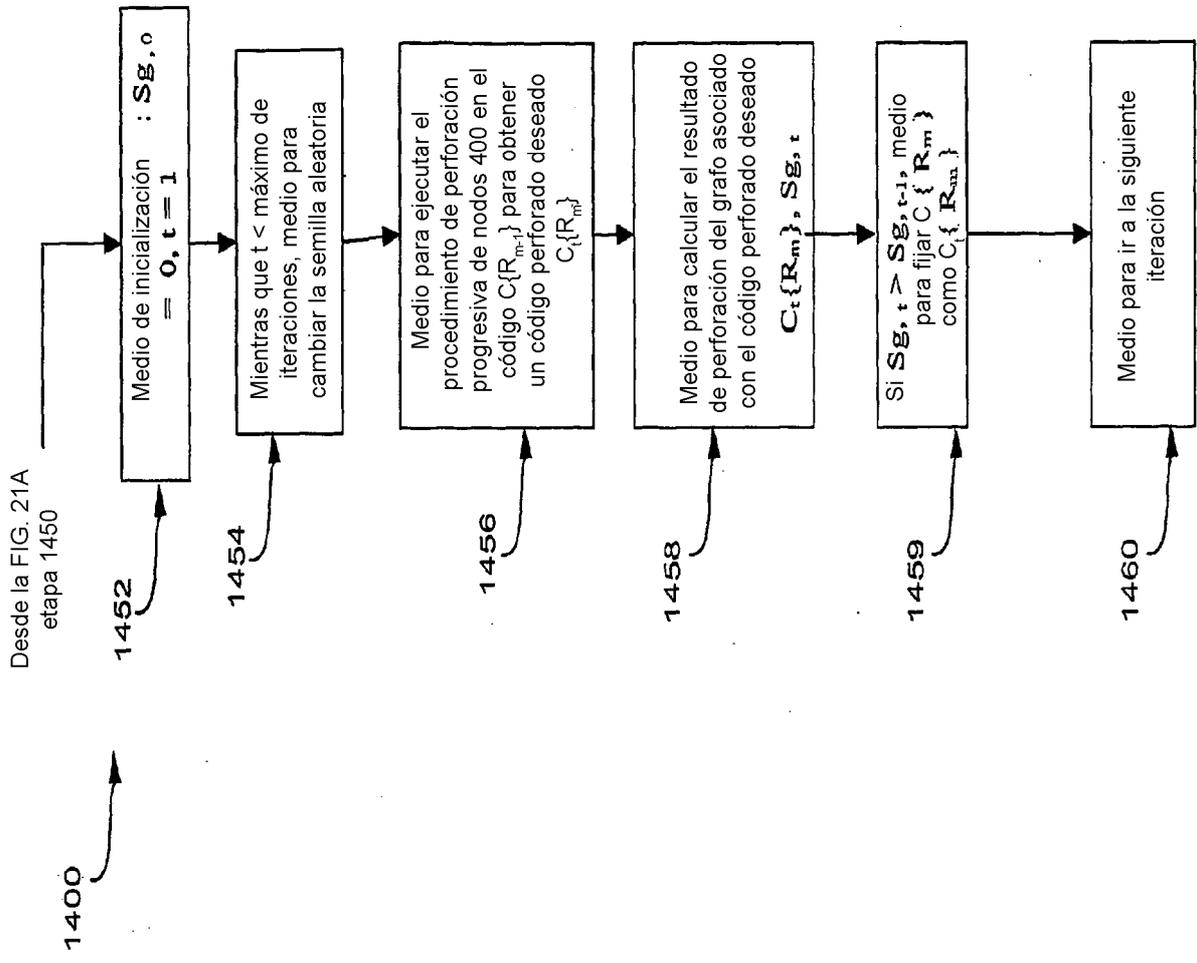


FIG. 21B