



UNIVERSIDAD NACIONAL AUTÓNOMA
DE MÉXICO

FACULTAD DE CIENCIAS

PROPIEDADES TOPOLÓGICAS Y ALGORÍTMICAS
DEL HIPERCUBO Y SUS VARIANTES

TESIS

QUE PARA OBTENER EL TÍTULO DE:

MATEMÁTICA

P R E S E N T A:

ESTHER ANAHÍ DOMINGUEZ JIMÉNEZ

DRA. MARÍA DE LUZ GASCA SOTO

Octubre, 2014





Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

Agradecimientos

Agradezco hoy y siempre a mi familia por alentarme y estar conmigo en todo momento. A mis padres, Arturo y Esther, mis abuelitas y abuelito, mis hermanos Citlali y Arturo. El ánimo, apoyo incondicional y alegría que me brindan me dá fortaleza para seguir adelante. A mi hijo Ian que es el motor de mi vida.

Un agradecimiento especial a la Dra. Luz Gasca que como directora de tesis, me orientó, ayudó e impulsó para terminar este proyecto y sobre todo por esa gran amistad que me brinda. También a la Dra. Elisa Viso, al Dr. Sergio Rajsbaum, al Dr. Bernardo Llano y la Dra. Hortensia Galeana que dedicaron gran parte de su tiempo para revisar y aportar ideas que enriquecieron este trabajo y sobre todo a mí.

A la Institución que me ha dado la oportunidad de estar aquí y crecer como persona en el ámbito académico, social y laboral.

A Marco por su ayuda incondicional desde que nos conocemos. A Paco y Paquito por sus risas y aliento; A Jorge por brindarme la oportunidad de trabajar con él y así esclarecer ciertos conceptos. A Victor que con sus risas, consejos, desvelos y ayuda me motivó para finalizar este trabajo. En general, agradezco a mis amigos porque siempre me alentaron y aportaron ideas de gran ayuda.

Índice general

Introducción.	1
1 Conceptos teóricos	3
1.1 Teoría de gráficas	3
1.2 Redes de interconexión	16
1.3 Algoritmos de comunicación	20
2 El hipercubo y sus propiedades	25
2.1 Propiedades básicas	26
2.2 Árboles generadores	32
2.3 Inmersión de otras topologías en el hipercubo	35
2.3.1 Inmersión sobre ciclos y arreglos lineales	36
2.3.2 Inmersión de mallas en el hipercubo	37
2.4 Algoritmo de enrutamiento	39
2.5 Difusión de la información	40
3 Variantes del hipercubo	43
3.1 Hipercubo torcido	43
3.2 Hipercubos localmente torcidos	44
3.3 Hipercubo cruzado	45
3.4 Hipercubo de Moebius	46
3.5 Hipercubo balanceado.	48
3.6 Gráfica n –estrella	49
3.7 (n, k) –estrella	49
3.8 (n, k) –cubos	50
4 Hipercubo aumentado	53
4.1 Propiedades topológicas básicas	54
4.2 Subgráficas de AQ_n	60
4.3 Algoritmo de enrutamiento	63
4.4 Difusión de la información	66

5 Hipercubo plegado	67
5.1 Propiedades topológicas básicas	68
5.2 Subgráficas del hipercubo plegado	71
5.3 Algoritmo de enrutamiento	77
5.4 Difusión de la información	78
Índice alfabético.	83
Conclusiones.	86
Bibliografía.	86

Índice de figuras

1.1	(a) Ejemplo de una gráfica, (b) $d(K_5) = 4$	4
1.2	Una gráfica G y una de sus subgráficas	4
1.3	Ejemplos de caminos	5
1.4	Gráfica bipartita	6
1.5	Gráfica bipartita completa $K_{4,4}$	6
1.6	Ejemplo de distancia, $d_G(u, y) = 3$	6
1.7	(a) G , (b) $G - \{w\}$, (c) $G - \{a_1, a_2, a_3\}$	7
1.8	(a) Trayectorias ajenas por aristas, (b) Trayectorias internamente disjuntas	8
1.9	Se ilustra el teorema de Menger	9
1.10	Ejemplos de árboles	9
1.11	Posiciones en un árbol binario	10
1.12	Definición de árboles binomiales	11
1.13	Ejemplo de árboles binomiales	11
1.14	Ejemplos de (a)gráficas isomorfas (b) gráficas no isomorfas	12
1.15	Trayectoria de longitud 2	12
1.16	Una gráfica transitiva en vértices y una transitiva en aristas	14
1.17	Producto cuadro de dos gráficas	14
1.18	Gráfica no panconexa	15
1.19	(a) $\ell(C) = 3$, (b) $\ell(C) = 4$, (c) $\ell(C) = 5$, (d) $\ell(C) = 6$	15
1.20	(a) malla (b) toro	17
1.21	Árbol	18
1.22	Gráfica completa K_6	18
1.23	Red estrella	19
1.24	Arreglos lineales	19
1.25	Algoritmos de comunicación	20
1.26	Toro de 6×6	22
1.27	Enrutamiento ingenuo	23
1.28	Rutas desde el nodo 00 al nodo 21	24
2.1	Ejemplo de Hipercubos	25
2.2	Separación de un 3-cubo	26
2.3	Numeración de los nodos en un 2-cubo	27
2.4	Correspondencia uno a uno entre nodos adyacentes a A y B	28

2.5	Se ilustran dos mallas de 4×4	29
2.6	Correspondencia uno a uno entre nodos rojos y azules	30
2.7	Q_4 menos las aristas cuyos extremos son rojo y azul	31
2.8	Nodos etiquetados en la gráfica del 4-cubo	31
2.9	4-cubo	32
2.10	T_1 : árbol generador, P_1 : Trayectoria de longitud uno en Q_2	32
2.11	Árboles ajenos por aristas para Q_4	33
2.12	Partición de Q_4 en cuatro hipercubos de dimensión 2	33
2.13	Árboles ajenos por aristas	34
2.14	Se ilustra la distancia de Hamming	35
2.15	Ciclo de longitud 8	36
2.16	Q_3 y un ciclo de longitud 8	37
2.17	Código Gray 2-dimensional para una malla de 8×4	38
2.18	Inmersión del 4-cubo en el toro de 4×4	39
2.19	Dos árboles binomiales generadores con $r = 0$: (a) sobre Q_3 y (b) sobre Q_4	40
3.1	Un 3-Cubo, Q_3 y un 3-Cubo Torcido, TQ_3	43
3.2	LTQ_n , para $n = 3, 4$	44
3.3	Imágenes de CQ_3 y CQ_4	46
3.4	(a) $0MQ_4$, (b) $1MQ_4$	47
3.5	Hipercubos balanceados BQ_1 y BQ_2	48
3.6	Gráficas estrella S_2 , S_3 y S_4	49
3.7	Gráfica estrella $S_{4,2}$	50
3.8	Se ilustra la gráfica de Q_2^8	51
4.1	Gráfica de AQ_3	53
4.2	Trayectorias internamente disjuntas en AQ_4	54
4.3	Gráfica de AQ_{n-1}^0 , para $n = 4$	55
4.4	Caso donde $y_1 = 0$	55
4.5	Vértices etiquetados conforme a la construcción de la demostración	56
4.6	Trayectoria de x a y	56
4.7	Trayectorias de x a y	57
4.8	$\kappa(AQ_3) = 4$	59
4.9	Se muestran 6 ciclos en AQ_3	60
4.10	Árboles binarios ajenos por aristas en AQ_3	61
4.11	Los árboles S_1 , S_2 y R_3 contenidos en AQ_3	61
4.12	Inciso (1) del teorema 4.6	63
4.13	Inciso (2. i) del teorema 4.6	64
4.14	Inciso (2. ii) del teorema 4.6	64
4.15	Trayectorias de enrutamiento (a) de 000000 a 101011 en AQ_6 y (b) de 1010010110 a 1000100011 en AQ_{10}	65
4.16	Árbol de difusión de AQ_5 con raíz en 00000	66
5.1	FQ_n para $n = 3$ y 4	67

5.2	(a): FQ_3 , (b): $FQ_3 - \{(010), (001), (100), (111)\}$	68
5.3	Cinco trayectorias internamente disjuntas en FQ_4	69
5.4	Diámetro en FQ_4	69
5.5	Árbol generador T_0 en FQ_4	75
5.6	Árbol generador T_1 en FQ_4	75
5.7	Árbol generador T_2 en FQ_4	76
5.8	Árbol generador T_3 en FQ_4	76
5.9	Árbol generador T_* en FQ_4	76
5.10	Árbol generador para un 4-cubo	78
5.11	Árbol generador de difusión para un FQ_n , (a) $n = 4$, (b) $n = 5$	80

Introducción

El desarrollo de los procesadores en paralelo ha recibido gran atención en los últimos años con el objetivo de obtener computadoras de alto desempeño. Este campo genera una gran cantidad de investigación en áreas de las Ciencias de la Computación y la Teoría de Gráficas. Un área importante de investigación en este campo es el estudio de las redes de interconexión, debido a que los sistemas de cómputo se ven afectados por la sobrecarga de comunicación. En esencia, el problema básico a resolver es: dado un conjunto de procesadores o nodos de una gráfica, definimos un conjunto de enlaces o aristas en la gráfica tal que los nodos se puedan comunicar entre ellos a un mínimo costo. Tal simplificación del problema no parece hacerle justicia a este campo de investigación; sin embargo, tanto su dominio de soluciones como las aplicaciones han generado una amplia investigación del tema en las últimas décadas.

Dado un conjunto de procesadores, o vértices, en una red de interconexión, podemos conectarlos todos contra todos mediante enlaces, o aristas; como resultado tendremos una gráfica completa. Cada procesador podrá enviar y recibir datos directamente pues dispondrá de $n - 1$ vecinos para ello. Aunque parece conveniente, este tipo de red es costosa y poco realista sobre todo para valores grandes de n ; por lo cual no resulta factible en la práctica. Como consecuencia se buscan topologías que permitan al conjunto de nodos, por medio de las aristas, comunicarse a costo mínimo.

El diseño de una red de interconexión es una parte integral de cualquier sistema. La simetría, un grado pequeño, un diámetro bajo, la tolerancia a fallos, la conexidad y los algoritmos de enrutamiento son propiedades deseables de una topología eficiente. Por ejemplo: el grado de una gráfica es una medida del costo de la red; el diámetro es una medida de la demora de la comunicación entre los nodos; la transitividad en vértices y en aristas permite minimizar el congestionamiento del envío de mensajes enrutados en la gráfica. Por otra parte, las trayectorias y ciclos son fundamentales para el diseño de algoritmos simples con bajo costo de comunicación en cualquier red de interconexión: La tolerancia a fallos junto con algoritmos simples de enrutamiento (*routing*) generará trayectorias de longitud mínima entre cualesquiera dos vértices, permitiendo que el desempeño en términos de retardo de comunicación sea óptimo.

La red de interconexión más popular es el hipercubo. En años recientes han surgido diversas variantes del hipercubo las cuales poseen atractivas propiedades topológicas.

- Presentaremos algunas de estas variantes para ilustrar la gran diversidad de las mismas.

- Describiremos con detalle las propiedades topológicas y algorítmicas tanto del hipercubo como de dos de sus variantes: el hipercubo Aumentado y el hipercubo Torcido.
- La mayoría de dichas propiedades serán demostradas formalmente e ilustradas para su comprensión. Compararemos las diversas propiedades entre el hipercubo y las variantes antes mencionadas.
- Finalmente, revisaremos algoritmos básicos de enrutamiento y distribución de la información (*broadcasting*) para el hipercubo y las variantes presentadas.

En este trabajo se mostrará que las redes de interconexión hipercubo, hipercubo Aumentado e hipercubo Plegado poseen excelentes propiedades topológicas que las hacen una red atractiva. Todo el desarrollo en este documento explotará la estructura que como gráfica tiene el hipercubo y sus variantes.

Este trabajo está organizado de la siguiente forma:

El primer capítulo contiene conceptos básicos y algunas propiedades de teoría de gráficas y redes de interconexión que utilizaremos durante el desarrollo de este documento, así como dos de los algoritmos generales de comunicación: enrutamiento y difusión de la información.

En el segundo capítulo definimos al hipercubo Q_n desde el punto de vista de teoría de gráficas, junto con sus propiedades topológicas básicas como cardinalidad, conexidad, transitividad, regularidad, entre otras. Realizamos la inmersión sobre algunas redes de interconexión tales como mallas, anillos y arreglos lineales. Los algoritmos de enrutamiento y difusión de la información definidos de manera general en el capítulo anterior se aplicarán en el hipercubo Q_n .

Posteriormente, en el siguiente capítulo, citamos algunas de sus variantes: hipercubo Torcido, hipercubo Localmente Torcido, hipercubo Cruzado, hipercubo de Moebius, hipercubo Balanceado, n -estrella, (n, k) -estrella, (n, k) -cubos, Aumentado y Plegado, revisando propiedades topológicas de cada una de estas redes.

Finalmente, en los capítulos 4 y 5 estudiamos detalladamente a los hipercubos Aumentado y Plegado, respectivamente. En las dos redes analizamos sus propiedades topológicas como conexidad y simetría, árboles como subgráficas y aplicaremos el algoritmo de enrutamiento y difusión de la información en cada una de éstas.

Concluimos con la comparación entre las redes del hipercubo Q_n , el hipercubo Aumentado AQ_n y el hipercubo Plegado FQ_n . Observamos que el diámetro es un parámetro importante ya que entre menor sea, el envío de un mensaje a través de toda la red llegará más rápido a su destino; de esta manera podremos decir cual de estas tres redes es la óptima.

Capítulo 1

Conceptos teóricos

En este capítulo presentaremos los conceptos teóricos que utilizaremos a lo largo de este trabajo. En la primera sección estudiamos a la teoría de gráficas; en la segunda sección a las gráficas de Cayley; en la tercera sección, redes de interconexión y en la última sección se revisan algoritmos generales de comunicación.

1.1. Teoría de gráficas

En esta sección definiremos los conceptos básicos de teoría de gráficas, así como algunas propiedades relacionadas con conexidad y simetría que usaremos durante el desarrollo de este documento.

Definiciones básicas

Definición 1.1. [13] **Gráfica.** Una gráfica $G = (V, A)$ es un conjunto finito no vacío de vértices, $V(G)$, junto con un conjunto de aristas, $A(G)$, tal que

$$A(G) \subseteq \{\{u, v\} \mid u, v \in V(G)\}$$

Es decir, $A(G)$ es un conjunto de pares no ordenados de vértices. Si la arista $a \in A(G)$, con $a = \{u, v\}$, se dice que u y v son **adyacentes**. También se dice que u y v son vecinos.

Si el número de vértices de la gráfica es p , $|V(G)| = p$, y el número de aristas es q , $|A(G)| = q$, se dice que G tiene **orden** p y **tamaño** q .

La figura 1.1 (a) muestra el ejemplo de una gráfica donde el conjunto de vértices es: $V(G) = \{1, 2, 3, 4, 5\}$ y $A(G) = \{\{1, 2\}, \{1, 5\}, \{2, 3\}, \{2, 4\}, \{2, 5\}, \{3, 5\}, \{3, 4\}\}$.

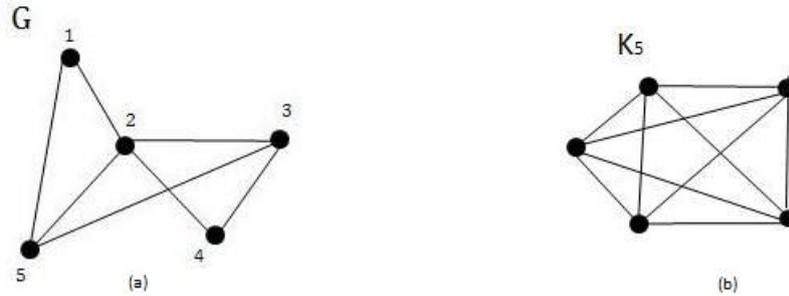


Figura 1.1: (a) Ejemplo de una gráfica, (b) $d(K_5) = 4$

Definición 1.2. [13] **Grado de un vértice.** Al número de aristas que inciden en un vértice v de una gráfica G , se le llama el grado de v en G y se denota por $d(v)$. El número $\delta(G) := \min\{d(v) | v \in V\}$ es el **grado mínimo** de G y $\Delta(G) := \max\{d(v) | v \in V\}$ es el **grado máximo**. Si todo vértice de una gráfica G tiene el mismo grado r , decimos que G es regular de grado r o r -regular.

Una gráfica es completa si cualesquiera dos de sus vértices son adyacentes. Una gráfica completa de orden p es $(p - 1)$ -regular y se denota por K_p .

La figura 1.1 (b) indica el grado de cada vértice de la gráfica K_5 , los cuales tienen valor 4. Decimos también que un vértice v es un **vértice terminal** si su grado es 1, para algún $v \in V(G)$. En la gráfica G' de la figura 1.2 el vértice v_7 es un vértice terminal.

Definición 1.3. [4] **Subgráfica.** Sea G una gráfica. Una gráfica H es una subgráfica de G si $V(H) \subseteq V(G)$ y $A(H) \subseteq A(G)$. Si una gráfica F es isomorfa a una subgráfica H de G , entonces F también es subgráfica de G .

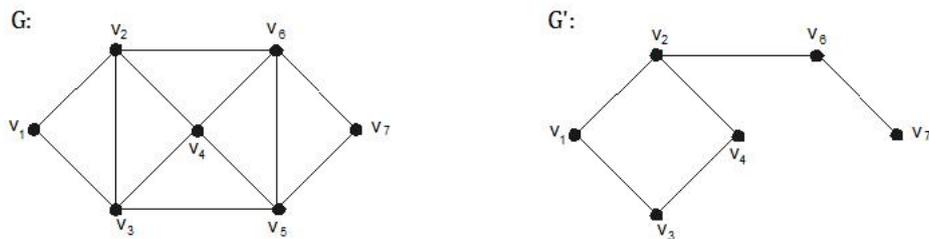


Figura 1.2: Una gráfica G y una de sus subgráficas

Por ejemplo, consideremos la gráfica G que se muestra en la figura 1.2. La gráfica G' es una subgráfica de G cuyo conjunto de vértices es $\{v_1, v_2, v_3, v_4, v_6, v_7\}$.

Camino, ciclos y trayectorias.

Un **camino** en una gráfica $G = (V, A)$ es una sucesión alternada de vértices y aristas $v_0, e_1, v_1, e_2, \dots, v_{k-1}, e_k, v_k$. Un camino puede denotarse también como una sucesión

finita de vértices $v_0, v_1, v_2, \dots, v_k$, tal que $\{v_{i-1}, v_i\}$, $1 \leq i \leq k$, es una arista en la gráfica G . Este camino usualmente es llamado un v_0v_k -camino refiriéndose a v_0 y v_k como los **vértices terminales** en la subgráfica dada por el camino. Notemos que en un camino, las aristas y los vértices pueden aparecer más de una vez. Un camino es abierto si sus vértices finales son distintos; de otra manera, se dice que es cerrado.

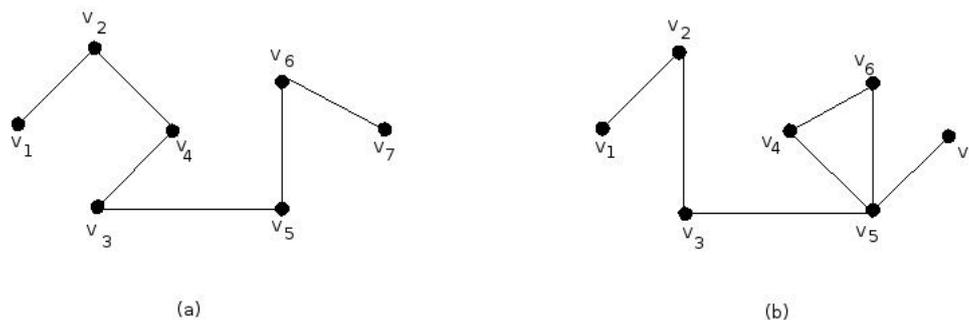


Figura 1.3: Ejemplos de caminos

Las gráficas de la figura 1.3 muestran dos ejemplos de caminos distintos para la gráfica G de la figura 1.2; en (a) se tiene el v_1v_7 -camino: $v_1, v_2, v_4, v_3, v_5, v_6, v_7$; y en (b) se tiene otro v_1v_7 -camino: $v_1, v_2, v_3, v_5, v_6, v_4, v_5, v_7$.

Un camino es un **paseo** si todas sus aristas son distintas. Un paseo es abierto si sus vértices finales son distintos; de otra manera, es un paseo cerrado.

En la figura 1.2, el camino $v_1, v_2, v_6, v_7, v_5, v_2, v_4$ es un paseo abierto en G , mientras que el camino $v_1, v_2, v_6, v_4, v_2, v_3, v_1$ es un paseo cerrado en G .

Una **trayectoria o ruta** es un camino en el cual todos sus vértices son distintos.

Denotaremos por xy -trayectoria a la trayectoria que va del vértice x al y .

El camino de la figura 1.3 (a) es una trayectoria, mientras que el de la figura 1.3 (b) es un paseo.

Un **ciclo** es un camino v_0, v_1, \dots, v_n , en el cual $n \geq 2$, $v_0 = v_n$, y los n vértices v_1, v_2, \dots, v_n son distintos.

Un paseo cerrado se llama **circuito**. Consecuentemente, todo ciclo es un circuito pero un circuito no necesariamente es un ciclo.

El número de aristas que hay en una trayectoria es llamada **longitud de la trayectoria** y la denotaremos por $\ell(T)$. Una trayectoria es par si el número de aristas que hay en ella es par; de otra manera se dice que es impar.

Un ciclo que contiene a todos los vértices de una gráfica G se llama **ciclo hamiltoniano**. En un ciclo hamiltoniano no se repiten vértices ni aristas.

Una gráfica G es **bipartita** si $V(G)$ es la unión de dos conjuntos disjuntos, digamos V_1 y V_2 , tal que toda arista de G une a un vértice de V_1 con un vértice de V_2 .

Mostramos un ejemplo de gráfica bipartita en la figura 1.4. En el conjunto V_1 están los vértices de color rosa y al conjunto V_2 pertenecen los vértices de color azul.

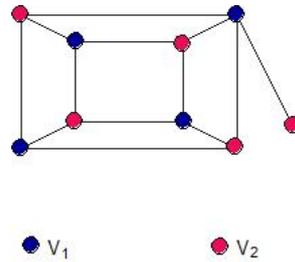
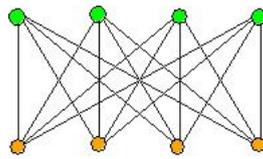


Figura 1.4: Gráfica bipartita

La gráfica **bipartita completa** $K_{m,n}$, es una gráfica bipartita con m vértices en V_1 y n vértices en V_2 tal que todo vértice en V_1 es adyacente a todo vértice de V_2 .

La figura 1.5 muestra la gráfica $K_{4,4}$ en la que se exhibe claramente la bipartición de sus vértices y todas las adyacencias permitidas para que siga siendo bipartita.

Figura 1.5: Gráfica bipartita completa $K_{4,4}$

Teorema 1.1. [13] Una gráfica G es bipartita si y sólo si todos sus ciclos son pares

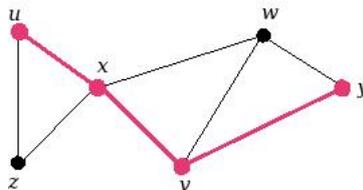
Distancias

Definición 1.4. Distancia. Dados dos vértices $u, v \in V(G)$, definimos la distancia de u a v como

$$d_G(u, v) = \min\{\ell(T) \mid T \text{ } uv \text{ - trayectoria en } G\}.$$

Esta distancia representa la longitud de la ruta más corta (mínimo número de aristas) entre u y v .

La figura 1.6 muestra un ejemplo donde la distancia entre u y y en G es 3. Esta trayectoria se exhibe en color rosa.

Figura 1.6: Ejemplo de distancia, $d_G(u, y) = 3$

Definición 1.5. Excentricidad. La excentricidad $e(v)$ de un vértice $v \in V(G)$ es la distancia máxima entre dos vértices u y v , para $u \in V(G)$; esto es,

$$e(v) = \max\{d_G(v, u) : u \in V(G) \text{ y } u \neq v\}.$$

La excentricidad de un vértice v representa la mayor de las rutas más cortas entre v y cualquier otro vértice. En la gráfica de la figura 1.6, $e(u) = 3$, $e(v) = 2$, $e(w) = 2$, $e(x) = 2$, $e(y) = 3$, $e(z) = 3$.

Definición 1.6. Radio de una gráfica. El radio de una gráfica, $r(G)$, es el valor mínimo de la excentricidad de los vértices; $r(G) = \min\{e(v) : v \in V(G)\}$.

Por ejemplo, en la gráfica de la figura 1.6, $r(G) = 2$.

Definición 1.7. Diámetro de una gráfica. El diámetro de una gráfica, denotado por $diam(G)$, es el valor máximo de la excentricidad de los vértices de G ; esto es:

$$diam(G) = \max\{e(v) : v \in V(G)\}.$$

Es decir, el diámetro de G es la máxima distancia entre dos vértices de G . En la gráfica de la figura 1.6, $diam(G) = 3$. Se tiene el siguiente resultado que relaciona estos últimos conceptos:

Teorema 1.2. [15] Sea G una gráfica, entonces:

$$r(G) \leq diam(G) \leq 2r(G).$$

Conexidad

Definición 1.8. Conexidad. Una gráfica G es conexa si para cada par de vértices existe una trayectoria entre ellos. Una gráfica es desconexa si no es conexa.

La figura 1.7 (a) muestra un ejemplo de gráfica conexa, mientras que en el inciso (b) de la misma figura la gráfica es no conexa.

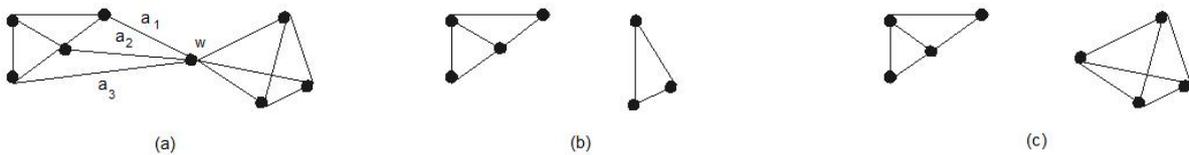


Figura 1.7: (a) G , (b) $G - \{w\}$, (c) $G - \{a_1, a_2, a_3\}$

Definición 1.9. Conexidad por vértices. La conexidad por vértices o conexidad, $\kappa(G)$, de una gráfica G , es el mínimo número de vértices cuya eliminación da como resultado una gráfica desconexa o la gráfica trivial, K_1 .

En otras palabras, la conexidad $\kappa(G)$ representa el mínimo número de vértices que se necesita quitar de G para que ésta se desconecte. En la figura 1.7, la conexidad de G , presentada en (a), es 1 pues basta quitar el vértice w para desconectarla, lo cual se ilustra en la figura 1.7 (b).

Definición 1.10. Conexidad por aristas. La conexidad por aristas (o lineal), $\lambda(G)$, de una gráfica G , es el mínimo número de aristas cuya eliminación da como resultado una gráfica no conexa.

En otras palabras, la conexidad $\lambda(G)$ representa el mínimo número de aristas que se necesita quitar de G para que ésta se desconecte. Podemos observar que en la figura 1.7 (c) la conexidad lineal de G es 3, pues al eliminar mínimo tres aristas la gráfica deja de ser conexa.

Definición 1.11. Trayectorias ajenas por aristas. Dos uv -trayectorias Q_1 y Q_2 de una gráfica G son ajenas por aristas si Q_1 y Q_2 no tienen aristas en común.

Por ejemplo en la figura 1.8 (a) las trayectorias que se muestran no comparten aristas.

Definición 1.12. Trayectorias internamente disjuntas. Dos uv -trayectorias Q_1 y Q_2 de una gráfica G son internamente disjuntas si $V(Q_1) \cap V(Q_2) = \{u, v\}$.

Un ejemplo se muestra en la figura 1.8 (b).

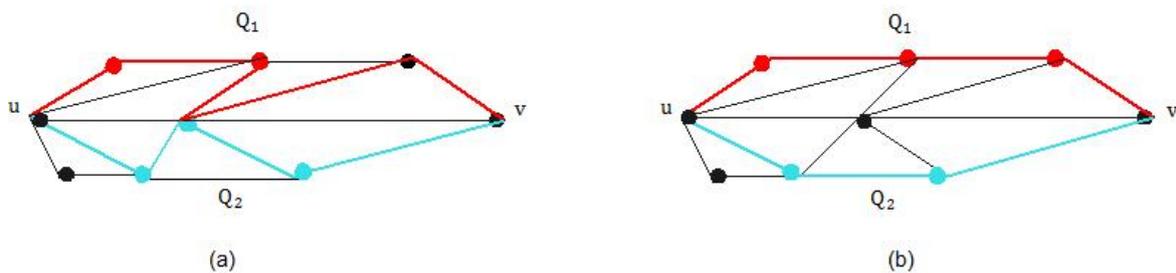


Figura 1.8: (a) Trayectorias ajenas por aristas, (b) Trayectorias internamente disjuntas

Teorema 1.3. [5] **Teorema de Menger.**

Sean u y v dos vértices no adyacentes en una gráfica G . Entonces el mínimo número de vértices que separan u y v es igual al máximo número de uv -trayectorias internamente disjuntas en G .

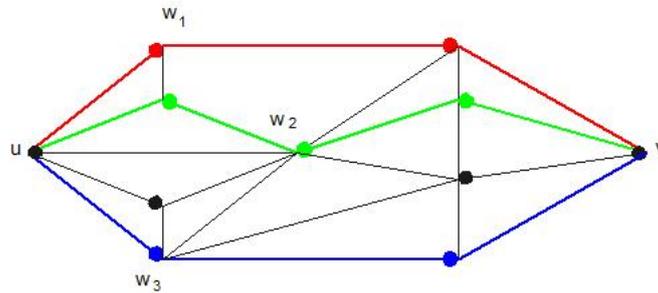


Figura 1.9: Se ilustra el teorema de Menger

Observemos que el conjunto $S = \{w_1, w_2, w_3\}$ de vértices en la gráfica G de la figura 1.9 tiene cardinalidad 3, que es el mínimo número de vértices que necesitamos para separar u de v , y éste es igual al máximo número de trayectorias que los separan.

Árboles

A continuación presentaremos algunas definiciones y propiedades sobre árboles. Además damos definiciones y conceptos sobre árboles binarios, los cuales son una variante especial de árboles.

Definición 1.13. Árbol. Un árbol es una gráfica conexa y acíclica.

Se muestra un ejemplo de árboles en la figura 1.10, en particular de árboles binarios.

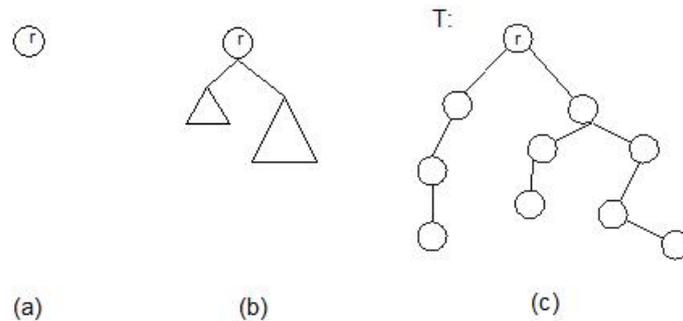


Figura 1.10: Ejemplos de árboles

Teorema 1.4. [5] Un árbol de orden p tiene tamaño $p - 1$; es decir, un árbol con p vértices tiene $p - 1$ aristas.

Corolario 1.1. [5] Sea G una gráfica de orden p . Entonces, los siguientes enunciados son equivalentes:

- i) G es un árbol.
- ii) G es conexa y tiene tamaño $p - 1$.
- iii) G tiene tamaño $p - 1$ y es acíclica.

Teorema 1.5. [5] Todo árbol no trivial tiene al menos dos vértices terminales, es decir, dos vértices cuyo grado es 1.

Teorema 1.6. [5] Si u y v son dos vértices distintos de un árbol T , entonces T posee exactamente una u, v -trayectoria.

Definición 1.14. Árbol generador. Un árbol generador de una gráfica G es una subgráfica de G que contiene a todos los vértices de la gráfica y además cumple con la propiedad de ser árbol.

Definición 1.15. [12] **Árbol binario.** Un árbol binario es vacío, o bien, consiste de un vértice raíz y un par de árboles binarios (posiblemente vacíos), denominados: subárbol izquierdo, T_L , y derecho, T_R .

En la figura 1.10 (a) se muestra un árbol binario constituido únicamente por el vértice raíz; en (b) se ilustra la definición de árbol binario y en (c) se presenta un árbol binario.

Cada elemento en un árbol binario está determinado de manera única por su posición en el árbol.

Considere la figura 1.11. Aquí decimos que el vértice x es el padre de y y de z ; y es el hijo izquierdo de x y z es el derecho.

En el árbol binario la raíz es el único elemento que no tiene padre; todos los demás tienen exactamente un padre. Un elemento (vértice) en el árbol que no tiene hijos se llama **hoja**.

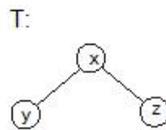


Figura 1.11: Posiciones en un árbol binario

Rigurosamente, la **altura** de un árbol binario $h(T)$ es el número de aristas entre la raíz y la hoja más lejana. De manera formal se define $h(T)$ como sigue:

Si T es vacío, entonces $h(T) = -1$; y en otro caso, $h(T) = 1 + \max\{h(T_L), h(T_R)\}$.

Sea T un árbol binario, no vacío. Para cada elemento x en T , se define $n(x)$, el nivel del nodo x , como sigue: Si x es el elemento raíz, $n(x) = 0$; y en otro caso. $n(x) = 1 + n(\text{padre}(x))$. Se dice que un árbol binario está **perfectamente balanceado** si sus subárboles, derecho e izquierdo, tienen la misma altura y ambos, o son vacíos o están perfectamente balanceados.

Definición 1.16. [12] **Árbol binario completo.** Un árbol binario T es completo si:

- (1) T está perfectamente balanceado excepto, quizá, por el último nivel.
- (2) Todas las hojas en el nivel más bajo están lo más a la izquierda posible.

Se tiene que todo árbol perfectamente balanceado es completo.

Definición 1.17. [12] **Árbol Binomial.** Los árboles Binomial B_k , con $k \in \mathbb{N}$, se definen inductivamente de la siguiente manera:

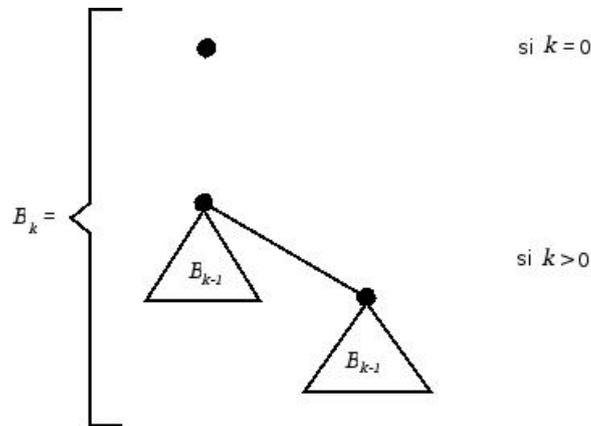


Figura 1.12: Definición de árboles binomiales

- B_0 es un árbol que consiste de un solo nodo; y
- cualquier otro árbol B_k se forma al ligar dos árboles B_{k-1} , esto es, colocando a la raíz de uno como hijo de la raíz del otro. Esto se ilustra en la figura 1.12.

La figura 1.13 muestra los primeros cuatro árboles binomiales. A continuación se describen algunas propiedades de los árboles binomiales.

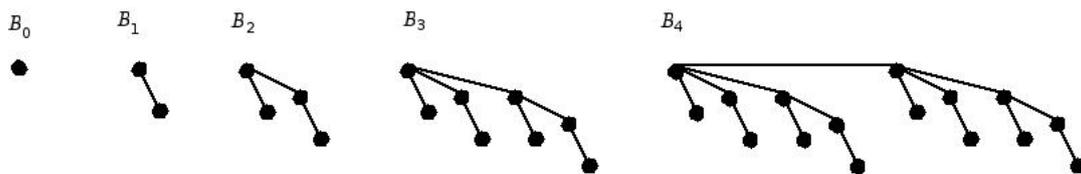


Figura 1.13: Ejemplo de árboles binomiales

Lema 1.1. [12] Para cualquier árbol binomial B_k se cumplen las siguientes propiedades;

1. El árbol B_k tiene 2^k nodos.
2. La altura del árbol es k .
3. Hay exactamente $\binom{k}{i}$ nodos a la profundidad de i para $i = 0, 1, \dots, k$.
4. La raíz tiene grado k , el cual es el mayor de cualquier otro grado de sus nodos; además si los hijos de la raíz son numerados de izquierda a derecha por $k-1, k-2, \dots, 0$, entonces el nodo i es la raíz de un subárbol B_i .

Propiedades especiales de gráficas

Definición 1.18. Isomorfismo [13] Sean G y H dos gráficas. Decimos que G y H son isomorfas, $G \cong H$, si existe una función $\phi : V(G) \rightarrow V(H)$ tal que dos vértices u_1 y v_1 son adyacentes en G si y sólo si los vértices $\phi(u_1)$ y $\phi(v_1)$ son adyacentes en H .

En la figura 1.14 (a) el par de gráficas que se muestran son isomorfas pues la estrella puede redibujarse como un ciclo de longitud 5, denotado por C_5 . En cambio, en la figura 1.14 (b) tenemos dos gráficas, donde todos sus vértices tienen grado 3. Sin embargo, este par de gráficas no son isomorfas, ya que la segunda gráfica contiene un ciclo de longitud 3, C_3 y de esta manera no puede dibujarse como la primera gráfica que es $K_{3,3}$.



Figura 1.14: Ejemplos de (a) gráficas isomorfas (b) gráficas no isomorfas

Definición 1.19. Automorfismo Un automorfismo es una permutación de los vértices de G que preserva adyacencias, es decir, un isomorfismo de G en G .

G :



Figura 1.15: Trayectoria de longitud 2

Sea G la gráfica de la figura 1.15 cuyo conjunto de vértices $V(G) = \{1, 2, 3\}$. Observemos que hay seis permutaciones sobre $V(G)$; sin embargo, solo dos de ellas son isomorfismos de G .

$$\alpha_0 = \begin{pmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \end{pmatrix} \text{ sin embargo } \alpha_1 = \begin{pmatrix} 1 & 2 & 3 \\ 3 & 2 & 1 \end{pmatrix}$$

es también un automorfismo. Consideremos la composición $\alpha_1 \circ \alpha_0$. Veamos que,

$$(\alpha_1 \circ \alpha_0)(1) = \alpha_1(\alpha_0(1)) = \alpha_1(1) = 3,$$

$$(\alpha_1 \circ \alpha_0)(2) = \alpha_1(\alpha_0(2)) = \alpha_1(2) = 2,$$

$$(\alpha_1 \circ \alpha_0)(3) = \alpha_1(\alpha_0(3)) = \alpha_1(3) = 1,$$

$$\text{así, } \alpha_1 \circ \alpha_0 = \begin{pmatrix} 1 & 2 & 3 \\ 3 & 2 & 1 \end{pmatrix} = \alpha_1$$

Consideremos ahora la composición $\alpha_1 \circ \alpha_1$, tenemos:

$$(\alpha_1 \circ \alpha_1)(1) = \alpha_1(\alpha_1(1)) = \alpha_1(3) = 1,$$

$$(\alpha_1 \circ \alpha_1)(2) = \alpha_1(\alpha_1(2)) = \alpha_1(2) = 2,$$

$$(\alpha_1 \circ \alpha_1)(3) = \alpha_1(\alpha_1(3)) = \alpha_1(1) = 3,$$

$$\text{así, } \alpha_1 \circ \alpha_1 = \begin{pmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \end{pmatrix} = \alpha_0$$

Para la gráfica G tenemos que el automorfismo de grupo $\text{Aut}(G) = \{\alpha_0, \alpha_1\}$. Como $\text{Aut}(G)$ tiene dos elementos, este grupo tiene orden dos. Con ayuda de la tabla que se muestra en el cuadro 1.1 podemos ver que la identidad de $\text{Aut}(G)$ es α_0 , el inverso de α_0 es α_0 y de α_1 es α_1 .

\circ	α_0	α_1
α_0	α_0	α_1
α_1	α_1	α_0

Cuadro 1.1: $\text{Aut}(G) = \{\alpha_0, \alpha_1\} \cong \mathbb{Z}_2$

Definición 1.20. Gráfica transitiva en vértices¹ Una gráfica G es transitiva en vértices si dados dos vértices $u, v \in V(G)$, existe un automorfismo $\phi : V(G) \rightarrow V(G)$ tal que $\phi(u) = v$.

Definición 1.21. Gráfica transitiva en aristas² Una gráfica es transitiva en aristas si para todo par $e = \{x, y\}, f = \{u, v\} \in A(G)$, existe un automorfismo $\varphi : e \rightarrow f$, tal que $\varphi(\{x, y\}) = \{u, v\}$.

Obviamente, todo ciclo es transitivo por vértices y por aristas. Supongamos que $n > 2$. Entonces cualquier trayectoria, P_n , no es transitiva en vértices, pues ningún automorfismo puede mandar un vértice de grado 1 a un vértice de grado 2.

La gráfica G en la figura 1.16 es transitiva en vértices pero no por aristas. Mientras que la gráfica H de la figura 1.16 es transitiva en aristas pero no por vértices.

Definición 1.22. Dadas las gráficas G_1 y G_2 , el **producto cuadro** $G_1 \square G_2$ [17] tiene como conjunto de vértices a $V(G_1) \times V(G_2)$, y dos vértices en el producto cuadro, digamos (u_1, u_2) y (v_1, v_2) son adyacentes en $G_1 \times G_2$ si y sólo si:

$$u_1 = v_1 \text{ y } \{u_2, v_2\} \in A(G_2) \text{ o bien } u_2 = v_2 \text{ y } \{u_1, v_1\} \in A(G_2).$$

¹En inglés el término es *vertex-transitive*. Algunos autores utilizan el término *vertex-symmetric* [7].

²En inglés el término es *edge-transitive*. Algunos autores utilizan el término *edge-symmetric* [7].

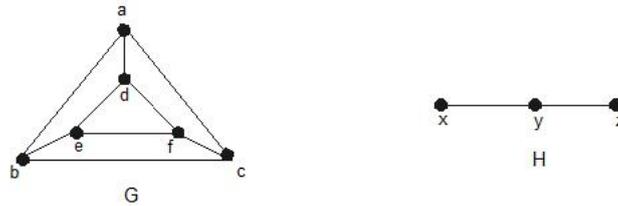


Figura 1.16: Una gráfica transitiva en vértices y una transitiva en aristas

Una manera intuitiva de visualizar el producto cuadro de dos gráficas es la siguiente: Supongamos que $V(G_1) = \{u_1, u_2, \dots, u_m\}$ y $V(G_2) = \{v_1, v_2, \dots, v_n\}$. Tomemos n copias de G_1 , a las que llamamos H_1, H_2, \dots, H_n , y coloquémoslas en el lugar de los vértices v_1, v_2, \dots, v_n , respectivamente, en G_2 . Unimos el vértice $u_i \in H_j$ al vértice $u_i \in H_k$ si y sólo si $\{v_j, v_k\} \in A(G_2)$. Esta operación se ilustra en la figura 1.17.

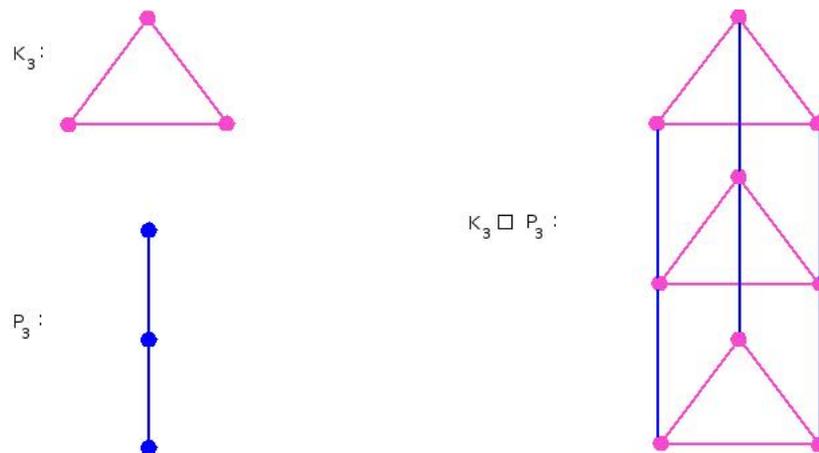


Figura 1.17: Producto cuadro de dos gráficas

Definición 1.23. Gráfica panconexa. Una gráfica G es panconexa si para cada par de vértices $u, v \in V(G)$, $u \neq v$, existe una uv -trayectoria de longitud l en G , tal que $d_G(u, v) \leq l \leq |V(G)| - 1$.

Sea G la gráfica de la figura 1.18 y sean $u, v \in V(G)$. Observemos que la longitud de la trayectoria en color rosa es la distancia entre u y v que es 3 y $|V(G)| = 7$. Para que G sea panconexa debe existir una uv -trayectoria cuya longitud cumpla con la definición anterior, verificamos que no existen trayectorias entre u y v en G cuya longitud sea 5 y 6, de esta manera tenemos que la gráfica no es panconexa.

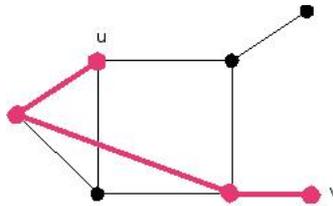


Figura 1.18: Gráfica no panconexa

Definición 1.24. Gráfica pancíclica. Una gráfica G con $|V(G)| = n$, es pancíclica si contiene un ciclo C , tal que $3 \leq \ell(C) \leq n$.

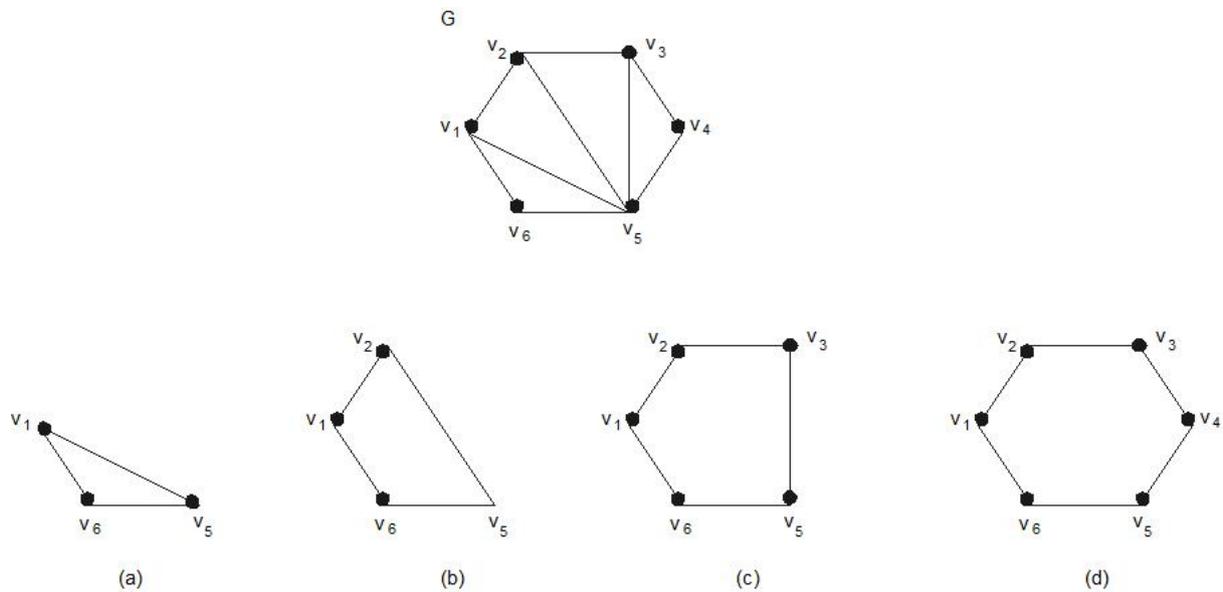


Figura 1.19: (a) $\ell(C) = 3$, (b) $\ell(C) = 4$, (c) $\ell(C) = 5$, (d) $\ell(C) = 6$

Observemos que en la gráfica G de la figura 1.19, con $|V(G)| = 6$, se muestran 4 ciclos, cuya longitud es 3, 4, 5, y 6, respectivamente. De esta manera, podemos concluir que G es pancíclica.

Gráficas de Cayley

En esta sección examinamos una familia grande y versátil de gráficas llamadas gráficas de Cayley o gráficas de grupo. Éstas ofrecen una variedad ilimitada para la construcción de la conexión entre topologías, pero también pueden ser definidas específicamente con un conjunto de transformaciones básicas. Primero daremos una definición formal y discutiremos las propiedades generales que poseen las gráficas de Cayley.

Definición 1.25. [18] **Gráficas de Cayley.** Los vértices de un gráfica de Cayley denotada por CG son los elementos de un grupo finito G , donde el vértice $g \in V(G)$ está conectado al vértice $g * \omega$, Ω denota el conjunto de generadores ω de G con $\omega \in \Omega$ si y sólo si $\omega^{-1} \in \Omega$ y el elemento identidad de G no está en Ω , $*$ es la multiplicación grupo.

Una gráfica de Cayley resulta cuando seleccionamos un conjunto de n símbolos y un conjunto de reglas (elementos de Ω) por lo cual una permutación de los símbolos se puede cambiar por otra. Los vértices se etiquetan con permutaciones de los n símbolos y una arista se dibuja entre dos vértices, si existe una regla por la que la etiqueta de un vértice es una permutación se puede cambiar en la etiqueta de la otra permutación. Las reglas, o transformaciones, son permutaciones en sí mismas.

Un ejemplo sencillo es el siguiente: consideremos un conjunto de símbolos $\{a, b, c, d, e\}$, y una sencilla regla de transformación $\omega = (b, c, d, e, a)$ y $\omega^{-1} = (a, e, d, c, b)$ un desplazamiento cíclico.

La estructura resultante es un ciclo de 5 vértices. Si el conjunto de reglas de transformación es igual a G , es decir, todas las permutaciones están permitidas, la estructura resultante es una gráfica completa.

Arden [1] presenta las siguientes propiedades sobre las Gráficas de Cayley:

1. Son gráficas no dirigidas.
2. Son conexas.
3. Son transitivas por vértices.
4. No tienen lazos; es decir un vértice $v \in V(CG)$ no puede estar conectado consigo mismo.
5. El número de vértices, N , está limitado por el número de permutaciones posibles $N \leq n!$.
6. El grado está dado por el número de generadores $d(v) = |\Omega|$.
7. El **enrutamiento** consiste en ordenar una permutación dada, la dirección o etiqueta del nodo origen, a otra permutación, la dirección o etiqueta del nodo destino.

1.2. Redes de interconexión

En un modelo matemático, podemos representar o identificar situaciones de la vida real o problemas con un sistema matemático. La construcción de modelos matemáticos puede tomar muchas formas e involucrar muchas áreas de las matemáticas. Un área particularmente propensa para modelar es la Teoría de Gráficas.

En ciencias de la computación, los modelos de cómputo se utilizan para describir entidades reales, llamadas computadoras. El modelo da una descripción abstracta que es simple de entender teóricamente y de fácil manipulación matemática.

Algunos ejemplos de redes de interconexión son: los sistemas informáticos, las redes de computadoras, los sistemas de comunicación y los sistemas de transportación. Una de las maneras de comunicarse entre los procesadores es mediante una red de interconexión; es decir, mantener la conexión recíproca entre los nodos de la red. Dos procesadores conectados por un enlace o arista pueden intercambiar datos de forma simultánea. Por lo cual, la estructura matemática es una gráfica no dirigida $G = (V, A)$ y es usada para describir una red de interconexión, donde cada procesador P_i es un vértice en V y si hay un enlace entre dos procesadores P_i y P_j en la red de interconexión, entonces la arista (P_i, P_j) existe entre los correspondientes vértices A de la gráfica. En este trabajo usaremos los términos red de interconexión y gráfica de forma indistinta.

Topologías en redes de interconexión

En esta sección introduciremos algunas redes de interconexión. Supondremos que todas las redes tienen n procesadores o nodos, centrando nuestra atención en redes de interconexión estáticas. Las topologías de red básicas se numeran a continuación:

- 1. Mallas (*Grids*):** Una red en forma de malla se compone de m columnas y n filas. Los procesadores están situados en las intersecciones de dos cables y los recursos computacionales son procesadores cercanos. Las direcciones de los procesadores y los recursos pueden ser fácilmente definido como coordenadas (x, y) en la malla. La figura 1.20 (a) ilustra una malla de dimensión 4×4 .

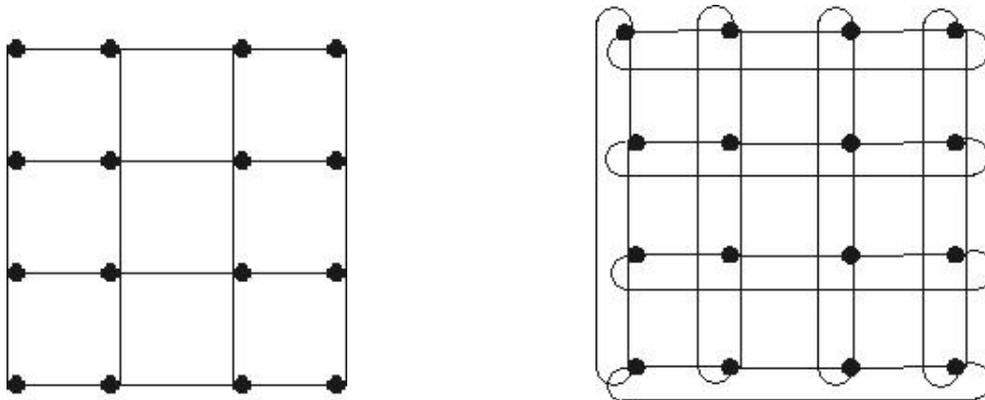


Figura 1.20: (a) malla (b) toro

- 2. Toros (*Torus*):** Una red de toro o toroidal es una versión mejorada de la malla. Una red toroidal es una malla en la que los nodos extremos están conectados. La red toroidal tiene más trayectorias que la malla y también tiene más rutas mínimas. La figura 1.20 (b) muestra un toro de 4×4 .
- 3. Árboles (*Trees*):** En la topología del árbol, los nodos internos son procesadores y las hojas son recursos computacionales. En una topología de árbol hay muchas rutas alternativas entre los nodos. Tiene el inconveniente de que

toda comunicación (ruta) debe pasar por la raíz. En la figura 1.21 se presenta un árbol de nueve nodos.

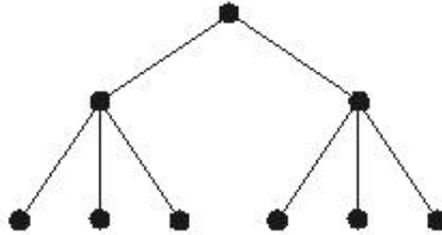


Figura 1.21: Árbol

4. Red Completa (*Complete Network*): La red de interconexión más obvia y general es la gráfica donde cada nodo en ella está conectado directamente con todos los otros nodos de la gráfica. Esta red de interconexión es conocida como **gráfica completa** o K_n , el grado de los vértices en ésta gráfica es $n - 1$ y el diámetro es 1. La figura 1.22 ilustra la gráfica completa K_6 .

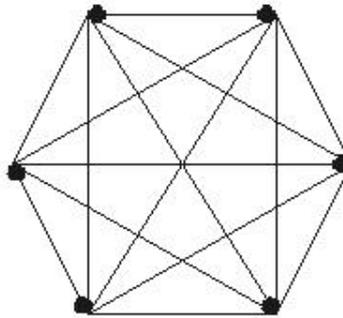


Figura 1.22: Gráfica completa K_6

Como se puede observar, hay $2(n - 1)/2$ enlaces en la red, lo que la hace poco viable en la práctica por dos razones: el alto costo asociado con el número total de aristas y el límite físico de conexiones que puede tener el procesador. Afortunadamente, un pequeño subconjunto de pares conectados suele ser suficiente para obtener algunos algoritmos eficientes en la mayoría de las aplicaciones. En algunos modelos el número de procesadores vecinos es constante, mientras que en otros es una función de n , donde n es la cardinalidad de los procesadores.

5. Estrella (*Star*): Una red en estrella consiste de un procesador en el centro de la estrella y los recursos computacionales o subredes en las puntas de la estrella. Los requerimientos de capacidad en el procesador central son bastante

grandes, porque todo el tráfico entre los picos pasa por el procesador central. Eso causa una posibilidad notable de congestión en el centro de la estrella. En la figura 1.23 se presenta una estrella con siete nodos en total.

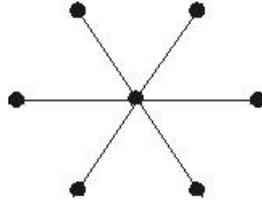


Figura 1.23: Red estrella

6. Arreglo Lineal. (*Linear Array*): El arreglo lineal es la red de interconexión más sencilla y fundamental. En ésta, los n procesadores forman una trayectoria. Cada procesador P_i , con $1 < i < n$, está conectado con dos vecinos llamados P_{i-1} y P_{i+1} , exceptuando los procesadores P_1 y P_n , los que sólo tienen un vecino. El grado de los vértices internos en el arreglo lineal es 2 y el diámetro, la longitud de la trayectoria más larga en el arreglo. La figura 1.24 (a) presenta un arreglo lineal de cinco procesadores.



Figura 1.24: Arreglos lineales

Un arreglo lineal especial es la red de interconexión **ciclo** (*ring*), tal que P_1 y P_n están conectados, por lo que todos los procesadores tienen exactamente dos vecinos. La figura 1.24 (b) muestra un ciclo.

1.3. Algoritmos de comunicación

En un sistema multiprocesador, en función del número de destinos, hay tres tipos de comunicación entre procesadores, los cuales son: enrutamiento, multidifusión y difusión de la información.

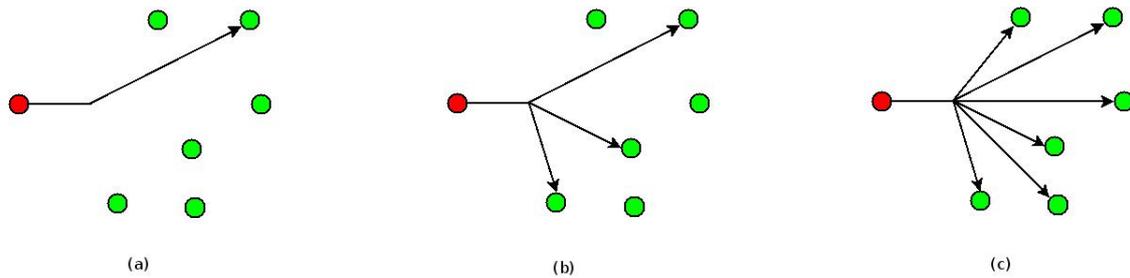


Figura 1.25: Algoritmos de comunicación

Un algoritmo de enrutamiento o *routing* es el proceso por el cual un nodo específico envía información a otro nodo. Esto se ilustra en la figura 1.25 (a).

Una comunicación de multidifusión en una red es donde un nodo origen desea enviar un mensaje a algunos nodos destino. La multidifusión es para determinar qué rutas deben ser seleccionados para entregar el mensaje a los nodos destino de tal manera que se reduce la cantidad de tráfico de comunicación. Este esquema es ilustrado en la figura 1.25 (b).

Se denomina difusión de la información o *broadcasting* al proceso mediante el cual la información que posee un nodo fuente es distribuida al resto de los nodos en la red. Esto se ilustra esquemáticamente en la figura 1.25 (c).

El objetivo de los algoritmos de comunicación es realizar este proceso de la manera más rápida posible. Sin embargo, es sabido que el problema de difusión en una red arbitraria es NP -Completo, incluso para gráficas planas 3- regulares.

Se conocen soluciones óptimas del problema de difusión para algunas familias de redes como la gráfica completa, el Hipercubo y algunas variantes del Hipercubo, entre otras. Algoritmos de enrutamiento y difusión de la información en el hipercubo se han estudiado intensamente, así que recientemente, los investigadores han centrado su atención en el algoritmo de multidifusión.

La construcción de algoritmos eficientes depende fuertemente de la forma en la que han sido generadas las familias de redes. En particular, el Hipercubo y algunas de sus variantes son gráficas sumamente estructuradas y amplias simetrías. Además, poseen propiedades topológicas como son:

1. Diversas trayectorias ajenas por vértices y aristas.
2. Distintas clases de árboles generadores ajenos por aristas.
3. Gráficas de Cayley.

Estas estructuras permiten construir fácilmente algoritmos de comunicación. Por ejemplo, para el problema de enrutamiento podemos aprovechar las trayectorias ajenas por aristas.

Para el de difusión de la información o *broadcasting* podemos utilizar los árboles generadores. Incluso, para ambos problemas, se puede utilizar el operador de grupo asociado a la gráfica de Cayley para mover la información de un nodo a otro, durante la ejecución del algoritmo.

A continuación presentamos algunos algoritmos de enrutamiento.

Enrutamiento

Un algoritmo de enrutamiento determina cómo se envían los datos desde el emisor al receptor. Los algoritmos de enrutamiento están divididos en dos grupos, algoritmos ingenuos³ y adaptivos [21].

Los algoritmos ingenuos también están divididos en dos subgrupos: determinísticos y estocásticos. Los algoritmos ingenuos envían paquetes sin ninguna información acerca de las cantidades de tráfico y las condiciones de la red, los algoritmos determinísticos envían paquetes siempre a lo largo de una misma ruta y los estocásticos se basan en la aleatoriedad.

El algoritmo de enrutamiento tiene como objetivos principales:

1. Encontrar una trayectoria de un nodo origen a un nodo destino en una topología dada.
2. Reducir el número de saltos y latencia, en general, ahorro de energía.
3. Equilibrar la carga de canales de red, maximizando rendimiento.
Ir por rutas de longitud mínima no siempre es eficiente.

Mostraremos ahora, con detalle, dos de estos algoritmos, en los cuales se enviará un mensaje desde un nodo origen a un nodo destino. Estos algoritmos son: el algoritmo de Valiant y el algoritmo ingenuo de enrutamiento mínimo.

Algoritmo de Valiant

El algoritmo de Valiant es un algoritmo de enrutamiento parcialmente estocástico. Un problema principal en el algoritmo de enrutamiento instintivo es que afecten a una carga irregular en la red. La carga es especialmente alta en las zonas medias de la red. El algoritmo de Valiant iguala las cargas de tráfico en las redes que tienen varias trayectorias. La entrega de paquetes se divide en dos fases:

Fase 1. Un paquete se envía primero de un nodo s a un nodo x , elegido al azar.

Fase 2. Después, se envía desde x hasta su destino, utilizando algún algoritmo de los antes mencionados, por ejemplo, el algoritmo mínimo.

³En inglés suelen denominarse *oblivious*.

Para su comprensión ilustraremos el Algoritmo de Valiant sobre el Toro de 6×6 , el cual se muestra en la figura 1.26

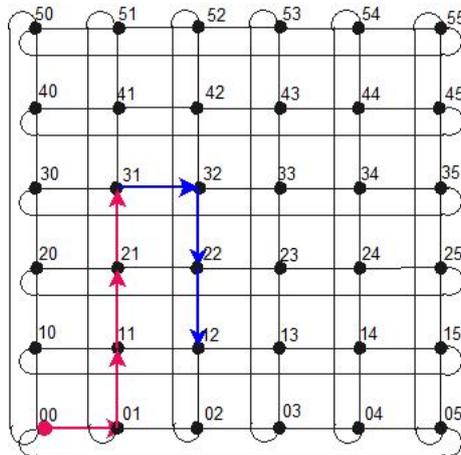


Figura 1.26: Toro de 6×6

- Un paquete será enviado del nodo (00) al nodo (12).
- El paquete se envía primero a un nodo escogido al azar, digamos (31). En este ejemplo, escogemos la longitud de la trayectoria más corta desde el (00) al (31): $00 \rightarrow 01 \rightarrow 11 \rightarrow 21 \rightarrow 31$. Se ilustra con color rosa en la figura 1.26.
- El paquete entonces se envía al nodo final, (12), nuevamente utilizando la trayectoria de longitud más corta del nodo (31) al nodo (12), la cual se muestra de color azul en la figura 1.26. De esta manera, resulta la siguiente trayectoria: $00 \rightarrow 01 \rightarrow 11 \rightarrow 21 \rightarrow 31 \rightarrow 32 \rightarrow 22 \rightarrow 12$. Ésta se puede ver claramente en la figura 1.26.

No olvidemos que este proceso se realiza en paralelo; es decir, todos los procesadores envían un mensaje a la vez. El algoritmo de Valiant no envía directamente por la ruta más corta para evitar colisiones y, en general, repartir el tráfico que se genera en toda la red. En el ejemplo anterior se ilustró únicamente como trabaja un procesador.

Algoritmo ingenuo de enrutamiento mínimo

Consideremos la malla de la figura 1.27. Utilizando el algoritmo ingenuo de enrutamiento mínimo, enviaremos un mensaje desde el nodo origen (00) al nodo destino (21). Comenzamos situándonos en el nodo origen y al azar enviamos el mensaje a algún otro nodo siempre y cuando esté dentro del cuadrante mínimo, el cual, en nuestro ejemplo está marcado de color azul.

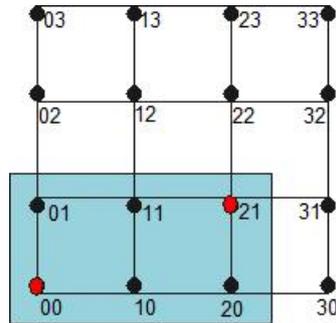


Figura 1.27: Enrutamiento ingenuo

Para cada nodo en el cuadrante (00, 10, 20, 01, 11, 21) determinamos una ruta mínima que pase por un nodo x . En la figura 1.28 se ilustran las rutas que a continuación se describen.

Comenzamos en el nodo $x = (00)$ y observemos que hay tres rutas posibles que son las siguientes: $00 \rightarrow 01 \rightarrow 11 \rightarrow 21$; $00 \rightarrow 10 \rightarrow 20 \rightarrow 21$; $00 \rightarrow 10 \rightarrow 11 \rightarrow 21$.

Con $x = 01$ existe una posible ruta : $00 \rightarrow 01 \rightarrow 11 \rightarrow 21$

Con $x = 10$ dos posibles rutas : $00 \rightarrow 10 \rightarrow 20 \rightarrow 21$ y $00 \rightarrow 10 \rightarrow 11 \rightarrow 21$.

Con $x = 11$ también son dos posibles rutas: $00 \rightarrow 10 \rightarrow 11 \rightarrow 21$ y $00 \rightarrow 01 \rightarrow 11 \rightarrow 21$.

Con $x = 20$ una posible ruta: $00 \rightarrow 10 \rightarrow 20 \rightarrow 21$.

Con $x = 21$ tres posibles rutas: $00 \rightarrow 01 \rightarrow 11 \rightarrow 21$; $00 \rightarrow 10 \rightarrow 20 \rightarrow 21$ y $00 \rightarrow 10 \rightarrow 11 \rightarrow 21$.

De entre todas estas posibles rutas, notemos que la ruta: $00 \rightarrow 01 \rightarrow 11 \rightarrow 21$ es la ruta más corta; es decir, la óptima para enviar el mensaje en el menor tiempo desde el nodo 00 al nodo 21.

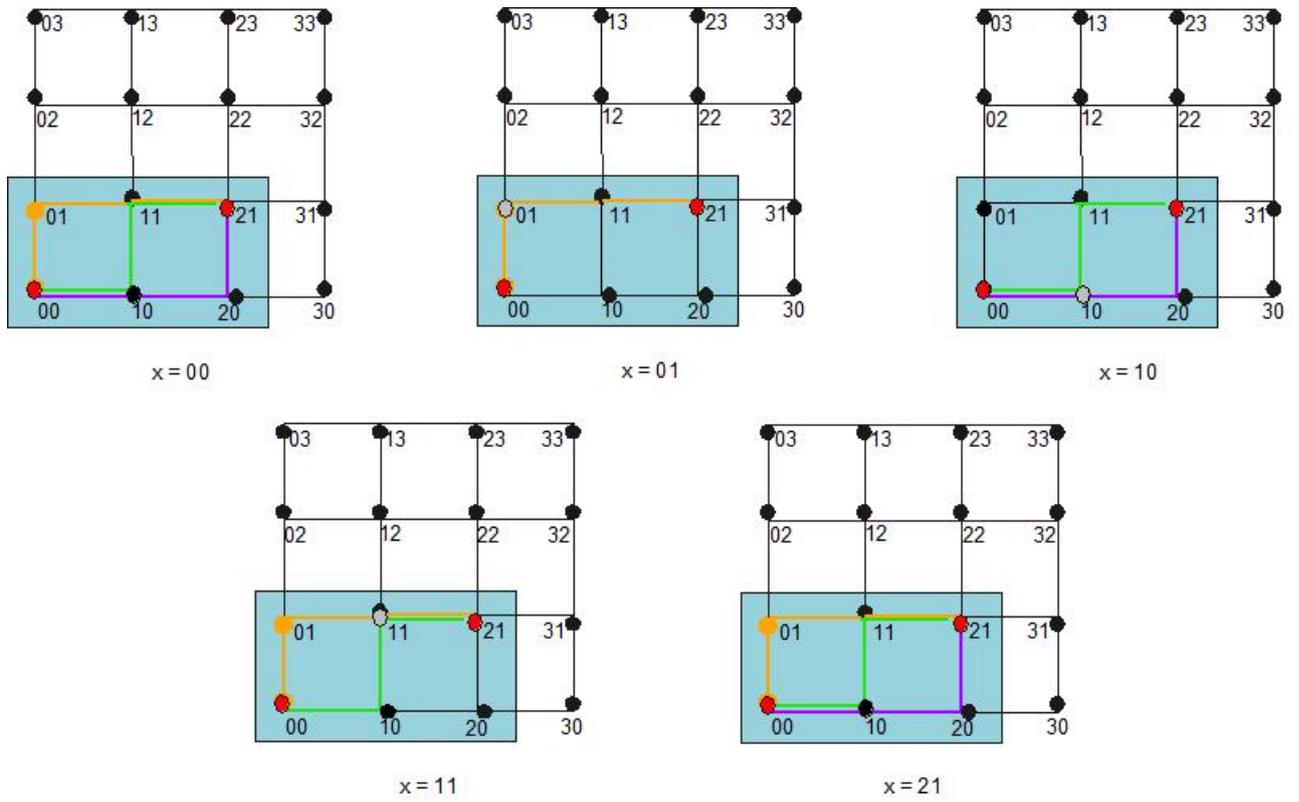


Figura 1.28: Rutas desde el nodo 00 al nodo 21

Capítulo 2

El hipercubo y sus propiedades

En este capítulo definiremos el hipercubo de dimensión n , Q_n , describiendo algunas de sus propiedades topológicas y algorítmicas más importantes. Aprovecharemos el enfoque de la teoría de gráficas para explotar y explicar sus propiedades topológicas; por otro lado, describiremos algoritmos de enrutamiento y difusión de la información para la red de interconexión Q_n .

En la gráfica del hipercubo se representa a los procesadores por vértices o nodos. Un 3-cubo se representa como un cubo ordinario en tres dimensiones, cuya cardinalidad es $8 = 2^3$ nodos. Más general, podemos construir un n -cubo de la siguiente forma: Primero, los 2^n nodos son etiquetados por 2^n números binarios de 0 a $2^n - 1$, de longitud n . Entonces una arista entre dos nodos es dibujada si y sólo si sus números binarios difieren en uno y sólo un bit.

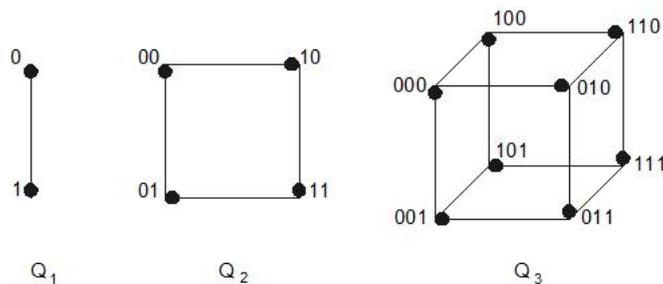


Figura 2.1: Ejemplo de Hipercubos

Definición 2.1. [22] **Hipercubo.** El hipercubo Q_n , también llamado n -cubo, se define recursivamente como $Q_1 = K_2$, si $n = 1$ y $Q_n = Q_{n-1} \square K_2$, si $n \geq 2$. Otra manera de describir al n -cubo consiste en representar sus vértices por una colección de n -adas formadas por ceros y unos, donde dos vértices de Q_n son adyacentes si y sólo si sus correspondientes n -adas difieren en exactamente una coordenada.

Ilustramos los primeros tres ejemplos del Hipercubo, Q_1 , Q_2 y Q_3 , en la figura 2.1.

2.1. Propiedades básicas

El hipercubo Q_n es una gráfica bipartita, n -regular, n -conexa, con 2^n vértices. Q_n tiene diámetro n y se sabe que es una gráfica hamiltoniana [25]; además de ser una gráfica de Cayley, es transitiva por vértices y por aristas.

Al dividir un n -cubo en la subgráfica de todos los nodos cuyo bit inicial es 0 y la subgráfica cuyo bit inicial es 1, las dos subgráficas obtenidas son tales que cada nodo de la primera subgráfica está conectada a un nodo de la segunda subgráfica. Si eliminamos las aristas que unen las dos subgráficas obtenemos dos $(n-1)$ -hipercubos. Esta operación de dividir el n -cubo en dos $(n-1)$ -cubos de manera que los nodos de los dos $(n-1)$ -cubos están en correspondencia uno a uno la definimos como **romper**¹ un n -cubo.

Dada una numeración cualquiera de los nodos, **romper** equivale a separar la gráfica en dos subgráficas obtenidas al considerar todos los nodos cuyo i -ésimo bit es 0 y aquellos cuyo i -ésimo bit es 1. Esto se conoce como romper a lo largo de la i -ésima dirección. Como hay n -bits, hay n -direcciones. Estas propiedades se resumen en la siguiente proposición.

Proposición 2.1. Hay n formas distintas de romper un n -cubo en dos $(n-1)$ -cubos de modo que sus respectivos vértices están relacionados en una correspondencia biyectiva.

Teniendo en cuenta el etiquetado de los nodos en la Definición 2.1, la gráfica del n -cubo se separa en dos subgráficas: Una cuya etiqueta tiene un 1 en la posición i y una cuya etiqueta tiene un 0 en la posición i . Ilustramos la separación de un 3-cubo en la figura 2.2.

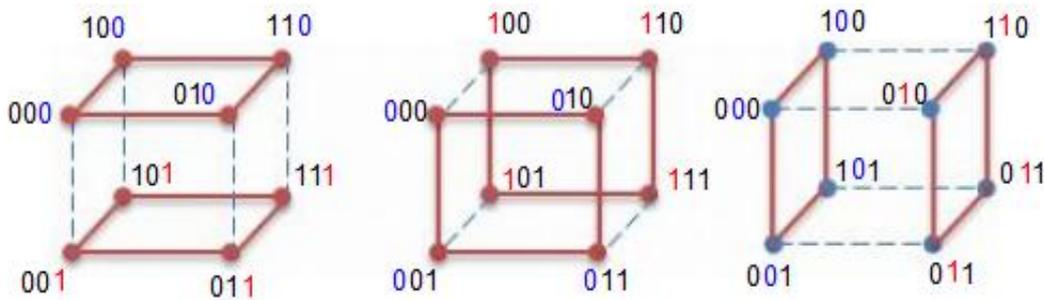


Figura 2.2: Separación de un 3-cubo

El siguiente resultado nos dice cuántas maneras hay para etiquetar un n -cubo. Ilustramos en la figura 2.3 esta proposición.

Proposición 2.2. Hay $n! \cdot 2^n$ formas distintas de numerar los nodos de un n -cubo de acuerdo a la definición anterior.

¹En inglés esta operación se denomina *tear*.

Demostración. Usaremos inducción sobre n , la dimensión del hipercubo.

El resultado es válido para $n = 1$. Ahora, suponemos que es verdad para $n - 1$, es decir hay $(n - 1)! \cdot 2^{n-1}$ formas distintas en las cuales los 2^{n-1} nodos de un $(n - 1)$ -cubo cumple con las características de la proposición.

Para numerar los nodos del n -cubo, primero escogemos su bit inicial, ya sea cero o uno. Para ello, vamos a romper al n -cubo en dos $(n - 1)$ -cubos, hay n maneras diferentes de hacerlo.

Después, numeramos los nodos del primer $(n - 1)$ -cubo. Por hipótesis de inducción hay $(n - 1)! \cdot 2^{n-1}$ formas distintas de hacerlo y agregamos un 1 como bit inicial. Análogamente para el segundo $(n - 1)$ -cubo y entonces agregamos un cero como bit inicial. Una segunda numeración se obtiene invirtiendo los bits de cero y uno. Así obtenemos un total de

$$\begin{aligned} n [(n - 1)! \cdot 2^{n-1} + (n - 1)! \cdot 2^{n-1}] &= n [2^{n-1}((n - 1)! + (n - 1)!)] \\ &= n [2^{n-1}(2(n - 1)!)] \\ &= n [2^n(n - 1)!] \\ &= n! \cdot 2^n \end{aligned}$$

numeraciones diferentes de los vértices del n -cubo. \diamond

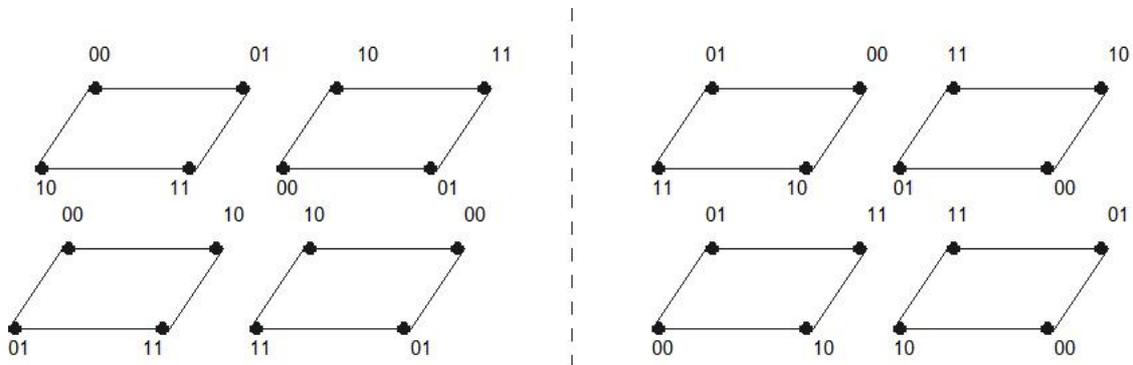


Figura 2.3: Numeración de los nodos en un 2-cubo

Observemos que sin la restricción de que la numeración debe ajustarse a la definición 2.1, tendríamos un total de $(2^n)!$ formas distintas de numerar 2^n vértices diferentes, un número mucho mayor al de la proposición. La Proposición 2.2 tiene la siguiente consecuencia importante.

Proposición 2.3. Cualquiera dos nodos adyacentes A y B de un n -cubo son tales que los nodos adyacentes a A y los adyacentes a B están en una correspondencia uno a uno.

Demostración. Como los nodos considerados son vecinos, las entradas de A y B difieren en un solo bit, digamos el i -ésimo bit. Vamos a romper el n -cubo a lo

largo de la dirección i . Entonces los vecinos de A y los de B se pueden poner en una correspondencia uno a uno mediante la asignación de un nodo cuya etiqueta tiene un 1 en la i -ésima posición a aquel cuya etiqueta tiene un 0 en la i -ésima posición. \diamond

Esta propiedad se ilustra en la figura 2.4. El nodo A y el nodo B son adyacentes, se indica la adyacencia por medio de la arista en color morado; los nodos adyacentes al nodo A están marcados con color rojo, excepto el nodo B ; los nodos adyacentes al nodo B están marcados en azul, excepto el nodo A ; de esta manera las aristas en amarillo indican la correspondencia uno a uno entre los nodos rojos y azules.

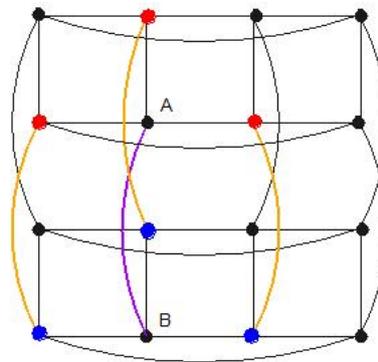


Figura 2.4: Correspondencia uno a uno entre nodos adyacentes a A y B

Podemos definir la **paridad** de un nodo como positiva si el número de bits 1 que están en su etiqueta binaria es par y negativa en otro caso. Observemos que nodos vecinos tienen paridad opuesta. El siguiente resultado se sigue de esta propiedad.

Proposición 2.4. Para toda $n \in \mathbb{N}$, el n -cubo, Q_n , no tiene ciclos de longitud impar.

Demostración. Consideremos C un ciclo en Q_n , tal que $C = \{a_1, a_2, \dots, a_t\}$, con $a_1 = a_t$. Conforme viajamos del nodo a_i , al nodo a_{i+1} , $1 \leq i \leq t-1$, la paridad cambia. Como $a_1 = a_t$, debe haber un número par de cambios, es decir, la longitud del ciclo es necesariamente par. \diamond

Esta proposición nos indica que el hipercubo es una gráfica bipartita, lo que significa que podemos partir a sus vértices en dos conjuntos, de tal manera que los vértices que estén en el mismo conjunto no sean adyacentes entre sí.

Dados dos nodos de un n -cubo siempre hay una trayectoria entre ellos. Una forma de llegar al nodo B desde el nodo A es modificando los bits de A , uno a la vez en orden, para transformar A en B . Cada vez que un bit cambia significa que hemos cruzado una arista. Esto demuestra una forma simple de construir una trayectoria de longitud a lo más n entre cualesquiera dos vértices de un n -cubo. Además, recalcando que el diámetro de una gráfica es la distancia máxima entre cualquiera dos de sus nodos, podemos concluir el siguiente resultado:

Proposición 2.5. [22] El n -cubo es una gráfica conexa de diámetro n .

Considerando que el hipercubo Q_n tiene $N = 2^n$ vértices, la proposición anterior nos dice que Q_n tiene diámetro $\log N$; es decir, que el diámetro de Q_n es logarítmico con respecto al número de vértices.

Las proposiciones anteriores establecen algunas de las propiedades básicas del n -cubo como gráfica. Nos gustaría reconocer a un hipercubo de una manera sencilla, es decir, caracterizarlo.

Fijemos nuestra atención en la malla G_1 de 4×4 que se muestra en la figura 2.5. Desearíamos saber si esta malla corresponde a la gráfica de un hipercubo; es decir, si los 16 nodos pueden ser numerados conforme a la Definición 2.1 y además cumplir con las proposiciones antes mencionadas. Es claro que todos los nodos en la malla deben tener el mismo grado, por lo que esta malla no puede ser la gráfica de un hipercubo aunque los nodos adyacentes a A y B estén en correspondencia uno a uno. Sin embargo, la malla G_2 de la figura 2.5 cumple con lo antes mencionado.

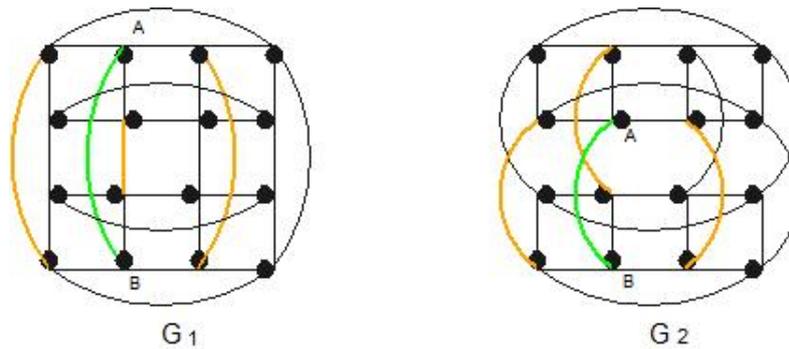


Figura 2.5: Se ilustran dos mallas de 4×4

De esta manera, tenemos la caracterización del hipercubo en el siguiente teorema. Este resultado se puede verificar directamente en la malla G_2 de la figura 2.5:

Teorema 2.1. Una gráfica $G = (V, A)$ es un n -cubo si y sólo si las siguientes cuatro condiciones se satisfacen:

1. V tiene 2^n vértices.
2. Cada vértice tiene grado n .
3. G es conexa.
4. Cualquier par de vértices adyacentes A y B son tales que los nodos adyacentes a A y los adyacentes a B están relacionados en una correspondencia uno a uno.

Demostración. Condición necesaria: Si G es un n -cubo, las condiciones 1 a 4 se satisfacen como resultado de las definiciones y algunas proposiciones previas.

Condición suficiente: La prueba es por inducción sobre n , la dimensión del hipercubo. Es claro que la propiedad se satisface para $n = 1$. Suponemos que es verdad para $n - 1$; es decir, que cualquier gráfica que tiene 2^{n-1} nodos y que satisface las propiedades 1 a 4 es un $(n - 1)$ -cubo. La prueba consiste en separar la gráfica en dos subgráficas cada una de las cuales tiene las mismas propiedades para $n - 1$.

Usaremos los colores rojo y azul para colorear los nodos de la gráfica. Consideremos cualesquiera dos nodos adyacentes de la gráfica, digamos R (un nodo rojo) y A (un nodo azul). Ilustraremos este teorema con la gráfica del 4-cubo que aparece en la figura 2.6. De acuerdo a la propiedad 4, los vecinos de R y los de A están relacionados en una correspondencia uno a uno. Podemos, además, colorear los vecinos del nodo rojo (excepto el nodo que está coloreado de azul) con rojo y los vecinos del nodo azul (excepto el nodo rojo) con azul. Este proceso puede continuarse hasta terminar con todos los nodos.

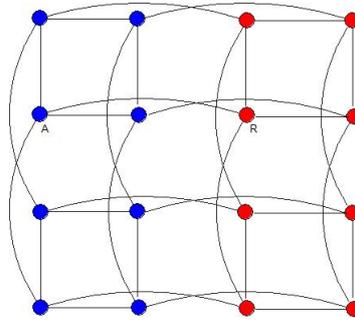


Figura 2.6: Correspondencia uno a uno entre nodos rojos y azules

Decimos que dos nodos de diferente color son **nodos opuestos** si están unidos por una arista. Observemos lo siguiente:

- a) Todos los nodos han sido coloreados con azul o rojo. Esto es porque la gráfica es conexa y además hay una trayectoria entre el nodo original R (ó A) hacia cualquier nodo.
- b) Exactamente la mitad de los nodos tienen color rojo y la otra mitad tienen color azul, pues todos los nodos A y R están conectados en una correspondencia uno a uno.
- c) Es claro que los nodos rojos forman una gráfica conexa: por construcción, cada nodo está conectado al nodo R . La misma propiedad se preserva para el nodo A .
- d) Consideremos las dos subgráficas obtenidas al remover todas las aristas con un extremo rojo y el otro azul. En la figura 2.7 se ilustra con claridad este hecho. Así, cada nodo pierde exactamente una arista, es decir, su grado es $(n - 1)$. Entonces la propiedad 4 se cumple para la subgráfica de los nodos rojos y similarmente para los nodos azules.

e) Por la propiedad 4, y por construcción, dos nodos rojos son adyacentes si y sólo si sus nodos opuestos de color azul son adyacentes.

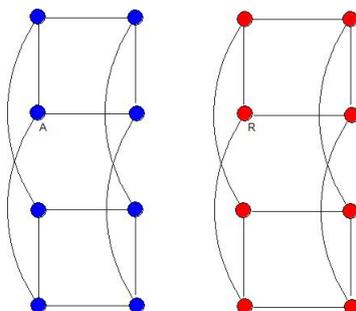


Figura 2.7: Q_4 menos las aristas cuyos extremos son rojo y azul

Por hipótesis de inducción y por las observaciones (b), (c), (d) y (e), la subgráfica de los nodos rojos es un $(n-1)$ -cubo. Ahora, etiquetemos a los nodos rojos de acuerdo a la definición y usemos las mismas etiquetas para los nodos azules opuestos a ellos. Podemos usar las mismas etiquetas por lo mencionado en el inciso (e). Agregamos el bit 0 al inicio de las etiquetas de los nodos rojos y el bit 1 al inicio de las etiquetas de los nodos azules y de esta manera obtenemos la etiqueta de los nodos de la gráfica inicial, como se observa en la figura 2.8. \diamond

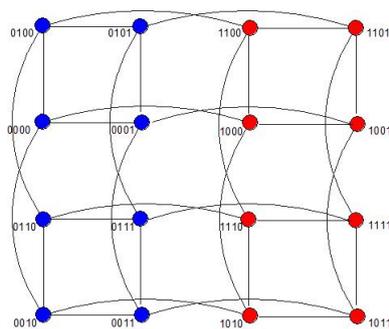


Figura 2.8: Nodos etiquetados en la gráfica del 4-cubo

En la figura 2.9 se muestra la gráfica del 4-cubo, la cual cumple con las condiciones antes mencionadas. Observemos que G es conexa, tiene $2^4 = 16$ vértices y además para todo $v \in V(G)$, $d(v) = 4$.

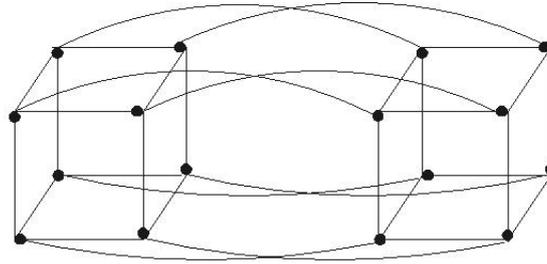


Figura 2.9: 4-cubo

2.2. Árboles generadores

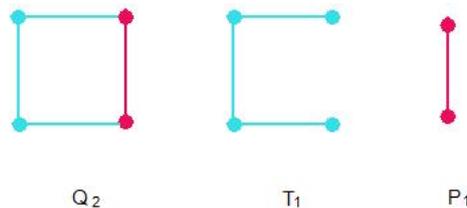
En esta sección describiremos propiedades de distancia, trayectorias y árboles en el hipercubo Q_n que nos servirán para optimizar la información que se envía entre cualesquier dos vértices. Recordemos que en las Ciencias de la computación a los procesadores se les representa por vértices de una gráfica.

Dado que cada vértice del hipercubo Q_n tiene grado n , el número de aristas es $n \cdot 2^{n-1}$ y puesto que el número de aristas en un árbol generador es uno menos que el número de vértices, un árbol generador de Q_n tiene $2^n - 1$ aristas.

Por lo tanto el máximo número de árboles generadores ajenos por aristas en un hipercubo de dimensión n [2] es $\lfloor (n \cdot 2^{n-1}) / (2^n - 1) \rfloor = \lfloor n/2 \rfloor$.

Fijaremos nuestra atención en hipercubos de dimensión par, $2n$, de modo que el máximo número de árboles generadores ajenos por aristas es exactamente n sobrando n aristas no utilizadas por algún otro árbol generador.

A continuación presentamos una demostración constructiva de que existen n árboles generadores ajenos por aristas en un hipercubo de dimensión $2n$. La prueba procede por inducción sobre n . Además, mostramos que las aristas que no se utilizan forman una trayectoria de longitud n .

Figura 2.10: T_1 : árbol generador, P_1 : Trayectoria de longitud uno en Q_2

Teorema 2.2. Sea $G = Q_{2n}$ un hipercubo de dimensión $2n$. Entonces G contiene n árboles generadores ajenos por aristas y con las n aristas restantes se forma una trayectoria.

Demostración. Haremos la demostración por inducción sobre n .

Para $n = 1$, el hipercubo de dimensión 2, Q_2 , mostrado en la figura 2.10, contiene un árbol generador T y la arista que sobra forma una trayectoria de longitud 1.

Supongamos por hipótesis de inducción que el hipercubo Q_{2n} contiene n árboles generadores ajenos por aristas y con las aristas que sobran se forma una trayectoria de longitud n .

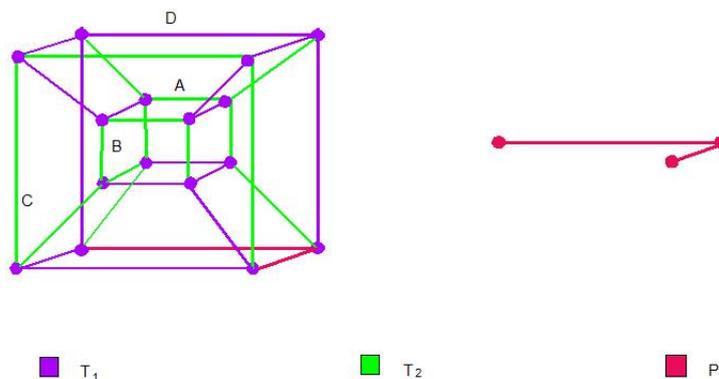


Figura 2.11: Árboles ajenos por aristas para Q_4

Sea $Q_{2(n+1)}$ un hipercubo $2(n+1)$ -dimensional. Deseamos construir $n+1$ árboles generadores ajenos por aristas en $Q_{2(n+1)}$ y con las aristas que sobran formar una trayectoria de longitud $n+1$.

Notemos que $Q_{2(n+1)}$ puede ser particionado en cuatro hipercubos de dimensión $2n$. En la figura 2.12 se ilustra a Q_4 y cuatro hipercubos de dimensión 2, denotados por A, B, C y D , tal que A está conectado a B y D . El hipercubo B está conectado a A y C ; C está conectado a B y D . Finalmente, el hipercubo D está conectado a C y A .

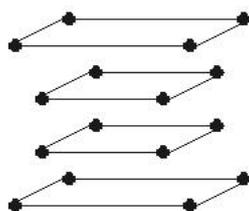


Figura 2.12: Partición de Q_4 en cuatro hipercubos de dimensión 2

A los árboles generadores en el subcubo A los denotamos por A_1, A_2, \dots, A_n , y a los subárboles idénticos y correspondientes en los subcubos B, C y D los denotamos por $B_1, B_2, \dots, B_n; C_1, C_2, \dots, C_n$ y D_1, D_2, \dots, D_n , respectivamente. De manera similar, a los vértices que están sobre la trayectoria de longitud n en el subcubo A se les denotará por a_1, a_2, \dots, a_{n+1} y a los vértices correspondientes en los subcubos B, C y D se les denotará por $b_1, b_2, \dots, b_{n+1}; c_1, c_2, \dots, c_{n+1}$ y d_1, d_2, \dots, d_{n+1} , respectivamente.

Los primeros $n - 1$ árboles generadores de $Q_{2(n+1)}$ se construyen de la siguiente manera: Para cada $i, 1 \leq i \leq n - 1$, sea T_i la subgráfica de $Q_{2(n+1)}$ que tiene las aristas en A_i, B_i, C_i, D_i y a las aristas $(a_i, b_i), (b_i, c_i)$ y (c_i, d_i) . Por construcción, cada subgráfica T_i es conexa, acíclica y contiene a todos los vértices en $Q_{2(n+1)}$, y es así un árbol generador de $Q_{2(n+1)}$. Además, por construcción, esos $n - 1$ árboles generadores de $Q_{2(n+1)}$ son ajenos por aristas.

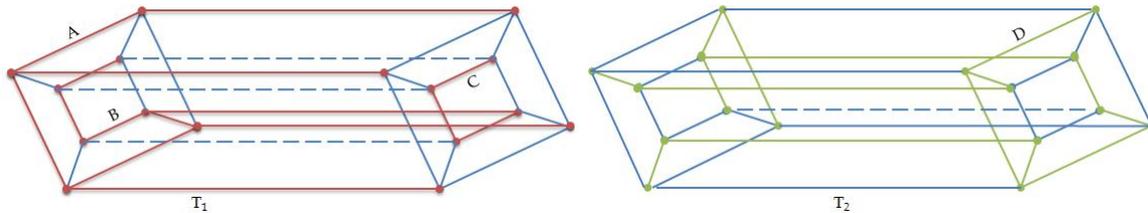


Figura 2.13: Árboles ajenos por aristas

El siguiente árbol de $Q_{2(n+1)}$ se construye como la unión de A_n, B_n y C_n , las aristas $(a_n, b_n), (b_n, c_n)$ y todas las 2^{2n} aristas entre los subcubos A y D . Por construcción, esta subgráfica es también un árbol generador de $Q_{2(n+1)}$ y es ajeno por aristas con los $n - 1$ árboles construidos previamente.

El último árbol generador de $Q_{2(n+1)}, T_{n+1}$, se construye de la siguiente forma: Primero, D_n es incluido en T_{n+1} . Después, a través de las 2^{2n} aristas entre los subcubos C y D , $n - 1$ fueron usadas en la construcción de T_1, \dots, T_{n-1} y así $2^{2n} - n + 1$ de esas aristas no fueron utilizadas. Excluyendo a la arista (c_{n+1}, d_{n+1}) , permanecen $2^{2n} - n$ aristas sin utilizar entre C y D que son incluidas en T_{n+1} ; así conectamos $2^{2n} - n$ vértices de C a T_n^D . \diamond

Distancias y trayectorias

Cualquier sistema multiprocesador debe permitir a sus procesadores el intercambio de datos entre todos sus vértices. Sean A y B dos vértices del n -cubo y consideremos el problema de enviar datos desde el vértice A al vértice B . La forma en que esto se logra es moviendo los datos a lo largo de una trayectoria de A a B cruzando un número, posiblemente pequeño, de los procesadores. Por definición, la longitud de una trayectoria entre dos vértices es simplemente el número de aristas de la ruta. Como ya se mencionó antes, existe una trayectoria de longitud a lo más n entre dos vértices cualesquiera. Para llegar de A a B , es suficiente pasar sucesivamente por los vértices cuyas etiquetas son las obtenidas mediante la modificación de los bits de A uno por uno y en orden para transformar A en B . Suponiendo que A y B difieren sólo en i bits, decimos que su **distancia de Hamming** es $H(A, B) = i$ la longitud de la trayectoria será i . Claramente, no hay trayectoria de menor longitud entre los vértices A y B .

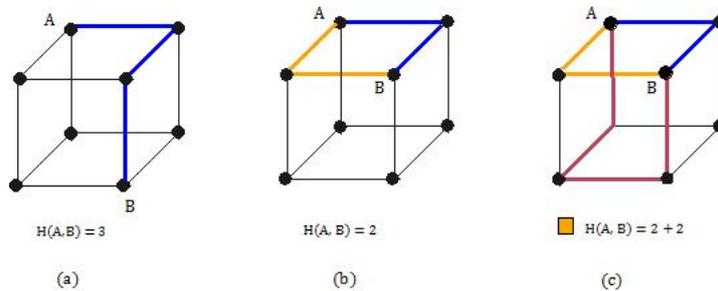


Figura 2.14: Se ilustra la distancia de Hamming

Proposición 2.6. La distancia mínima entre los vértices A y B es igual al número de bits en los que difieren A y B , es decir a la distancia de Hamming, $H(A, B)$.

Proposición 2.7. Sean A, B dos vértices de un n -cubo.

- La mínima distancia entre A y B es $H(A, B)$.
- Si $H(A, B) < n$ entonces hay $H(A, B)$ trayectorias ajenas por aristas de longitud $H(A, B)$ entre A y B .
- Si $H(A, B) < n$ entonces hay n trayectorias paralelas entre A y B cuya longitud es a lo más $H(A, B) + 2$.

Ilustramos la proposición con la gráfica del 3-cubo que se muestra en la figura 2.14 donde en (a) se muestra la distancia mínima entre el vértice A y B ; en (b) dos trayectorias paralelas entre los vértices A y B ; y en (c) tres trayectorias ajenas por aristas.

2.3. Inmersión de otras topologías en el hipercubo

En esta sección analizaremos el problema de mapear otras topologías (anillos y mallas) en el hipercubo. Dada una gráfica $G = (V, A)$ cuya cardinalidad de los vértices es a lo más 2^n , asignaremos los vértices de la gráfica a los nodos del hipercubo de modo que todo vértice adyacente en la gráfica corresponda a nodos adyacentes en el n -cubo. Existen dos razones principales por las cuales dichas asignaciones son importantes.

- Algunos algoritmos pueden ser desarrollados para otra arquitectura para la cual funciona perfectamente. Entonces se podría aplicar el mismo algoritmo con un poco de esfuerzo de programación. Si la arquitectura original se puede mapear en el hipercubo, esto será fácil de lograr.
- Un problema dado puede tener una estructura bien definida que conduce a un patrón particular de comunicación. El mapeo de la estructura puede resultar en un ahorro sustancial en el tiempo de comunicación. Si la malla está perfectamente mapeada en el hipercubo, entonces sólo se requiere comunicación local entre los nodos del hipercubo, lo que resulta en ahorros importantes en tiempos de transferencia.

2.3.1. Inmersión sobre ciclos y arreglos lineales

Dado un ciclo con 2^n vértices, consideremos el problema de asignar sus vértices en los nodos de un hipercubo de tal manera que se preserven las adyacencias entre los nodos. Otra forma de ver este problema es buscando un ciclo de longitud $N = 2^n$ que pase por cada nodo una y sólo una vez, es decir, buscamos un ciclo hamiltoniano en un hipercubo.

Si numeramos los nodos de un hipercubo de acuerdo con la definición 2.1, es decir, de modo que dos nodos vecinos difieren en uno y sólo un bit, un ciclo hamiltoniano simplemente representa una secuencia de números binarios de n bits tales que cualesquiera dos números sucesivos tienen sólo un bit diferente, de modo que todos los números binarios que tengan n bits estarán representados. Las sucesiones binarias con estas propiedades se llaman **códigos Gray**.

Hay varias formas distintas en las cuales un código Gray puede ser generado, pero el método más conocido es el **código Gray reflejado**, que se construye de la siguiente manera:

- Comenzamos con la sucesión de los dos números 1-bit, 0 y 1. Éste es un código Gray 1-bit.
- Para construir un código Gray 2-bit tomamos la misma sucesión e insertamos un 0 en frente de cada número, después tomamos la sucesión en orden inverso e insertamos un 1 en frente de cada número. En otras palabras, obtenemos la sucesión:

$$G_1 = \{0, 1\} \qquad G_2 = \{00, 01, 11, 10\}$$

- Repetimos el mismo proceso para construir un código Gray 3-bit: tomamos la sucesión anterior e insertamos un 0 en frente, invertimos la sucesión e insertamos un 1 como bit inicial. Así, generamos la siguiente sucesión:

$$G_3 = \{000, 001, 011, 010, 110, 111, 101, 100\}.$$

De manera general, denotamos por G_i^R a la sucesión obtenida a partir de G_i invirtiendo el orden de las entradas y por $0G_i$ a la sucesión obtenida al poner un 0 a cada elemento de la sucesión. Análogamente para $1G_i$. Así, el código Gray de cualquier orden puede ser generado de manera recursiva por: $G_{n+1} = \{0G_n, 1G_n^R\}$.

En la figura 2.15 se muestra un ciclo de longitud 8 generado por el código Gray.

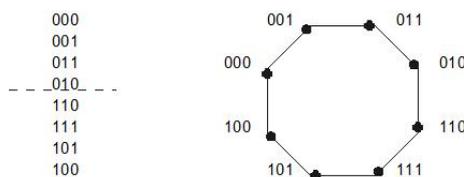


Figura 2.15: Ciclo de longitud 8

Los códigos Gray nos permiten mapear ciclos, cuyas longitudes son potencias de 2, en hipercubos. Supongamos ahora que tenemos un ciclo de longitud ℓ que queremos mapear en un hipercubo. Primero, observemos que el mapeo es posible sólo cuando ℓ es par ya que de acuerdo con la proposición 2.4, un hipercubo no admite ciclos de longitud impar. Por lo tanto, suponemos que $4 \leq \ell \leq 2^n$. El problema es encontrar un ciclo de longitud ℓ en el n -cubo, donde ℓ es par.

Sea $m = \frac{\ell-2}{2}$ y denotamos por $G_{n-1}(m)$ el código Gray $(n-1)$ -bits que consta de los primeros m elementos de G_{n-1} . Entonces, un ciclo que tiene la propiedad deseada es el siguiente: $\{0G_{n-1}(m), 1G_{n-1}(m)^R\}$.

Cuando $\ell = 2^n$, se obtiene un caso particular de la fórmula $G_{n+1} = \{0G_n, 1G_n^R\}$. Así, podemos afirmar lo siguiente.

Proposición 2.8. Un ciclo de longitud ℓ puede ser mapeado en el n -cubo cuando ℓ es par y $4 \leq \ell \leq 2^n$.

Por ejemplo, en la figura 2.16 el ciclo de dimensión 8 es inmerso en el 3-cubo y cumple que $4 \leq 8 \leq 2^3$.

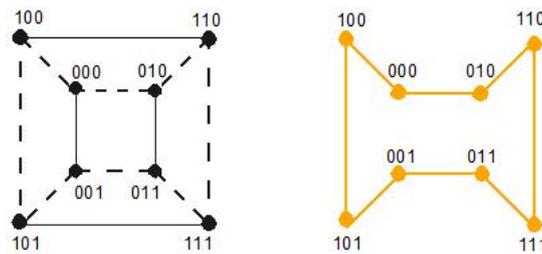


Figura 2.16: Q_3 y un ciclo de longitud 8

Finalmente, no hay dificultad en realizar la inmersión de una trayectoria, en lugar de un ciclo, en el n -cubo. Es suficiente mapear los nodos del arreglo lineal P_0, P_1, \dots, P_l de longitud arbitraria $l \leq 2^n - 1$, sucesivamente, en los nodos g_0, g_1, \dots, g_l del n -cubo. Dado un arreglo lineal de longitud arbitraria l , el n -cubo de dimensión más pequeña en el cual puede ser mapeado es el cubo de dimensión $n = \lceil \log_2(l + 1) \rceil$.

2.3.2. Inmersión de mallas en el hipercubo

La inmersión de mallas, de cualquier dimensión, en el hipercubo es una de las propiedades más atractivas de la topología del n -cubo. Observemos que también es un código Gray. Ésta es una de las razones principales por la cual la arquitectura del hipercubo es exitosa. Sea $M = m_1 \times m_2 \times \dots \times m_d$ una malla en el espacio d -dimensional R^d , además suponemos que el tamaño de la malla en cada dirección es una potencia de 2; es decir, para cada i , $m_i = 2^{p_i}$. Sea $n = p_1 + p_2 + \dots + p_d$ y consideremos el problema de mapear los vértices de la malla en el n -cubo, de tal manera que se asigne un vértice de la malla a un nodo del n -cubo. Observemos que tenemos los nodos suficientes para acomodar un vértice de la malla por nodo en el n -cubo.

Entenderemos por mapeo de la malla en el cubo a la asignación de vértices de la malla en los nodos del n -cubo, de tal manera que la propiedad de preservar adyacencias se conserve, es decir, tal que dos vértices vecinos en la malla son asignados a nodos vecinos en el n -cubo.

En el caso $d = 1$, el problema se resolvió en la sección anterior mediante el uso de códigos Gray. Se muestra a continuación cómo extender las ideas de la sección anterior para más de una dimensión. Nuestro argumento se ilustra mejor con un ejemplo.

Consideremos una malla bi-dimensional y de 8×4 ; es decir, tenemos $d = 2$, $p_1 = 3$, $p_2 = 2$, $n = p_1 + p_2 = 5$. A un número binario A de cualquier nodo del 5-cubo lo podemos separar en dos partes: sus primeros 3 bits y sus últimos 2 bits, lo cual escribimos de la siguiente forma:

$$A = b_1, b_2, b_3, c_1, c_2$$

donde b_i y c_j son bits 0 ó 1. Se desprende de la definición del n -cubo que cuando los dos últimos bits son fijos, los 2^{p_1} nodos resultantes forman un p_1 -cubo, con $p_1 = 3$. Del mismo modo, cada vez que fijamos los primeros tres bits obtenemos un p_2 -cubo. Entonces el mapeo es claro. La elección de un código Gray de 3 bits para la dirección x y un código Gray de 2 bits para la dirección y , el vértice (x_i, y_j) de la malla es asignado al nodo b_1, b_2, b_3, c_1, c_2 donde b_1, b_2, b_3 es el código Gray de 3 bits para x_i , mientras $c_1 c_2$ es el código Gray de 2 bits para y_j .

El mapeo se ilustra en la figura 2.17, donde se obtiene el número de nodo binario de cualquier vértice de la malla mediante la concatenación de la coordenada binaria x y su coordenada binaria y . Por ejemplo, el nodo $A = 00101$ del 5-cubo se mapea al vértice $(001, 01)$ de la malla. También se ilustran los nodos $B = 00111$ y $C = 01010$

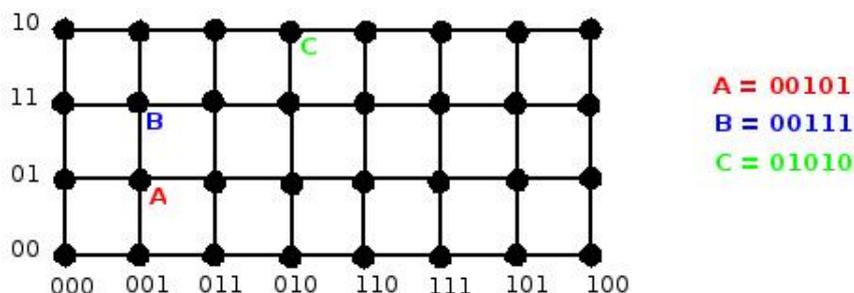


Figura 2.17: Código Gray 2-dimensional para una malla de 8×4

Así, si llamamos sucesión Gray cualquier subsucesión de un código Gray, observamos que cualquier columna de la malla forma una sucesión Gray y cualquier renglón de puntos de la malla forma una sucesión Gray. Entonces nos referimos a los códigos antes definidos como $2D$ -código Gray. Podemos formular el siguiente teorema.

Teorema 2.3. [22] Cualquier malla $M = m_1 \times m_2 \times \cdots \times m_d$ en el espacio d -dimensional R^d , donde $m_i = 2^{p_i}$ puede ser mapeada en un n -cubo donde $n = p_1 + p_2 + \cdots + p_d$. La numeración de los vértices en la malla es cualquier numeración de tal manera que su restricción a cada i -ésima variable es una sucesión Gray.

2.4. Algoritmo de enrutamiento

En esta sección aplicamos el algoritmo de enrutamiento sobre el hipercubo Q_n . Hemos visto que existe la inmersión del hipercubo Q_n a cualquier malla, en particular a mallas de dos dimensiones. También tenemos que Q_n con $n \leq 6$, puede mapearse directamente a un Toro de 4×4 . Entonces podemos aplicar de forma inmediata los algoritmos de enrutamiento vistos anteriormente, sobre la malla bi-dimensional y el Toro.

A continuación ilustramos cómo aplicar el enrutamiento sobre Q_4 . Primero realizamos la inmersión de Q_4 en el Toro de 4×4 ; la figura 2.18 muestra la misma. Después, utilizamos el algoritmo de Valiant y el de la ruta más corta.

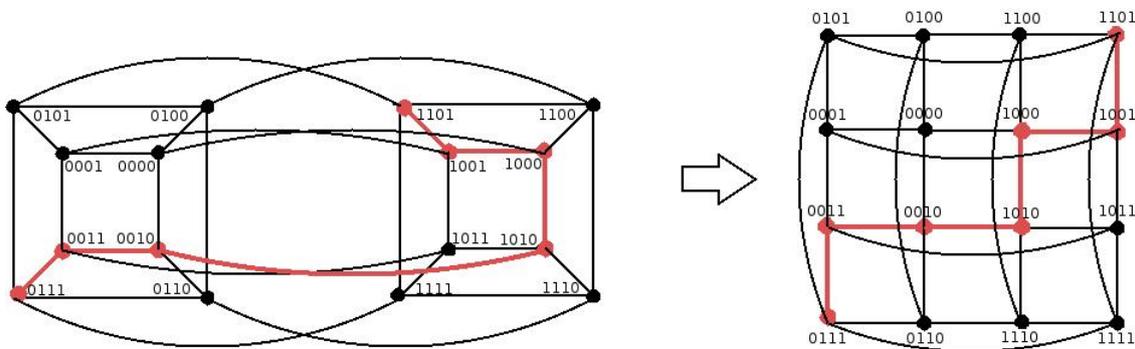


Figura 2.18: Inmersión del 4-cubo en el toro de 4×4

Supongamos que deseamos enviar un mensaje desde el nodo origen 0111 al nodo destino 1101. Los pasos realizados por el enrutamiento son:

1. Situados en el nodo origen, de acuerdo al algoritmo de Valiant, escogemos un nodo al azar, sea 0010 dicho nodo.
2. Encontramos la ruta más corta del nodo 0111 al nodo 0010; la cual resulta ser: $0111 \rightarrow 0011 \rightarrow 0010$.
3. A partir de este nodo, nuevamente con el algoritmo de ruta mínima, encontramos la trayectoria del nodo 0010 al nodo 1101, la cual resulta ser: $0010 \rightarrow 1010 \rightarrow 1000 \rightarrow 1001 \rightarrow 1101$.
4. Finalmente, unimos estas dos rutas obteniendo la ruta más corta por la cual se envía el mensaje y es la siguiente: $0111 \rightarrow 0011 \rightarrow 0010 \rightarrow 1010 \rightarrow 1000 \rightarrow 1001 \rightarrow 1101$.

La existencia de las inmersiones del hipercubo en las mallas facilita el enrutamiento.

2.5. Difusión de la información

Los hipercubos juegan un papel importante en los sistemas de computación en paralelo debido a su estructura simple e idónea para el desarrollo de algoritmos. Una gran cantidad de literatura se ha dedicado a la investigación de sus propiedades topológicas, así como en aplicaciones de computación paralela.

El uso de árboles generadores ajenos por aristas, para la difusión de la información en redes, proporciona un número importante de ventajas, incluyendo el aumento a la tolerancia a fallos. Por lo tanto, los diseños de árboles en varias clases de redes han sido ampliamente investigados. En esta sección, basado en un concepto simple llamado distancia de Hamming en el cuadrado latino, presentamos un nuevo algoritmo para generar n árboles generadores ajenos por aristas de Q_n [23].

El algoritmo de difusión de la información en el hipercubo, Q_n , consiste de n pasos ($i = 0, 1, 2, \dots, n - 1$), que se puede representar en un árbol generador binomial (SBT), representado en la figura 2.19.

En el árbol generador binomial basado en el algoritmo de difusión de la información, para el paso i el nodo raíz r y los nodos activos $x : x_{n-1}, \dots, x_i, \dots, x_1, x_0$ envían los datos recibidos a sus correspondientes nodos $y : x_{n-1}, x_{n-2}, \dots, \bar{x}_i, \dots, x_1, x_0$, donde \bar{x}_i se refiere a la negación del i -ésimo bit del nodo x para $1 \leq i \leq n$. Si en un árbol generador, cualquier arista se rompe, entonces los datos se bloquean y el proceso de transmisión no se completará. Una solución se logra mediante el uso del algoritmo de difusión de la información basado en los árboles generadores ajenos por aristas para aumentar la fiabilidad de la transmisión de datos.

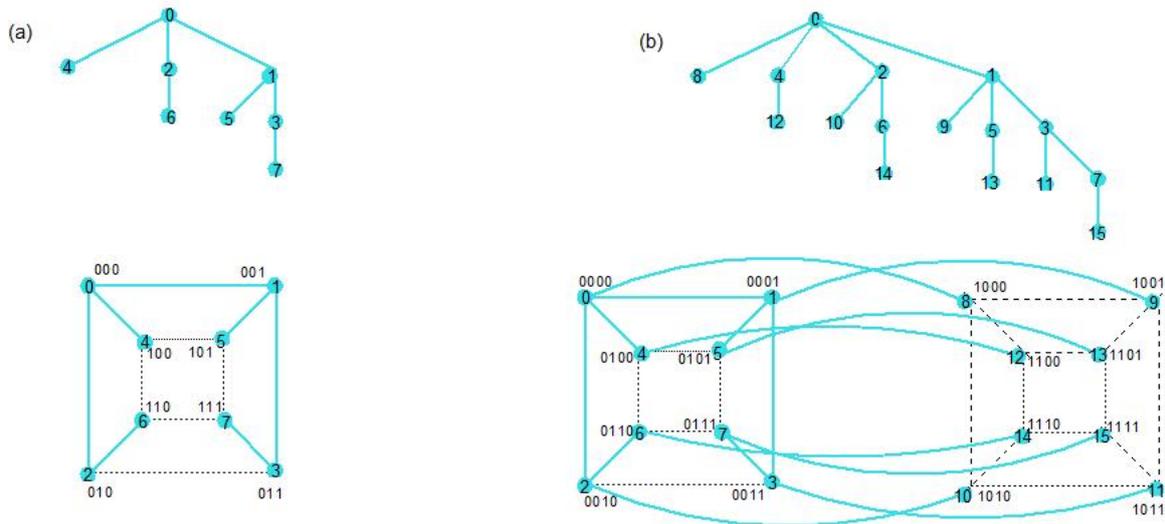


Figura 2.19: Dos árboles binomiales generadores con $r = 0$: (a) sobre Q_3 y (b) sobre Q_4

Recientemente aparecieron dos algoritmos paralelos para construir n árboles generadores ajenos por aristas con una altura óptima sobre Q_n , [23]. A través de los dos árboles ajenos por aristas se puede enviar la información a toda la red.

Para enviar un mensaje de un nodo origen a un nodo destino en Q_k , de manera simultánea, existen algoritmos que nos permiten enviar el mensaje a través de k trayectorias ajenas por aristas.

Un **cuadrado latino** es una matriz cuadrada con n^2 entradas, tomadas de un conjunto H de n elementos distintos tal que ninguno de ellos aparece más de una vez en un renglón o columna de la matriz. Sea $x, x \neq 0$, un vértice de Q_k con $x = (x_{k-1}, x_{k-2}, \dots, x_0)$, un **cuadrado latino con distancia de Hamming**. Respecto a x es un cuadrado latino cuyas entradas pertenecen al conjunto $H_x = \{i | 0 \leq i \leq k-1 \text{ y } x_i = 1\}$. Nótese que la distancia de Hamming del vértice x al vértice 0 en Q_k se define como la cardinalidad del conjunto, $|H_x|$. En seguida se da la definición de las matrices.

Definición 2.2. Sea $x \in V(Q_k)$, supongamos que la distancia de Hamming de x a 0 en Q_k es t y sea $H_x = \{i_0, i_1, \dots, i_{t-1}\}$, tal que $i_0 < i_1 < \dots < i_{t-1}$. Entonces las matrices abajo definidas se llaman circulante creciente, $I_k(x)$, y circulante decreciente, $D_k(x)$, respectivamente.

$$I_k(x) = \begin{pmatrix} i_0 & i_1 & \cdots & i_{t-2} & i_{t-1} \\ i_1 & i_2 & \cdots & i_{t-1} & i_0 \\ \vdots & & \cdots & & \\ i_{t-2} & i_{t-1} & \cdots & i_{t-4} & i_{t-3} \\ i_{t-1} & i_0 & \cdots & i_{t-3} & i_{t-2} \end{pmatrix} \text{ y } D_k(x) = \begin{pmatrix} i_{t-1} & i_{t-2} & \cdots & i_1 & i_0 \\ i_{t-2} & i_{t-3} & \cdots & i_0 & i_{t-1} \\ \vdots & & \cdots & & \\ i_1 & i_0 & \cdots & i_3 & i_2 \\ i_0 & i_{t-1} & \cdots & i_2 & i_1 \end{pmatrix}$$

Además, la entrada i_{j+1} en $I_k(x)$ es llamada sucesor de i_j y se denota por $\text{succ}(i_j)$, donde los índices j y $j+1$ se toman módulo t . Análogamente para i_{j-1} en $D_k(x)$.

Consideremos el vértice 13, con etiqueta $x = 1101$ en Q_4 ; entonces $H_x = \{0, 2, 3\}$ y, así, las dos matrices definidas anteriormente están dadas por:

$$I_k(x) = \begin{pmatrix} 0 & 2 & 3 \\ 2 & 3 & 0 \\ 3 & 0 & 2 \end{pmatrix} \text{ y } D_k(x) = \begin{pmatrix} 3 & 2 & 0 \\ 2 & 0 & 3 \\ 0 & 3 & 2 \end{pmatrix}$$

$$\begin{aligned} \text{donde } \text{succ}(0) &= 2 \\ \text{succ}(2) &= 3 \\ \text{succ}(3) &= 0. \end{aligned}$$

Existen dos algoritmos paralelos para construir k árboles generadores ajenos por aristas sobre Q_k , basados en el cuadrado latino con distancia de Hamming ($HDLS$), uno cuya raíz r es el 0 y en el otro r es arbitraria para $0 \leq r \leq k-1$.

Yang *et al.* [28] propusieron el algoritmo paralelo para la construcción de los k árboles generadores ajenos por aristas sobre Q_k con raíz en el vértice 0. En este caso, la matriz $HDLS$ con $r = 0$ se introdujo para generar todas las trayectorias únicas. Para cada vértice $x \in V(Q_k) \setminus r$ hay una única $HDLS_k(r, x)$ de tamaño $t \times t$, donde $t \leq k$ representa el número de 1 en x . A continuación se presenta, en el Algoritmo 2.1, el pseudo-código para

la construcción paralela de k árboles generadores ajenos por aristas considerando como raíz al vértice $r = 0$.

En el listado del algoritmo 2.2, se da el pseudo-código para la construcción paralela de k arboles generadores para cualquier r arbitraria.

Algorithm 2.1: Construcción paralela de k *ISTs*(raíz $r = 0$)

```

1 foreach  $x \neq r$  en  $Q_k$  con etiqueta  $x = x_{k-1}, \dots, x_i, \dots, x_1, x_0$  do
2   for  $i = 0$  a  $k - 1$  do
3     if  $(x_i = 0)$  then
4       padre  $(T_i, x) = x + 2^i$ 
5     if  $(x_i = 1)$  then
6       padre  $(T_i, x) = x - 2^{succ(i)}$  // aplicar HDLS para  $succ(i)$ 

```

Algorithm 2.2: Construcción paralela de k *ISTs*, con r arbitraria

```

1 foreach  $x \neq r$  en  $Q_k$  do
2   con  $r = r_{k-1}, \dots, r_i, \dots, r_1, r_0$  y  $x = x_{k-1}, \dots, x_i, \dots, x_1, x_0$ 
3 for  $i = 0$  a  $k - 1$  do
4   if  $(r_i = x_i)$  then
5     if  $x_i = 0$  then
6       padre  $(T_i, x) = x + 2^i$ 
7     if  $x_i = 1$  then
8       padre  $(T_i, x) = x - 2^i$ 
9   if  $(r_i \neq x_i)$  then
10    // aplicar GHDLS para  $succ(i)$ 
11    if  $(x_{succ(i)} = 0)$  then
12      padre  $(T_i, x) = x + 2^{succ(i)}$ 
13    if  $(x_{succ(i)} = 1)$  then
14      padre  $(T_i, x) = x - 2^{succ(i)}$ 

```

Capítulo 3

Variantes del hipercubo

En este capítulo se definen e ilustran algunas de las variantes del hipercubo que han surgido en los últimos años, citando algunas de sus propiedades topológicas.

3.1. Hipercubo torcido

Observemos que mediante el intercambio de cualesquiera dos aristas independientes, es decir que no son incidentes en un mismo vértice, en cualquier 4-ciclo del n -cubo ($n \geq 3$), su diámetro disminuye en una unidad. Esto nos lleva a definir una nueva clase de gráficas n -regulares: el hipercubo torcido, denotado por TQ_n , con 2^n vértices y diámetro $n - 1$, que tiene al $(n - 1)$ -cubo como subgráfica.

A continuación presentaremos la definición formal.

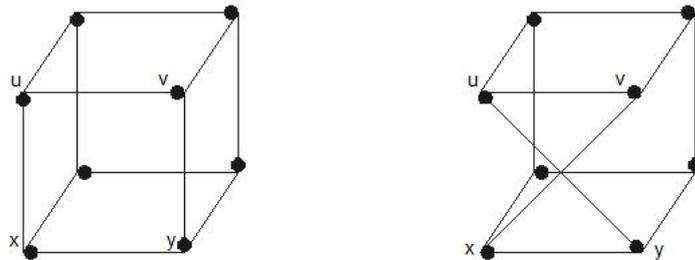


Figura 3.1: Un 3-Cubo, Q_3 y un 3-Cubo Torcido, TQ_3

Definición 3.1. [8] **Hipercubo torcido n -dimensional** (*Twisted n -cube*).

Sea C un 4-ciclo, el de longitud más corta en Q_n . Sean $\{u, x\}$ y $\{v, y\}$ dos aristas en C . La gráfica del cubo torcido n -dimensional, TQ_n , se construye como se muestra:

- Se eliminan las aristas $\{u, x\}$ y $\{v, y\}$ de Q_n .
- Entonces, conectamos el vértice u al vértice y , y el vértice v al vértice x .
- $TQ_n = Q_n - \{ux, vy\} + \{uy, vx\}$.

Nótese que por construcción, TQ_n es n -regular, tiene 2^n vértices y $n \cdot 2^{n-1}$ aristas como el hipercubo Q_n . También, observemos que TQ_n tiene dos copias disjuntas de Q_{n-1} como subgráficas. Se sabe que tiene diámetro de $\lceil n+1/2 \rceil$ y es hamiltoniana; además se ha demostrado que el árbol binario completo sobre $2^n - 1$ vértices, que no es una subgráfica del n -cubo, es una subgráfica de TQ_n .

Aunque el cubo puede ser torcido alrededor de cualquier 4-ciclo, usualmente usaremos el Cubo Canónico Torcido Q_n , donde los vértices u, v, x y y tienen las entradas $b(u) = 000 \dots 0$, $b(v) = 010 \dots 0$, $b(x) = 100 \dots 0$ y $b(y) = 110 \dots 0$.

La figura 3.1 muestra a Q_3 y a TQ_3 .

3.2. Hipercubos localmente torcidos

Los hipercubos localmente torcidos, LTQ_n , son una red de interconexión introducida recientemente en la computación en paralelo. Como variante del hipercubo, Q_n , LTQ_n tiene mejores propiedades conservando el mismo número de aristas y de procesadores; es una gráfica n -regular y n -conexa; es hamiltoniana para $n \geq 3$, además:

$$\text{diam}(LTQ_n) = \begin{cases} n - 1 & \text{si } n = 3 \text{ o } 4 \\ \lceil \frac{1}{2}(n + 3) \rceil & \text{si } n \geq 5. \end{cases}$$

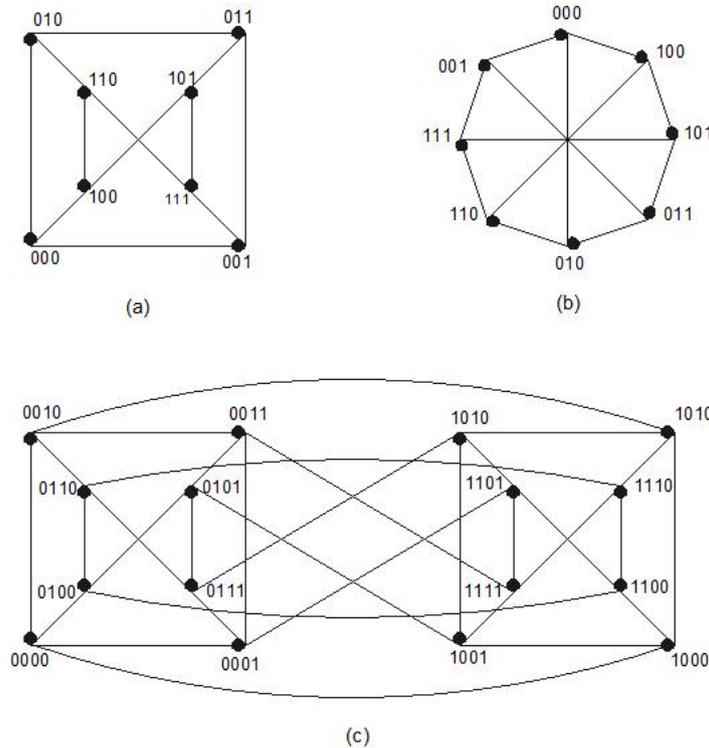


Figura 3.2: LTQ_n , para $n = 3, 4$

Definición 3.2. [25] **Hipercubos localmente torcidos** (*Locally Twisted Cube*).

Los cubos localmente torcidos, LTQ_n ($n \geq 2$), se definen recursivamente de la siguiente manera:

1. LTQ_2 es una gráfica isomorfa a Q_2 .
2. Para $n \geq 3$, LTQ_n se construye a partir de dos copias disjuntas de LTQ_{n-1} de acuerdo a los siguientes pasos:
 - a) Sea $0LTQ_{n-1}$ la gráfica obtenida al poner 0 al inicio de cada vértice de LTQ_{n-1} .
 - b) Sea $1LTQ_{n-1}$ la gráfica obtenida al poner 1 al inicio de cada vértice de LTQ_{n-1} .
 - c) Todo par de vértices x y x' con $x = 0x_2, x_3, \dots, x_n$ en $0LTQ_{n-1}$ y $x' = 1(x_2 + x_n)x_3, \dots, x_n$ en $1LTQ_{n-1}$, serán unidos por una arista, donde '+' representa la suma módulo 2.

Para $n \leq 3$, Yang *et al.* [25] mostraron que LTQ_n es pancíclica. Las gráficas que se muestran en la figura 3.2 son LTQ_3 y LTQ_4 .

3.3. Hipercubo cruzado

El hipercubo cruzado [19] es una alternativa atractiva para el hipercubo. Es una versión más generalizada del hipercubo torcido [3] que se obtiene mediante el intercambio de un par de aristas del hipercubo. Tanto el hipercubo cruzado como el hipercubo tienen el mismo número de vértices y el mismo grado en ellos.

Recientes estudios de investigación han revelado que el hipercubo cruzado, como una red de interconexión, posee muchas propiedades deseadas que no están disponibles en el hipercubo. El diámetro del hipercubo cruzado es aproximadamente la mitad que la del hipercubo. Como consecuencia, el número de pasos para la comunicación de ciertos algoritmos necesarios en el hipercubo cruzado es sólo la mitad de la necesaria en el hipercubo. El hipercubo cruzado contiene a un árbol binario completo óptimo como una subgráfica mientras que el hipercubo no [22].

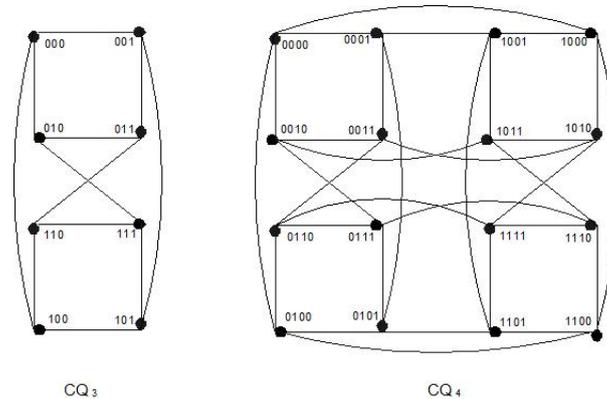


Figura 3.3: Imágenes de CQ_3 y CQ_4

Definición 3.3. [25] **Hipercubo cruzado** (*Crossed Cube*).

El hipercubo cruzado n -dimensional CQ_n , es una gráfica $G = (V, A)$ que se construye recursivamente como se muestra: CQ_1 es la gráfica completa K_2 . Para $n \geq 2$, CQ_n consta de dos cubos cruzados idénticos $(n-1)$ -dimensionales, CQ_{n-1}^0 y CQ_{n-1}^1 .

El vértice $u = 0u_{n-2}, \dots, u_0 \in V(CQ_{n-1}^0)$ y el vértice $v = 1v_{n-2}, \dots, v_0 \in V(CQ_{n-1}^1)$ son adyacentes en CQ_n si y sólo si:

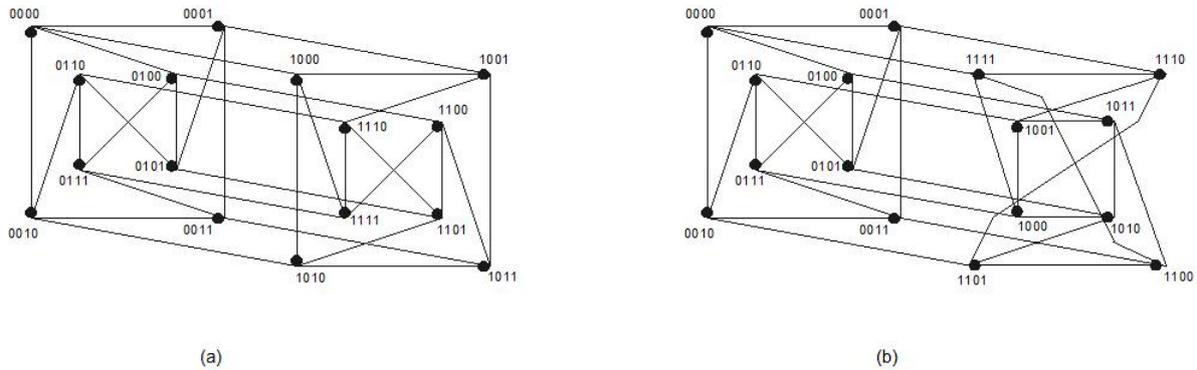
1. $u_{n-2} = v_{n-2}$, si n es par y
2. $u_{2i+1}u_{2i} \sim v_{2i+1}v_{2i} \in \{(00, 00), (10, 10), (01, 11), (11, 01)\}$, para $0 \leq i < \lfloor \frac{n-1}{2} \rfloor - 1$.

El hipercubo cruzado tiene una estructura similar a la del hipercubo, incluyendo la construcción recursiva, el mismo número de vértices, el mismo número de aristas y la conexidad. Sin embargo, el diámetro de CQ_n es aproximadamente la mitad que el diámetro del hipercubo y el diámetro es un factor importante en la velocidad para el envío de mensajes de computación en paralelo. Además, si $n \geq 5$, CQ_n no es transitiva en vértices y para $n \geq 3$ no es transitiva en aristas. Para $n = 1, 2, 3, 4$ CQ_n es una gráfica de Cayley.

Los hipercubos cruzados CQ_3 y CQ_4 se ilustran en la figura 3.3.

3.4. Hipercubo de Moebius

Los hipercubos de Moebius son variantes del hipercubo que dan un mejor rendimiento con el mismo número de aristas y vértices. El diámetro de los hipercubos de Moebius es alrededor de la mitad del diámetro del hipercubo equivalente, y el promedio de número de pasos entre los vértices o procesadores para un hipercubo de Moebius es cerca de dos tercios de la mitad de un hipercubo.

Figura 3.4: (a) $0MQ_4$, (b) $1MQ_4$

Definición 3.4. [9] **Hipercubo de Moebius** (*Moebius cube*).

El hipercubo de Moebius n -dimensional, MQ_n , es una gráfica no dirigida cuyo conjunto de vértices es el mismo que el conjunto de vértices de Q_n y el vértice $X = x_1, x_2, \dots, x_n$ es adyacente a otros n vértices Y_i , ($1 \leq i \leq n$), donde cada Y_i satisface una de las siguientes ecuaciones:

$$Y_i = \begin{cases} x_1, x_2, \dots, x_{i-1}, x_i, x_{i+1}, \dots, x_n, x_{i-1} = 0 \\ x_1, x_2, \dots, x_{i-1}, x_i, x_{i+1}, \dots, x_n, x_{i-1} = 1 \end{cases}$$

De acuerdo a la definición anterior, X se conecta a Y_i complementando el bit x_i si $x_{i-1} = 0$ ó complementando todos los bits de x_i, \dots, x_n si $x_{i-1} = 1$.

La relación entre X y Y_1 no está definida, así que podemos suponer que x_0 es igual a 0 o es igual a 1, lo cual nos da diferentes topologías para la red de interconexión.

Si suponemos que $x_0 = 0$, podemos llamar a la gráfica 0-Cubo de Moebius, y si suponemos que $x_0 = 1$, la llamamos 1-Cubo de Moebius, denotadas por $0 - MQ_n$ y $1 - MQ_n$, respectivamente. Las gráficas mostradas en la figura 3.4 son $0 - MQ_4$ y $1 - MQ_4$.

El cubo de Moebius fue propuesto por Cull y Larson [9]. Como el hipercubo Q_n , el de Moebius MQ_n es una gráfica n -regular, n -conexa con 2^n vértices y $n2^{n-1}$ aristas. Además,

$$\text{diam}(MQ_n) = \begin{cases} \lceil \frac{n+2}{2} \rceil & \text{para } 0 - MQ_n, n \geq 4 \\ \frac{n+1}{2} & \text{para } 1 - MQ_n, n \geq 1. \end{cases}$$

Sin embargo, para $n \geq 4$, MQ_n no es transitiva ni en vértices ni en aristas. Cull y Larson primero demostraron la existencia de ciclos hamiltonianos en MQ_n probando que en un hipercubo Q_n , 0-Moebius o 1-Moebius, hay 2^{n-k} ciclos disjuntos de longitud 2^k para cualquier $k \geq 2$. Huang *at el.* [25] y Fan [11], de manera independiente, mostraron que MQ_n es pancíclica para $n \geq 2$.

3.5. Hipercubo balanceado.

Se ha demostrado que la red del hipercubo es una de las redes de interconexión más populares. El hipercubo balanceado, BQ_n , propuesto por Huang and Wu [16] es una de sus variantes. Al igual que el hipercubo, el hipercubo balanceado es una gráfica bipartita y transitiva en vértices. También es una gráfica hamiltoniana.

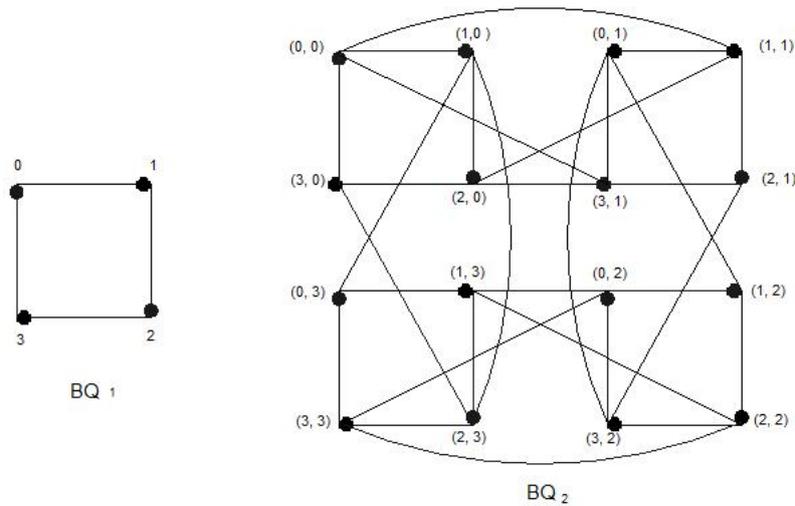


Figura 3.5: Hipercubos balanceados BQ_1 y BQ_2

Definición 3.5. [26] **Hipercubo balanceado** (*Balanced Hypercube*).

El hipercubo balanceado n -dimensional, denotado por BQ_n , tiene 4^n vértices. Cada vértice tiene un único vector con n entradas sobre $\{0, 1, 2, 3\}$.

Un vértice $(a_0, a_1, \dots, a_{n-1})$ conecta a los $2n$ vértices como se muestra:

$$\left\{ \begin{array}{l} ((a_0+1) \bmod 4, a_1, \dots, a_{i-1}, a_i, a_{i+1}, \dots, a_{n-1}), \\ ((a_0-1) \bmod 4, a_1, \dots, a_{i-1}, a_i, a_{i+1}, \dots, a_{n-1}), \end{array} \right\}$$

$$\left\{ \begin{array}{l} ((a_0+1) \bmod 4, a_1, \dots, a_{i-1}, (a_i+(-1)a_0) \bmod 4, a_{i+1}, \dots, a_{n-1}), \\ ((a_0-1) \bmod 4, a_1, \dots, a_{i-1}, (a_i+(-1)a_0) \bmod 4, a_{i+1}, \dots, a_{n-1}) \end{array} \right\}$$

para $1 \leq i \leq n-1$.

La figura 3.5 muestra a BQ_1 y BQ_2 . El hipercubo balanceado BQ_n es una gráfica bipartita $2n$ -regular, transitiva en vértices, con 4^n vértices.

No se conocen otros resultados sobre esta red.

3.6. Gráfica n -estrella

La gráfica n -estrella es una alternativa atractiva a la gráfica del hipercubo, Q_n . Esta tiene ventajas significativas sobre el n -cubo, pues tiene un diámetro de $\lfloor 3(n-1)/2 \rfloor$ que es menor que el diámetro de Q_n ; el número de aristas en S_n es $(n-1)n!/2$, así el grado de los vértices es menor que en Q_n .

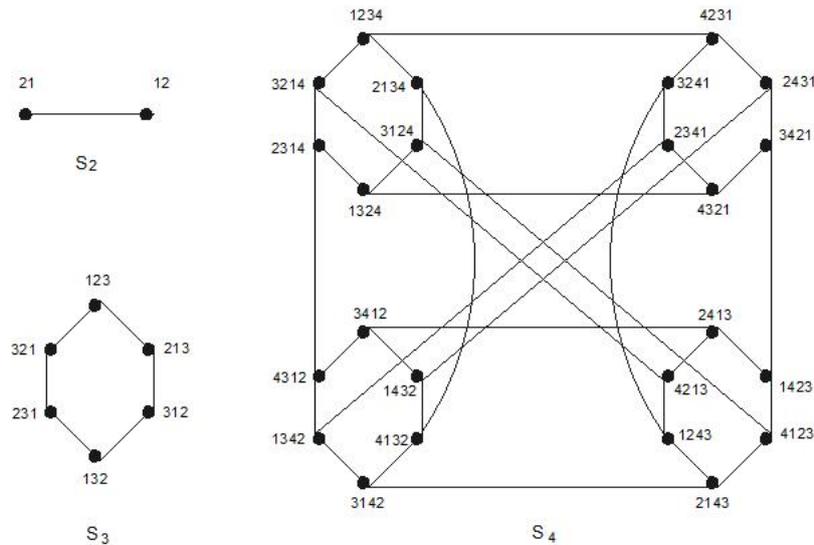


Figura 3.6: Gráficas estrella S_2 , S_3 y S_4

Definición 3.6. [25] **Gráfica n -estrella** (n -Star graph).

La gráfica estrella n -dimensional, denotada por S_n , es no dirigida y consta de $n!$ vértices formados con $n!$ permutaciones sobre un conjunto de $\{1, 2, \dots, n\}$ símbolos. Hay una arista entre cualesquiera dos vértices si y sólo si sus entradas difieren sólo en la primera y cualquier otra posición. Las gráficas mostradas en la figura 3.6 son S_2 , S_3 y S_4 .

Igual que el hipercubo, la gráfica estrella es una gráfica transitiva en vértices y aristas. Además se ha demostrado que S_n es una gráfica de Cayley sobre el grupo de simetría S_n con respecto al conjunto generado $\{t_1, t_2, \dots, t_{n-1}\}$, donde $t_i = (1, i+1)$, ($1 \leq i \leq n-1$), denota una permutación que intercambia dos símbolos en la primera y en la posición $(i+1)$, lo cual implica que S_n es $(n-1)$ -regular. Además, S_n es bipartita y no contiene ciclos de longitud 4.

3.7. (n, k) -estrella

La red de interconexión (n, k) -estrella, fue propuesta en 1995 como una alternativa atractiva a la topología de la n -estrella en la computación paralela. La (n, k) -estrella tiene ventajas significativas sobre la n -estrella que a su vez fue propuesta como una alternativa atractiva para el hipercubo. La principal ventaja de la (n, k) -estrella es su

capacidad de ampliación, lo que hace que sea más flexible que la n -estrella como red de interconexión.

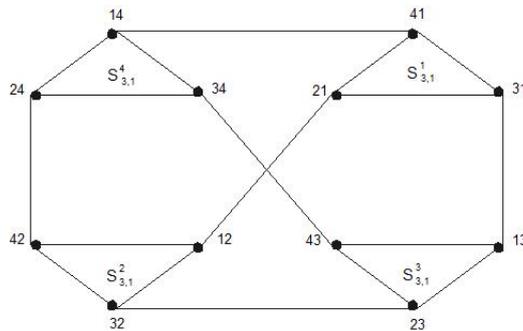


Figura 3.7: Gráfica estrella $S_{4,2}$

Definición 3.7. [25] (n, k) -estrella $((n, k)$ -star).

Hemos visto que el número de vértices para una gráfica n -estrella, S_n , es $n!$ y hay una gran distancia entre $n!$ y $(n+1)!$ para expandir una S_n a una S_{n+1} .

Para remediar esta desventaja, Chiang y Chen [25] propusieron la gráfica (n, k) -estrella, denotada por $S_{n,k}$, cuyo conjunto de vértices

$$\{u_1, u_2, \dots, u_k : u_i \in \{1, 2, \dots, n\}, u_i \neq u_j \text{ para } i \neq j\}.$$

Las adyacencias están definidas de la siguiente manera:

Un vértice $u_1, u_2, \dots, u_i, \dots, u_k$ es adyacente a:

1. el vértice $u_i, u_2, \dots, u_1, \dots, u_k$, donde $2 \leq i \leq k$ (es decir, cambia u_i con u_1),
2. el vértice x, u_2, u_3, \dots, u_k , donde $x \in \{1, 2, \dots, n\} - \{u_i : 1 \leq i \leq k\}$.

La figura 3.7 muestra una $(4, 2)$ -gráfica estrella $S_{4,2}$.

Por definición, $S_{n,n-1}$ es isomorfa a S_n y $S_{n,1}$ es isomorfa a K_n . Así, $S_{n,k}$ es una generalización de S_n . Se tiene que $S_{n,k}$ es una gráfica $(n-1)$ -regular, $(n-1)$ -conexa, transitiva en vértices, con $n!/(n-k)!$ vértices. Además,

$$\text{diam}(S_{n,k}) = \begin{cases} 2k-1, & \text{para } k \leq n/2 \\ \lfloor \frac{(n-1)}{2} \rfloor + k, & \text{para } k \geq \lfloor \frac{n}{2} \rfloor + 1 \end{cases}$$

3.8. (n, k) -cubos

La elección de una red de interconexión es muy importante en el diseño de multiprocesadores de acuerdo a sus propiedades topológicas, algorítmicas y de comunicación.

Una de las redes de interconexión más populares es sin duda el hipercubo Q_n . Sin embargo, todos los vértices de Q_n tiene grado n y, en consecuencia, a medida que n

aumenta también lo hace el grado de cada vértice. Los vértices de mayor grado en las redes de interconexión pueden causar problemas tecnológicos en ordenadores paralelos cuya topología es la de dicha red de interconexión.

Una forma de evadir este problema con el fin de conservar una red de interconexión como el hipercubo Q_n , es la construcción de computadoras paralelas para que la topología subyacente sea el (n, k) -cubo.

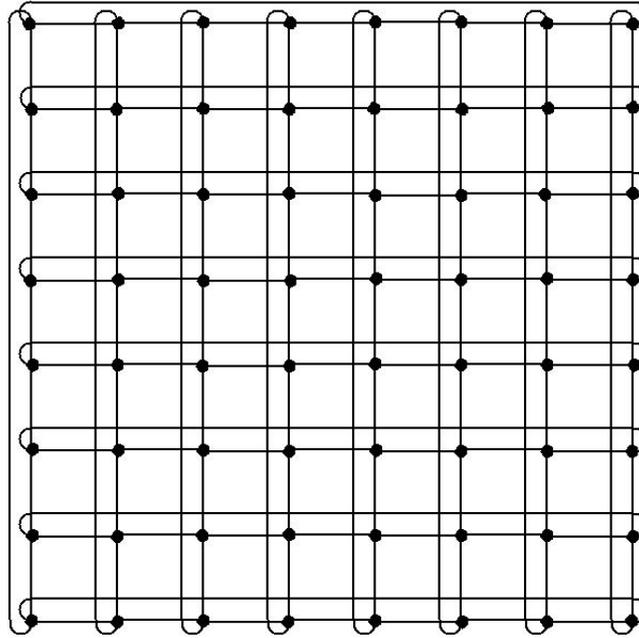


Figura 3.8: Se ilustra la gráfica de Q_2^8

Definición 3.8. [25] (n, k) -cubos (k -ary n -cubes).

La malla toroidal n -dimensional no dirigida, denotada por $Q(k_1, \dots, k_n)$, está definida como el producto cuadro $C_{k_1} \square C_{k_2} \square \dots \square C_{k_n}$, donde C_{k_i} es un ciclo de longitud k_i , $k_i \geq 3$, para cada $i = 1, 2, \dots, n$ y $n \geq 2$. Es claro que $Q(k_1, \dots, k_n)$ tiene k_1, \dots, k_n vértices, tiene $g = \min\{4, k_i, 1 \leq i \leq n\}$, es bipartita si y sólo si k_i es par, para cada $i = 1, 2, \dots, n$. Por propiedades del producto cuadro, obtenemos que $Q(k_1, \dots, k_n)$ es una gráfica de Cayley.

El (n, k) -cubo, Q_n^k , no se ha investigado en la misma medida que el hipercubo, pero se sabe que tiene las siguientes propiedades: es transitivo por vértices y por aristas, es hamiltoniano, tiene una descomposición recursiva y contiene muchas redes de interconexión importantes como ciclos de ciertas longitudes, mallas de ciertas dimensiones e, incluso, hipercubos de ciertas dimensiones. También es $2n$ -regular y $2n$ -conexo; además,

$$\text{diam}(Q_n^k) = \begin{cases} n \lceil k/2 \rceil, & \text{para } n \geq 3 \\ 2 \lceil k/2 \rceil, & \text{para } n = 2. \end{cases}$$

En la figura 3.8 se ilustra la red del hipercubo Q_2^8 , la cual cumple también ser un toro. Podemos observar por simple inspección que es una gráfica 4-regular y 4-conexa.

En los cuadros 3.1 y 3.2 se resumen las propiedades topológicas de las variantes del hipercubo antes definidas, así como de los hipercubos aumentado y plegado que analizaremos con detalle en los siguientes capítulos.

Red	$ V(G) $	$ A(G) $	diam
Q_n	2^n	$n \cdot 2^{n-1}$	n
TQ_n	2^n	$n \cdot 2^{n-1}$	$\lceil (n+1)/2 \rceil$
LTQ_n	2^n	$n \cdot 2^{n-1}$	$n-1$ si $n=3,4$ $\lceil 1/2(n+3) \rceil$ si $n \geq 5$
CQ_n	2^n	$n \cdot 2^{n-1}$	$\lceil (n+1)/2 \rceil$
MQ_n	2^n	$n \cdot 2^{n-1}$	$\lceil (n+2)/2 \rceil$ para $0-MQ_n, n \geq 4$ $(n+1)/2$ para $1-MQ_n, n \geq 1$
BQ_n	4^n	—	$2n-1$ cuando n es impar
S_n	$n!$	$(n-1)n!/2$	$\lfloor 3(n-1)/2 \rfloor$
$S_{n,k}$	$n!/(n-k)!$	$n!(n-1)/2(n-k)!$	$2k-1$ para $1 \leq k \leq \lfloor n/2 \rfloor$ $\lfloor (n-1)/2 \rfloor + k$ para $k \geq \lfloor n/2 \rfloor + 1$
Q_n^k	k^n	nk^n	$n \lfloor k/2 \rfloor, n \geq 3$ $2 \lfloor k/2 \rfloor, n = 2$
AQ_n	2^n	$2^{n-1}(2n-1)$	$\lceil n/2 \rceil$
FQ_n	2^n	$2^{n-1}(n+1)$	$\lceil n/2 \rceil$

Cuadro 3.1: Propiedades de algunas variantes del hipercubo

Red	H	GC	TV	TA	
Q_n	sí	sí	sí	sí	n -regular y n -conexa
TQ_n	sí	no	sí	$n < 4$	n -regular y n -conexa
LTQ_n	$n \geq 3$	—	$n < 4$	—	n -regular y n -conexa
CQ_n	sí	$1 \leq n \leq 4$	$n < 5$	$n < 3$	n -conexa y n -regular
MQ_n	sí	no	$n \leq 3$	$n \leq 2$	n -regular y n -conexa
BQ_n	—	—	sí	—	$2n$ -regular
S_n	no	sí	sí	sí	$(n-1)$ -regular
$S_{n,k}$	$n \geq 3$	sí	sí	sí	$(n-1)$ -regular y $(n-1)$ -conexa
Q_n^k	sí	sí	sí	sí	$2n$ -regular y $2n$ -conexa
AQ_n	sí	sí	sí	sí	$(2n-1)$ -regular y $(2n-1)$ -conexa, $n \neq 3$
FQ_n	sí	sí	sí	sí	$(n+1)$ -regular y $(n+1)$ -conexa

Cuadro 3.2: H:Gráfica hamiltoniana; GC: Gráfica de Cayley; TV: Transitiva en vértices; TA: Transitiva en aristas

Capítulo 4

Hipercubo aumentado

En este capítulo estudiaremos el hipercubo aumentado, describiendo detalladamente algunas de sus propiedades topológicas y algorítmicas.

Siguiendo la definición recursiva del hipercubo Q_n , definimos el hipercubo aumentado, AQ_n . Después mostramos que esta gráfica es simétrica por vértices, $(2n - 1)$ -regular, y $(2n - 1)$ -conexa y que tiene diámetro $\lceil n/2 \rceil$; describimos los algoritmos de enrutamiento y difusión de la información, que son similares a los del hipercubo Q_n . El hipercubo aumentado también es una gráfica de Cayley.

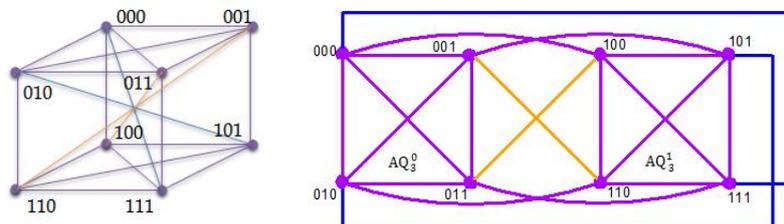


Figura 4.1: Gráfica de AQ_3

Definición 4.1. Hipercubo aumentado (*Augmented Hypercube*) [7]. Sea $n \geq 1$. El cubo aumentado AQ_n de dimensión n tiene 2^n vértices, cada vértice es una n -ada de elementos en $\{0, 1\}$. Definimos $AQ_1 = K_2$.

Para $n \geq 2$, AQ_n se obtiene tomando dos copias del cubo aumentado AQ_{n-1} , denotado por AQ_{n-1}^0 y AQ_{n-1}^1 , y agregando $2 \times 2^{n-1}$ aristas entre las dos copias de la siguiente manera:

$$\text{Sean } V(AQ_{n-1}^0) = \{0, a_2, a_3, \dots, a_n : a_i = 0 \text{ ó } 1\} \text{ y}$$

$$V(AQ_{n-1}^1) = \{1, b_2, b_3, \dots, b_n : b_i = 0 \text{ ó } 1\}.$$

Un vértice $A = 0, a_2, a_3, \dots, a_n$ de AQ_{n-1}^0 es unido a un vértice $B = 1, b_2, b_3, \dots, b_n$ de AQ_{n-1}^1 si y sólo si para toda i , $2 \leq i \leq n$, se cumple (i) $a_i = b_i$, en este caso, AB es una arista del hipercubo, ó (ii) $a_i = \bar{b}_i$, en este caso, AB es una arista complemento. A la construcción recursiva de AQ_n la representamos como $AQ_n = AQ_{n-1}^0 \diamond AQ_{n-1}^1$.

Nótese que AQ_n tiene $(2n - 1)2^{(n-1)}$ aristas. La figura 4.1 muestra dos formas de representar la gráfica del hipercubo aumentado de dimensión 3, AQ_3 .

4.1. Propiedades topológicas básicas

En esta sección citamos algunas de las propiedades topológicas más relevantes que el hipercubo aumentado tiene como gráfica. Demostraremos la mayoría de ellas e ilustraremos para su mejor comprensión algunas otras.

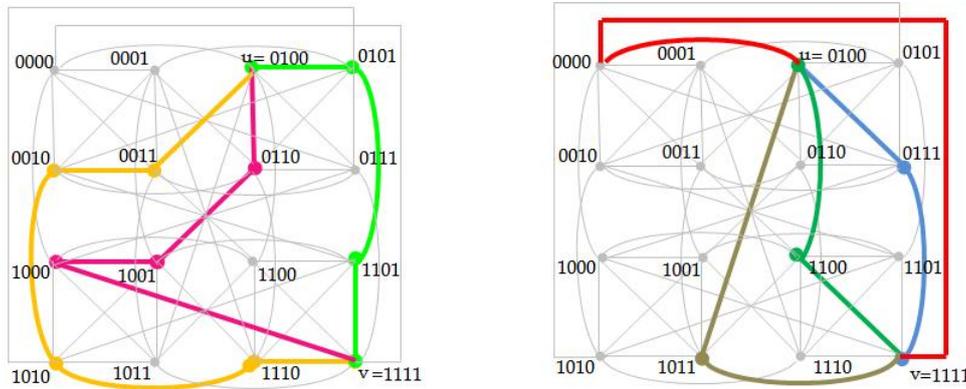


Figura 4.2: Trayectorias internamente disjuntas en AQ_4

Proposición 4.1. Sea $n \geq 4$. Entre cualesquiera dos vértices $x, y \in V(AQ_n)$ existen $2n - 1$ trayectorias internamente disjuntas de x a y .

Demostración. Haremos inducción sobre n .

Para $n = 4$, tendremos $2(4) - 1 = 7$ trayectorias internamente disjuntas, las cuales se verifican directamente. Para ilustrar, considere el vértice $u = 0100$ al vértice $v = 1111$, las siete (u, v) -trayectorias son:

$$P_1 : 0100 - 0011 - 0010 - 1010 - 1110 - 1111,$$

$$P_2 : 0100 - 0110 - 1001 - 1000 - 1111,$$

$$P_3 : 0100 - 0101 - 1101 - 1111,$$

$$P_4 : 0100 - 1011 - 1111,$$

$$P_5 : 0100 - 1100 - 1111,$$

$$P_6 : 0100 - 0111 - 1111,$$

$$P_7 : 0100 - 0000 - 1111.$$

La figura 4.2 muestra estas trayectorias.

Ahora, supongamos que la proposición es válida para $n - 1$, entonces para todo $x, y \in V(AQ_{n-1})$ existen $2n - 3$ trayectorias internamente disjuntas en la gráfica.

Debemos demostrar que la proposición se cumple para $n = k$; es decir, que entre cualesquiera dos vértices $x, y \in V(AQ_k)$ existen $2n - 1$ trayectorias internamente disjuntas.

Sabemos que $AQ_n = AQ_{n-1}^0 \diamond AQ_{n-1}^1$; sin pérdida de generalidad, sea $x \in V(AQ_{n-1}^0)$, con $x = 0x_2, x_3, \dots, x_n = 0A$. En la figura 4.3 ilustramos a la gráfica de AQ_{n-1}^0 . Para hacer más clara la demostración la ilustraremos tomando como caso particular a $n = 4$.

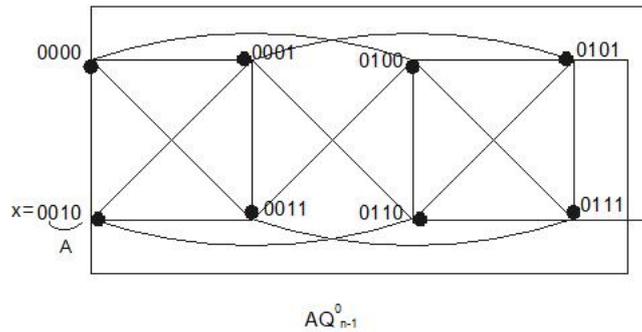


Figura 4.3: Gráfica de AQ_{n-1}^0 , para $n = 4$

Sea $y = y_1, y_2, \dots, y_n = y_1B$, donde $y_1 \in \{0, 1\}$. Sean $A_1 (= \bar{A})$, A_2, \dots, A_{2n-3} vecino de $A \in AQ_{n-1}$ y $B_1 (= \bar{B})$, B_2, \dots, B_{2n-3} vecino de $B \in AQ_{n-1}$. Consideramos dos casos para y_1 y definimos explícitamente las $(2n - 1)$ -trayectorias internamente disjuntas de x a y en una tabla.

CASO 1: $y_1 = 0$, esto es que $x = 0A$ y $y = 0B$.

Lo cual es ilustrado en la figura 4.4.

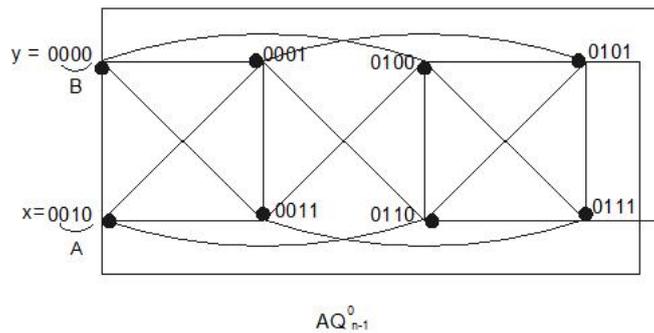


Figura 4.4: Caso donde $y_1 = 0$

Por hipótesis de inducción, existen $2n - 3$ trayectorias internamente disjuntas de A a B en AQ_{n-1} , que las denotaremos con $P_i(A, B)$, para i tal que $1 \leq i \leq 2n - 3$.

Como la construcción del hipercubo aumentado es recursiva, podemos afirmar que $AQ_{n-1}^1 = AQ_{n-2}^{10} \diamond AQ_{n-2}^{11}$; sin pérdida de generalidad, llamaremos $10C$ y $11\bar{C}$ a los vértices de AQ_{n-1}^1 . Específicamente se ilustra en la figura 4.5.

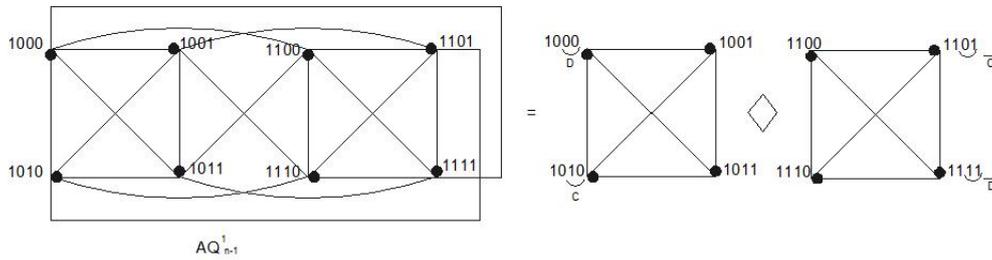


Figura 4.5: Vértices etiquetados conforme a la construcción de la demostración

Ahora vamos a construir la trayectoria de x a y en AQ_{n-1}^0 que pase por $10C$ y $10D$ en AQ_{n-1}^1 , como se ilustra en la figura 4.6, pues estamos agregando una dimensión y , además, la gráfica de AQ_n es $(2n - 1)$ -regular.

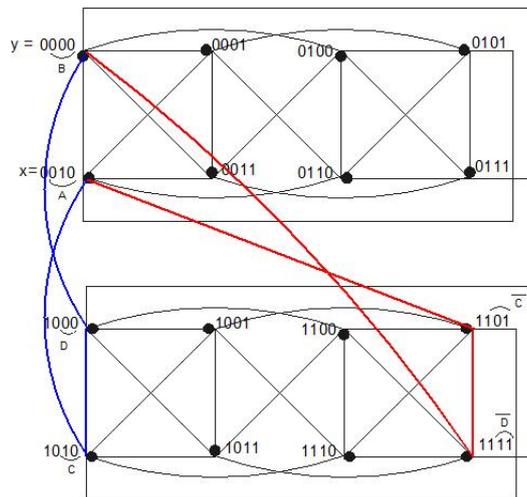


Figura 4.6: Trayectoria de x a y

Entonces, podemos afirmar que tenemos dos salidas hacia AQ_{n-1}^1 y así podemos construir las dos trayectorias que faltan, teniendo las $(2n - 1)$ -trayectorias en AQ_n . Específicamente se muestran en el Cuadro 4.1

Trayectoria	No. Trayectorias
$P_i^0(0A, 0B), 1 \leq i \leq 2n - 3$	$2n - 3$
$0A \rightarrow P^{10}(10C, 10D) \rightarrow 0B$	1
$0A \rightarrow \bar{P}^{11}(10\bar{C}, 11\bar{D}) \rightarrow 0B$	1

Cuadro 4.1: $(2n - 1)$ -Trayectorias

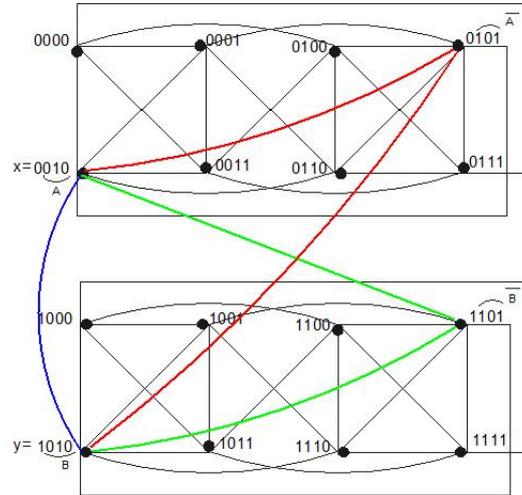


Figura 4.7: Trayectorias de x a y

CASO 2. $y_1 = 1$; es decir, que esté en los vértices de AQ_{n-1}^1 . Esto es, $x = 0A$ y $y = 1B$.

La figura 4.7 muestra este caso. Tenemos dos subcasos:

Caso 2.1. Por definición del hipercubo aumentado tenemos que $B = A$; o bien, $B = \bar{A}$. Mostramos las $(2n - 1)$ -trayectorias disjuntas de x a y en el Cuadro 4.2:

Trayectoria	No. Trayectorias
$0A \rightarrow 0A_i \rightarrow 1B_i \rightarrow 1B, 2 \leq i \leq 2n - 3$	$2n - 4$
$0A \rightarrow 0\bar{A} \rightarrow 1B$	1
$0A \rightarrow 1B$	1
$0A \rightarrow \bar{B} \rightarrow 1B$	1

Cuadro 4.2: Trayectorias

De esta manera, obtenemos las $(2n - 1)$ -trayectorias para este caso.

Caso 2.2. $B \neq A$ ni $B \neq \bar{A}$. Existen otros vértices en AQ_{n-1}^1 que no son adyacentes a $x \in V(AQ_{n-1}^0)$ pero como la gráfica es conexa podemos encontrar la trayectoria entre cualquier par de vértices; así tenemos el caso en el que $B \neq A, \bar{A}$.

Por hipótesis de inducción, existen $(2n - 3)$ -trayectorias internamente disjuntas de A a B en AQ_{n-1} . Sean $P_i(A, B), 1 \leq i \leq 2n - 3$, esas trayectorias. Aquí tenemos dos subcasos:

Caso 2.2.1. Entre las $(2n - 3)$ -trayectorias $P_i(A, B)$, hay una trayectoria, digamos $P_1(A, B)$, que contiene tanto a \bar{A} y \bar{B} como vértices internos.

Las $(2n - 1)$ trayectorias internamente disjuntas de x a y son las que se muestran en el Cuadro 4.3

Trayectoria	No. Trayectorias
$P_i^0(0A, 0B_i) \rightarrow 1B_i \rightarrow 1B, 3 \leq i \leq 2n - 3$	$2n - 5$
$P_1^0(0A, 0\bar{B}) \rightarrow 1B$	1
$P_2^0(0A, 0B) \rightarrow 1B$	1
$0A \rightarrow P_1^1(1\bar{A}, 1B)$	1
$0A \rightarrow P_2^1(1A, 1B)$	1

Cuadro 4.3: Trayectorias

Caso 2.2.2. No existe ninguna trayectoria $P_i(A, B)$, $1 \leq i \leq 2n - 3$ que contenga a \bar{A} y a \bar{B} como vértices internos. Sean $P_1(A, B)$ la trayectoria que sí contiene a \bar{A} y $P_2(A, B)$ la trayectoria que contiene a \bar{B} .

Sea $A_1 = \bar{A}$ vecino de A en P_1 , B_1 vecino de B en P_1 , A_2 vecino de A en P_2 , $B_2 = \bar{B}$ vecino de B en P_2 . Así obtenemos las $(2n - 1)$ -trayectorias internamente disjuntas de x a y . Las cuales se muestran en el Cuadro 4.4.

Trayectoria	No. Trayectorias
$P_i^0(0A, 0B_i) \rightarrow 1B_i \rightarrow 1B, 3 \leq i \leq 2n - 3$	$2n - 5$
$P_1^0(0A, 0B) \rightarrow 1B$	1
$P_2^0(0A, 0\bar{B}) \rightarrow 1B$	1
$0A \rightarrow P_1^1(1\bar{A}, 1B)$	1
$0A \rightarrow P_2^1(1A, 1B)$	1

Cuadro 4.4: Trayectorias

Así hemos demostrado que entre cualquier par de vértices x, y en AQ_n , existen $(2n - 1)$ -trayectorias internamente disjuntas de x a y . \diamond

Otra de las propiedades que tiene el hipercubo aumentado es:

Corolario 4.1. [7] Para toda $n \geq 1$, AQ_n es transitiva por vértices.

El teorema que se presenta a continuación nos muestra que el hipercubo aumentado tiene un diámetro menor que el hipercubo Q_n .

Teorema 4.1. [7] Para el hipercubo aumentado AQ_n se tiene que $diam(AQ_n) = \lceil n/2 \rceil$, para toda $n \geq 1$.

Recordemos que la conexidad por vértices $\kappa(G)$ de una gráfica G es el mínimo número de vértices cuya eliminación da como resultado una gráfica disconexa o la gráfica trivial K_1 . A continuación demostramos la conexidad de AQ_n .

Teorema 4.2. Para el hipercubo aumentado, AQ_n , se tiene que:

$$\kappa(AQ_1) = 1, \kappa(AQ_2) = 3, \kappa(AQ_3) = 4 \text{ y } \kappa(AQ_n) = 2n - 1, \text{ para } n \geq 4.$$

Demostración. Por definición tenemos que $AQ_1 = K_2$ y $AQ_2 = K_4$, por lo que el teorema es verdadero para $n = 1$ y $n = 2$. Para $n = 3$, podemos verificar fácilmente que $\kappa(AQ_3) = 4$. En la figura 4.8 se ilustra este hecho.

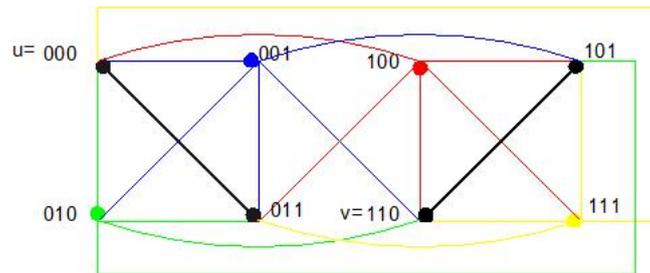


Figura 4.8: $\kappa(AQ_3) = 4$

Sabemos por la Proposición 4.1 que entre cualesquiera dos vértices $x, y \in V(AQ_n)$ existen $(2n - 1)$ -trayectorias internamente disjuntas de x a y , así que aplicando el Teorema de Menger podemos concluir que $\kappa(AQ_n) = 2n - 1$. \diamond

4.2. Subgráficas de AQ_n

La topología de redes y sus propiedades de inmersión son temas importantes en el diseño de redes de interconexión. En esta sección describiremos algunas subgráficas de AQ_n que facilitan la comunicación entre nodos.

Al primer teorema que presentamos en esta sección, lo ilustramos con la gráfica de AQ_3 que se muestra en la figura 4.9, la cual tiene 6 ciclos cuyas longitudes son 3, 4, 5, 6, 7 y 8, respectivamente.

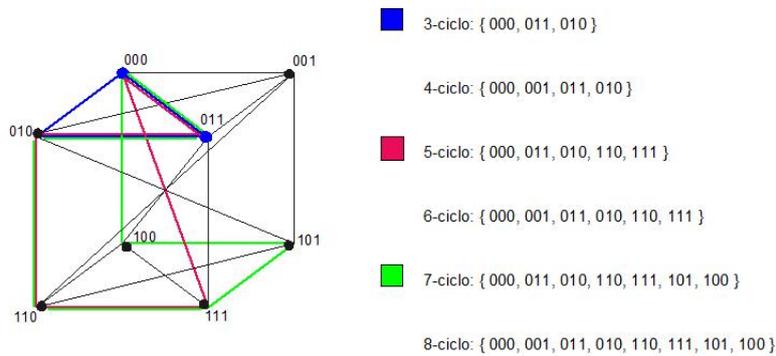


Figura 4.9: Se muestran 6 ciclos en AQ_3

Teorema 4.3. El hipercubo aumentado AQ_n , para toda $n \geq 2$, contiene un k -ciclo para toda k , $3 \leq k \leq 2^n$.

Demostración. El hipercubo Q_n y el hipercubo aumentado AQ_n contienen un $2l$ -ciclo para cada l , $2 \leq l \leq 2^{n-1}$.

Una técnica para construir tal ciclo es similar a la del código Gray. Después de generar $(n-1) \cdot 2^{n-1}$ -sucesiones de elementos del código Gray, escogemos el primer l elemento de esta sucesión, escribimos los mismos elementos en orden inverso y agregamos un 0 a los primeros l elementos y agregamos un 1 a cada uno de los últimos l elementos. Esta sucesión de $2l$ -elementos induce un $2l$ -ciclo, C_{2l} , en Q_n .

Los primeros tres elementos en esta sucesión son 0^n , $0^{n-1}1$ y $0^{n-2}11$. Observemos que en AQ_n , 0^n y $0^{n-2}11$ son adyacentes, así $C_{2l} - \{0^{n-1}1\}$ es un $(2l-1)$ -ciclo en AQ_n . \diamond

Por ejemplo, para ilustrar el teorema, el ciclo de longitud 8 para la gráfica de la figura 4.9 queda definido por: $000 \rightarrow 001 \rightarrow 011 \rightarrow 010 \rightarrow 110 \rightarrow 111 \rightarrow 101 \rightarrow 100$.

Teorema 4.4. [7] El hipercubo aumentado AQ_n , para $n \geq 3$ contiene dos árboles binarios completos ajenos por aristas sobre $2^n - 1$ vértices y cuya raíz está en 0^n .

La figura 4.10 presenta al hipercubo aumentado AQ_3 y a sus dos árboles binarios completos ajenos por aristas.

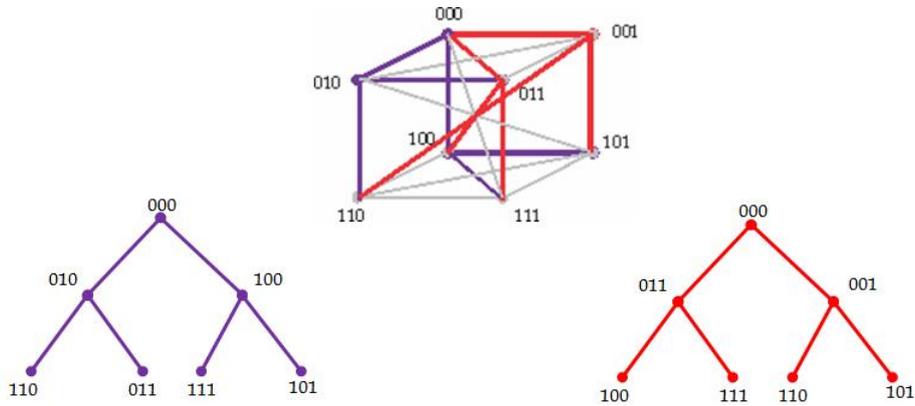


Figura 4.10: Árboles binarios ajenos por aristas en AQ_3

Teorema 4.5. Sea $n \geq 3$, el hipercubo aumentado AQ_n contiene $n - 1$ árboles generadores ajenos por aristas $(S_1, S_2, \dots, S_{n-1})$ y con las aristas restantes se forma un árbol (R_n) .

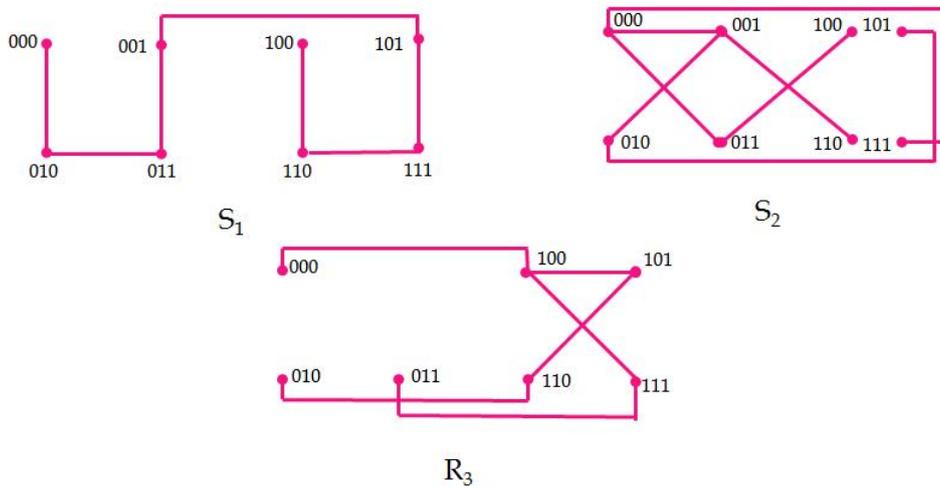


Figura 4.11: Los árboles S_1, S_2 y R_3 contenidos en AQ_3

Demostración. La prueba es por inducción sobre n , la dimensión. Primero, observemos que el árbol R_n que se forma con las aristas restantes debe tener

$$(2n - 1) \cdot 2^{n-1} - (n - 1) \cdot (2^n - 1) = 2^{n-1} + n - 1 \text{ aristas.}$$

Para $n = 3$, AQ_3 contiene dos árboles generadores ajenos por aristas S_1 y S_2 ; con las aristas que sobran se forma el árbol R_3 . Estos árboles se muestran en la figura 4.11.

Supongamos que el teorema es válido para AQ_n y sabemos que $AQ_{n+1} = AQ_n^0 \diamond AQ_n^1$.

Sean $S_1^i, S_2^i, \dots, S_{n-1}^i, R_n^i$ los árboles contenidos en $AQ_n^i \simeq AQ_n$ ($i = 0, 1$). Denotamos por $v_1^i, v_2^i, \dots, v_r^i, r = 2^{n-1} + n$, a los vértices de $R_n^i, i = 0, 1$.

Construimos los árboles requeridos $S_1, S_2, \dots, S_n, R_{n+1}$ contenidos en AQ_{n+1} de la siguiente manera:

Para $1 \leq i \leq n-2$, S_i es el árbol que se obtiene al unir S_i^0 y S_i^1 con la arista del hipercubo (v_i^0, v_i^1) . S_{n-1} es el árbol generado por las aristas de S_{n-1}^0 y el complemento de todas las aristas entre AQ_n^0 y AQ_n^1 . S_n se construye tomando S_{n-1}^1, R_n^0 , la arista (v_{n-1}^0, v_{n-1}^1) , y el conjunto de todas las aristas del hipercubo entre $V(AQ_n^0) - V(R_n^0)$ y $V(AQ_n^1) - V(R_n^1)$.

Hasta ahora, hemos usado todas las aristas de $AQ_n^0 \cup (AQ_n^1 - A(R_n^1))$ y todas las aristas entre AQ_n^0 y AQ_{n-1} excepto las aristas $(v_i^0, v_i^1), n \leq i \leq r$. R_{n+1} es el árbol generado por las aristas de R_n^1 y todas las aristas del hipercubo $(v_i^0, v_i^1), n \leq i \leq r$. Notemos que R_{n+1} contiene $(2^{n-1} + n - 1) + (r - (n - 1)) = 2^n + n$ aristas, como se requiere. \diamond

4.3. Algoritmo de enrutamiento

El algoritmo de enrutamiento para el hipercubo aumentado AQ_n se describe a continuación. Nótese que en esta variante el envío de información es a través de la ruta más corta entre cualquier par de nodos.

Teorema 4.6. [7] Sean $x, y \in V(AQ_n) = V(AQ_{n-1}^0) \diamond V(AQ_{n-1}^1)$.

(1) Si $x, y \in V(AQ_{n-1}^0)$ ó $x, y \in V(AQ_{n-1}^1)$, existe una ruta más corta de x a y en AQ_n con todos sus vértices en AQ_{n-1}^0 y análogamente para AQ_{n-1}^1 .

La figura 4.12 ilustra este enunciado.

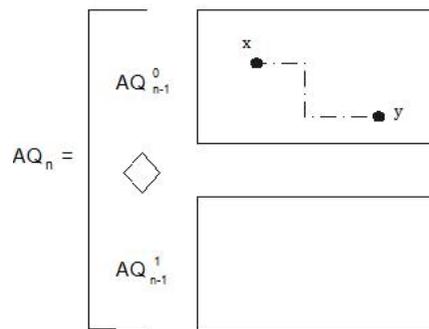


Figura 4.12: Inciso (1) del teorema 4.6

(2) Sean $x \in V(AQ_{n-1}^0)$ y $y \in V(AQ_{n-1}^1)$. Entonces,

(i) Existe una (x, y) -ruta más corta T , en AQ_n con todos sus vértices, excepto x , en AQ_{n-1}^1 . Además, el segundo vértice de T , el vecino de x , está definido por $(1, x_2, x_3, \dots, x_n)$; o bien, por $(1\bar{x}_2, \overline{x_3, \dots, x_n})$ de acuerdo a si:

$$d(x_2, x_3, \dots, x_n, y_2, y_3, \dots, y_n) \leq d(\bar{x}_2, \overline{x_3, \dots, x_n}, y_2, y_3, \dots, y_n);$$

o bien,

$$d(\overline{x_2, x_3, \dots, x_n}, y_2, y_3, \dots, y_n) \leq d(\bar{x}_2, \overline{x_3, \dots, x_n}, y_2, y_3, \dots, y_n).$$

Gráficamente se representa en la figura 4.13.

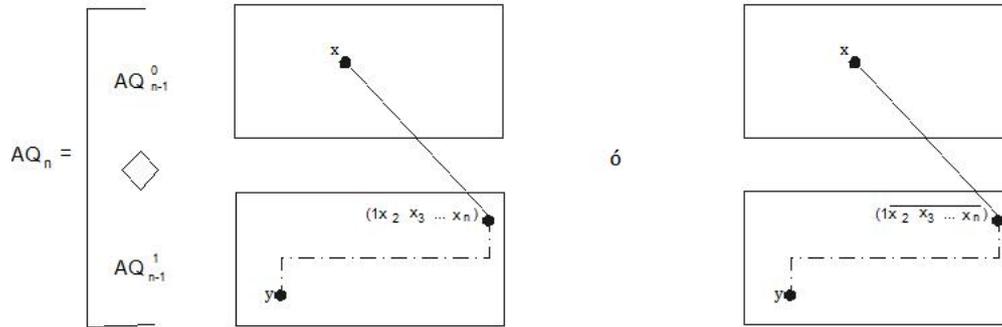


Figura 4.13: Inciso (2. i) del teorema 4.6

(ii) Existe una (x, y) -ruta más corta T en AQ_n , con todos sus vértices, excepto y , en AQ_{n-1}^0 . Además el penúltimo vértice de T , es decir el vecino de y en T , está definido por $(0, y_2, y_3, \dots, y_n)$; o bien, por $(0, \bar{y}_2, y_3, \dots, y_n)$. Se ilustra en la figura 4.14.

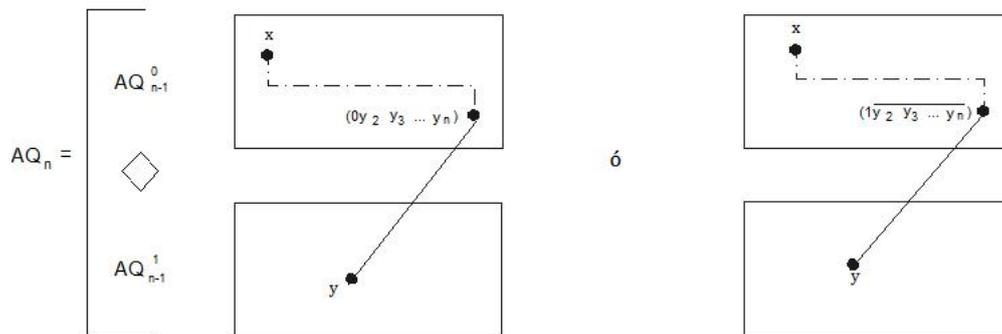


Figura 4.14: Inciso (2. ii) del teorema 4.6

Utilizaremos el teorema anterior para enviar un mensaje desde un vértice S (origen), a un vértice D (destino), a través de la (S, D) -ruta más corta.

Mientras el mensaje viaja a lo largo de la trayectoria, cualquier vértice activo C efectúa tres tareas:

1. Calcula $\text{tag}(C \oplus_2 D) = (c_1 \oplus_2 d_1, c_2 \oplus_2 d_2, \dots, c_n \oplus_2 d_n)$.
 2. Busca $\text{tag}(C \oplus_2 D)$ para el menor índice i tal que $c_i \oplus_2 d_i = 1$.
 3. Si $c_{i+1} \oplus_2 d_{i+1} = 0$ entonces,
 - cambia la i -ésima entrada de C a d_i ;
 - envía el mensaje al siguiente vértice activo $C' = (d_1, \dots, d_i, c_{i+1}, \dots, c_n)$ a lo largo de las aristas de peso $2i - 1$;
- en otro caso,
- cambia la i -ésima entrada de C a d_i ;

envía el mensaje al siguiente vértice activo $C' = (d_1, d_2, \dots, d_i, \overline{c_{i+1}}, c_{i+2}, \dots, c_n)$ a lo largo del complemento de la arista de peso $2i$.

Observemos las ilustraciones en la figura 4.15. Formalmente, el procedimiento se describe como lo muestra el Algoritmo 4.1

Algorithm 4.1: Algoritmo de enrutamiento

Data: Envía un mensaje desde un vértice S (origen) a un vértice D (destino) a través de la (S, D) -trayectoria más corta; $tag \leftarrow S \oplus_2 D$.

```

1 if  $tag = 00\dots 0$  then
2   | STOP //El mensaje ha llegado a  $D$ .
3 else
4   | revisar la función  $tag$  de izquierda a derecha hasta encontrar un bit con valor 1,
   | digamos que es el  $i$ .
5 if  $tag(i+1) = 0$  then
6   |  $tag(i) \leftarrow 0$ ,
7   | se transmite el mensaje a través de las aristas de peso  $2i - 1$ .
8 else
9   |  $tag(i) \leftarrow 0$ ;
10  |  $tag(j) \leftarrow \overline{tag(j)}$ ,  $i < j \leq n$ ;
11  | transmite el mensaje a través de las aristas de peso  $2i$ .
  
```

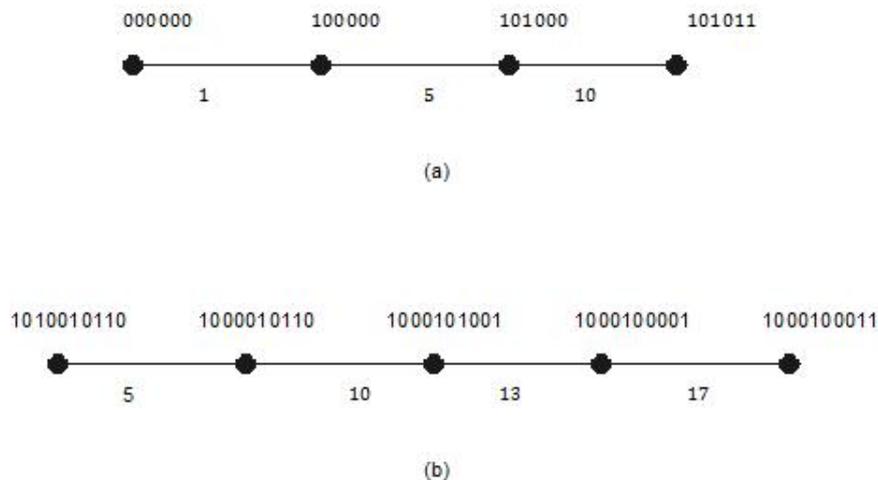


Figura 4.15: Trayectorias de enrutamiento (a) de 000000 a 101011 en AQ_6 y (b) de 1010010110 a 1000100011 en AQ_{10}

4.4. Difusión de la información

El algoritmo de difusión de la información para cubos aumentados es similar al algoritmo para hipercubos. Como AQ_n es una gráfica transitiva por vértices, es suficiente describir el algoritmo con origen en el $0^n = 0, 0, \dots, 0$ (n veces).

Algorithm 4.2: Algoritmo de difusión de la información

Data: Envía el mensaje desde 0^n a través de todas las aristas incidentes en él.

- 1 Etiquetamos todos los nodos y aristas cuyo vértice inicial es 0^n .
 - 2 **while** exista un vértice $x = 0^k A$ con $2 \leq k < n$, $x_{k+1} = 1$ e ingrado 1 **do**
 - 3 **in parallel:** Envía el mensaje desde x a través de las aristas que no están etiquetadas de peso a lo más $2k - 2$ incidentes en él.
-

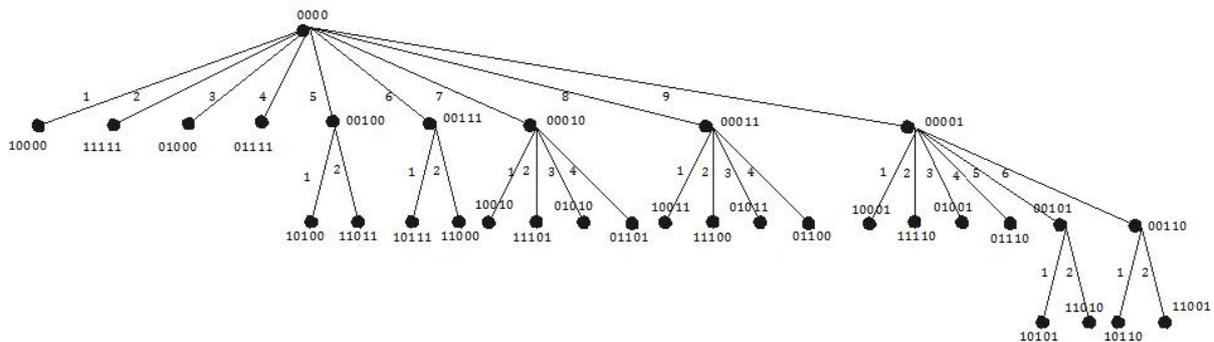


Figura 4.16: Árbol de difusión de AQ_5 con raíz en 00000

El algoritmo se ilustra en la figura 4.16. Éste toma $\lceil \frac{n}{2} \rceil$ pasos para transmitir el mensaje a todos los vértices y al mismo tiempo. En comparación con el hipercubo cruzado y el hipercubo aumentado, en el hipercubo aumentado el número de vértices que reciben el mensaje es mayor. Estas propiedades del algoritmo 4.2 se muestran en el siguiente resultado:

Teorema 4.7. La subgráfica generada por el algoritmo 4.2 es un árbol generador T de AQ_n con $d(0^n, x; T) = d(0^n, x; AQ_n)$, para toda $x \in V(AQ_n)$.

Demostración. Podemos observar que el árbol T es una subgráfica generadora conexa de AQ_n . Para mostrar que es un árbol, sumamos los grados de sus vértices y llegamos a que tiene $2^n - 1$ aristas. \diamond

Nota: En el algoritmo 4.2, la sentencia "do in paralel" se refiere a realizar la instrucción de la línea 3, simultáneamente para todos los valores de k dados.

Capítulo 5

Hipercubo plegado

Recientemente, se han propuesto diversas variantes del hipercubo para mejorar su rendimiento y confiabilidad. El hipercubo Plegado es una variante del hipercubo que mejora su rendimiento y confiabilidad, en el que se añade una arista por cada nodo, proporcionando una conexión directa con el nodo más alejado a él.

Definición 5.1. Hipercubos plegados (Folded Hypercube) [25]

El hipercubo plegado n -dimensional, denotado por FQ_n , es una gráfica obtenida a partir de Q_n agregando todas las aristas complementarias, las cuales unen a un vértice $x = x_1, x_2, \dots, x_n$ con otro vértice $\bar{x} = \bar{x}_1, \bar{x}_2, \dots, \bar{x}_n$ para toda $x \in V(Q_n)$, donde $\bar{x}_i = 1 - x_i$.

El hipercubo plegado FQ_n , propuesto por El-Amawy y Latifi[20], es una gráfica $(n + 1)$ -regular, $(n + 1)$ -conexa con $2n$ vértices. Como Q_n , FQ_n es una gráfica de Cayley y por ello es transitiva en vértices. Sin embargo, FQ_n tiene diámetro de $\lceil \frac{n}{2} \rceil$, que es menor al diámetro de Q_n . Xu y Ma [24] mostraron que FQ_n es bipartita si y sólo si n es impar; la longitud del ciclo impar más corto es $n + 1$ si n es par.

Las gráficas que mostramos en la figura 5.9 son FQ_3 y FQ_4 , respectivamente.

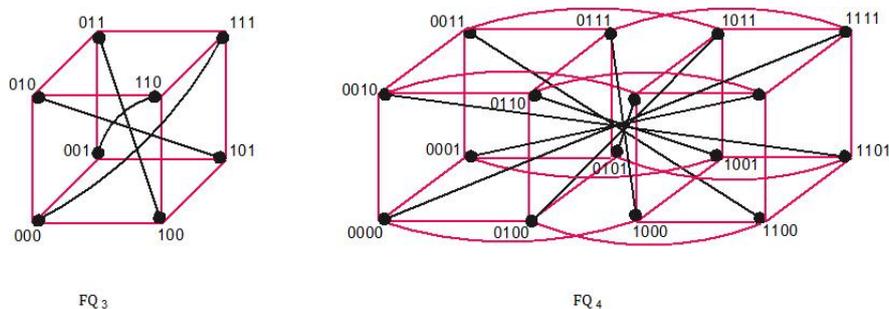


Figura 5.1: FQ_n para $n = 3$ y 4

5.1. Propiedades topológicas básicas

Comenzamos con una lista de propiedades importantes del hipercono plegado que han sido reportadas en los últimos años.

Cada nodo etiquetado, u , en FQ_n , está conectado a n nodos a distancia de Hamming 1 y a un nodo a distancia de Hamming n . Este último nodo se alcanza a través de una arista y su etiqueta se obtiene al complementar todos los n bits de u . Además, $d_F(u) = n + 1$ y F es regular de grado $n + 1$. Puesto que cada nodo en FQ_n tiene $n + 1$ vecinos, es necesario eliminar al menos $n + 1$ nodos para desconectar FQ_n . De esta manera, tenemos el siguiente teorema:

Teorema 5.1. [10] La conexidad de FQ_n es $n + 1$.

En la gráfica de la figura 5.2 ilustramos este teorema, donde (a) es FQ_3 y (b) FQ_3 menos los cuatro vértices adyacentes al vértice (110), los cuales están en color blanco. Observemos que al eliminar los cuatro vértices adyacentes a (110) la gráfica se desconecta, quedando solamente los vértices coloreados en negro. De esta manera, la conexidad de FQ_3 es 4.

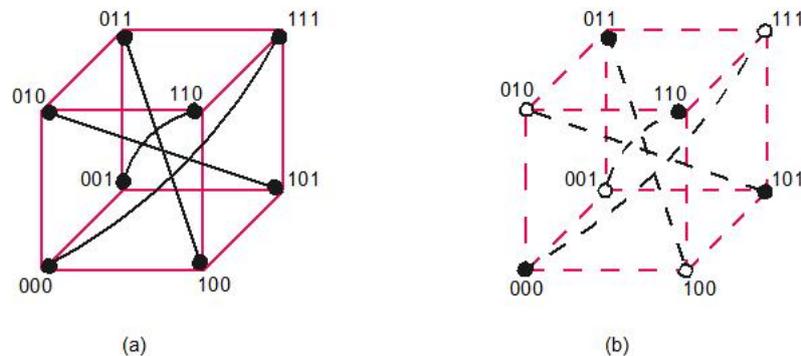


Figura 5.2: (a): FQ_3 , (b): $FQ_3 - \{(010), (001), (100), (111)\}$

Como resultado del Teorema 5.1, tenemos los siguientes lemas:

Lema 5.1. El número de trayectorias internamente disjuntas entre cualesquiera dos nodos en FQ_n es $n + 1$.

Demostración. La demostración de este lema resulta directamente del Teorema de Menger, el cual establece que en una gráfica G con conexidad n , existen exactamente n trayectorias internamente disjuntas entre cualquier par de vértices en G . \diamond

Mostramos las cinco trayectorias internamente disjuntas del vértice u a el vértice v en FQ_4 en la figura 5.3.

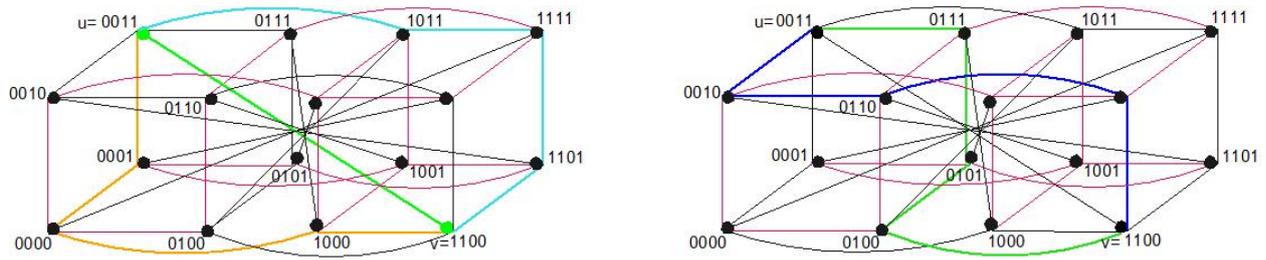


Figura 5.3: Cinco trayectorias internamente disjuntas en FQ_4

Establecemos el diámetro de FQ_n en el siguiente teorema:

Teorema 5.2. [10] El diámetro de FQ_n es $\left\lceil \frac{n}{2} \right\rceil$.

Señalamos en la figura 5.4 con una trayectoria morada el diámetro de FQ_4 que es 2. Recordemos que el diámetro es la longitud de la trayectoria más larga entre cualquier par de vértices.

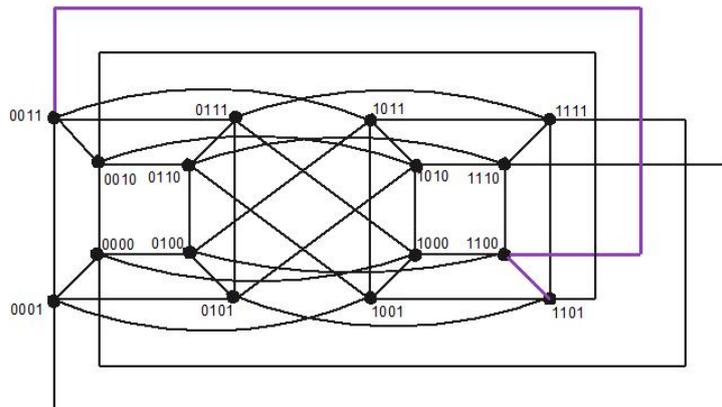


Figura 5.4: Diámetro en FQ_4

De acuerdo a la definición del hipercubo plegado, es fácil ver que cualquier hipercubo plegado de dimensión $(n + 1)$, FQ_{n+1} , puede verse como $G(Q_n^0, Q_n^1; M_0 \cup M)$, donde Q_n^0 y Q_n^1 son dos hipercubos de dimensión n con los prefijos 1 y 0 de cada vértice, respectivamente, y $M_0 = \{0u1u \mid 0u \in V(Q_n^0) \text{ y } 1u \in V(Q_n^1)\}$, $M = \{0u1u \mid 0u \in V(Q_n^0) \text{ y } 1u \in V(Q_n^1)\}$.

Para una cadena binaria u con n -bits, usamos u_i para denotar la cadena binaria que difiere en la i -ésima posición con u . Similarmente, usamos u_{ij} para denotar la cadena binaria con n -bits la cual difiere en la j -ésima posición con u_i . Claramente, $u_{ii} = u$. Usamos \bar{u} para denotar la cadena binaria con n -bits la cual difiere con u en cada uno de los bits. Usamos $\bar{e}(u) \in \bar{M}$ para denotar la arista en \bar{M} incidente con u , y $e_i(u)$ para denotar la arista (u, u_i) , para $i \in \{1, 2, \dots, n\}$.

En los siguientes resultados, usamos $0u$ para denotar un vértice de FQ_{n+1} , lo cual significa que $0u \in V(Q_n^0)$. Similarmente, escribimos $1u$, $1u \in V(Q_n^1)$. Además por conveniencia, consideramos FQ_{n+1} en lugar de FQ_n .

Lema 5.2. Si cualesquiera dos vértices u, v en $V(FQ_{n+1})$ tienen vecinos en común, éstos son exactamente dos, para $n \geq 3$.

Demostración. Tenemos dos casos. Sean $u, v \in V(FQ_{n+1})$.

CASO 1: Ambos vértices están en $V(Q_n^0)$ o en $V(Q_n^1)$.

Sin pérdida de generalidad, consideremos a $u, v \in V(Q_n^0)$. Supongamos que los dos vértices son $0u$ y $0v$, y que $0u$ y $0v$ tienen dos vecinos en común en Q_n^0 .

Por definición de Q_n^0 , $0u$ y $0v$ difieren en exactamente dos posiciones. Entonces $\{1u, 1\bar{u}\} \cap \{1v, 1\bar{v}\} = \varphi$ para $n \geq 3$. Así $0u$ y $0v$ no tienen vecinos en común en Q_n^1 pero tienen exactamente dos vecinos en común en FQ_{n+1} .

Supongamos que $0u$ y $0v$ no tienen vecinos en común en $V(Q_n^0)$. Si $u = \bar{v}$ entonces $\{1u, 1\bar{u}\} = \{1v, 1\bar{v}\}$. Entonces $0u$ y $0v$ tienen exactamente dos vecinos en común en FQ_{n+1} . Si $u \neq \bar{v}$, entonces $\{1u, 1\bar{u}\} \cap \{1v, 1\bar{v}\} = \phi$. Así $0u$ y $0v$ no tienen vecinos en común en FQ_{n+1} .

CASO 2: Uno de los vértices está en $V(Q_n^0)$ y el otro está en $V(Q_n^1)$.

Sin pérdida de generalidad, supongamos que $0u \in V(Q_n^0)$ y $1v \in V(Q_n^1)$. Si existe una i tal que $v \in \{u_i, \bar{u}_i\}$, entonces

$$|N_{FQ_{n+1}}(0u) \cap N_{FQ_{n+1}}(1v)| = |\{0u_1, 0u_2, \dots, 0u_n, 1u, 1\bar{u}\} \cap \{1v_1, 1v_2, \dots, 1v_n, 0\bar{v}\}| = 2.$$

Así $0u$ y $1v$ tienen exactamente dos vecinos en común en FQ_{n+1} .

Si $v \notin \{u_i, \bar{u}_i\}$ para cualquier $i \in \{1, 2, \dots, n\}$, entonces $u \notin \{v_i, \bar{v}_i \mid 1 \leq i \leq n\}$. Así,

$$|N_{FQ_{n+1}}(0u) \cap N_{FQ_{n+1}}(1v)| = |\{0u_1, 0u_2, \dots, 0u_n, 1u, 1\bar{u}\} \cap \{1v_1, 1v_2, \dots, 1v_n, 0v, 0\bar{v}\}| = 0,$$

lo cual implica que $0u$ y $1v$ no tienen vecinos en común en FQ_{n+1} . \diamond

Usaremos $g(G)$ para denotar el **cuello** de una gráfica G , es decir, la longitud mínima de entre todos los ciclos en G .

Lema 5.3. $g(FQ_{n+1}) = 4$ para $n \geq 2$

Demostración. FQ_{n+1} puede verse como $G(Q_n^1, Q_n^2; M_0 + \bar{M})$ y $g(Q_n) = 4$ para $n \geq 2$, sólo necesitamos demostrar que cualquier arista en M_0 o en \bar{M} no está contenida en un triángulo.

Sea $e_0 = (0u, 1u)$ una arista en M_0 . Entonces e_0 no está contenida en un triángulo pues $1u$ no es adyacente ni a $1\bar{u}$ para $n \geq 2$ ni a $0u_i$ para $i \in \{1, 2, \dots, n\}$. Simultáneamente cualquier arista $\bar{e} = (0u, 1\bar{u})$ no está contenida en un triángulo. \diamond

5.2. Subgráficas del hipercubo plegado

Un vértice $u \in V(Q_n)$ se dice que es par o impar si la suma de sus entradas es par o impar. Sea $X = \{u : u \text{ es impar}\}$ e $Y = \{u : u \text{ es par}\}$. Entonces $\{X, Y\}$ es una bipartición de Q_n ; es claro que si dos vértices en Q_n están en el mismo conjunto de la bipartición tienen la misma paridad.

Teorema 5.3. La gráfica de FQ_n es bipartita si y sólo si n es impar.

Demostración. Dado que FQ_n se obtiene a partir de Q_n añadiendo 2^{n-1} aristas complementarias, para demostrar el teorema, es suficiente considerar las aristas complementarias. Sea $\{X, Y\}$ una bipartición de Q_n . Dado que cualquier vértice u y su complemento \bar{u} en Q_n tienen distinta paridad si y sólo si n es impar y, por lo tanto, cualquier arista complementaria en FQ_n une dos vértices que pertenecen a diferentes particiones de $\{X, Y\}$, se deduce que FQ_n es una gráfica bipartita si y sólo si n es impar. \diamond

Teorema 5.4. La longitud de cualquier ciclo en FQ_n que contiene exactamente dos aristas complementarias es par.

Demostración. Sea $C = \{v, \bar{v}, v_1, \dots, v_m, \bar{u}, u_1, \dots, u_n, v\}$ un ciclo en FQ_n que contiene exactamente dos aristas complementarias $e_1 = (u, \bar{u})$ y $e_2 = (v, \bar{v})$. Claramente, $v \neq u$ y, así $\bar{v} \neq \bar{u}$ por lo que $e_1 \neq e_2$.

Si FQ_n es bipartita, entonces terminamos la demostración. Supongamos ahora que FQ_n no es bipartita. Por el Teorema 5.3, n es par y, por lo tanto, u y \bar{u} tienen la misma paridad.

Sea $\{X, Y\}$ una bipartición de Q_n . Entonces, u y \bar{u} están en el mismo conjunto de la bipartición $\{X, Y\}$. Análogamente para v y \bar{v} . Puesto que C contiene exactamente dos aristas complementarias, las otras aristas en C están en Q_n .

Si u, \bar{u}, v, \bar{v} están en la misma parte, entonces la longitud de las rutas: $\{v, v_1, \dots, v_m, u\}$ y $\{u, u_1, \dots, u_n, v\}$ son pares.

Si u, \bar{u}, v, \bar{v} no están en la misma parte, las longitudes de las dos rutas son impares. De ello se deduce que la longitud de C es impar en ambos casos. \diamond

Teorema 5.5. La longitud de cualquier ciclo en FQ_n que contiene exactamente una arista complementaria es al menos $n + 1$. Además cualquier arista complementaria y cualquier vértice sobre un ciclo en común de longitud $n + 1$ contiene la única arista complementaria en FQ_n .

Demostración. Suponemos que $e = (u, \bar{u})$ es una arista complementaria en FQ_n y que C es un ciclo en FQ_n que contiene exactamente una arista complementaria e . Sea $P = C - e$ una (u, \bar{u}) -trayectoria en Q_n . Claramente $d_{Q_n}(u, \bar{u}) = n$ y así, la

longitud de cualquier (u, \bar{u}) -trayectoria en Q_n es al menos n . Así la longitud de P es al menos n y la longitud de C es $n + 1$.

Sea P' la (u, \bar{u}) -trayectoria más corta en Q_n y sea v cualquier vértice en FQ_n . Si $v \in \{u, \bar{u}\}$, entonces $P' + e$ es un ciclo de longitud $n + 1$ y contiene la única arista complementaria e en FQ_n . Ahora, suponemos que $v \notin \{u, \bar{u}\}$. Como $d_H(u, v) + d_H(\bar{u}, v) = n$ hay una $R = (v, u)$ -trayectoria más corta y una $R' = (v, \bar{u})$ -trayectoria más corta en Q_n . Entonces la suma de sus longitudes es n . Observemos que $R \cup R' + (u, \bar{u})$ es un ciclo de longitud $n + 1$ y contiene el vértice v y la única arista complementaria.

◇

Lema 5.4. [25] Si $n \geq 2$, entonces

- (a) para cualesquiera dos vértices $x, y \in V(Q_n)$ existe una (x, y) -trayectoria de longitud ℓ con $d_{Q_n}(x, y) \leq \ell \leq 2^{n-1}$, donde ℓ y $d_{Q_n}(x, y)$ tienen la misma paridad;
- (b) toda arista de Q_n pertenece a un ciclo de cualquier longitud par de 4 a 2^n .

Teorema 5.6. Toda arista complementaria en FQ_n se encuentra sobre un ciclo de longitud par desde 4 hasta 2^n para $n \geq 2$.

Demostración. Sea e una arista complementaria en FQ_n . Sin pérdida de generalidad, suponemos que $e = (0u, 1\bar{u})$ con $0u \in V(Q_{n-1}^0)$ y $1\bar{u} \in V(Q_{n-1}^1)$. Escogemos una arista $(0u, 0v)$ en Q_{n-1}^0 y la arista correspondiente $(1\bar{u}, 1\bar{v})$ en Q_{n-1}^1 . Queremos demostrar que la arista $e = (0u, 1\bar{u})$ pertenece a un ciclo de longitud ℓ en FQ_n con $4 \leq \ell \leq 2^n$ y ℓ par.

Podemos expresar a $\ell = \ell' + \ell'' + 2$ donde $1 \leq \ell' \leq 2^{n-1} - 1$, $1 \leq \ell'' \leq 2^{n-1} - 1$ y tanto ℓ' como ℓ'' son enteros impares. Por el lema 5.4(a) existe una trayectoria P' de longitud ℓ' en Q_{n-1}^0 que une $0u$ con $0v$ y una trayectoria P'' de longitud ℓ'' en Q_{n-1}^1 que une $1\bar{u}$ con $1\bar{v}$. Entonces $P' + (0v, 1\bar{v}) + P'' + (1\bar{v}, 0u)$ es un ciclo que contiene a la arista e en FQ_n , cuya longitud es igual a ℓ . ◇

Recordemos que un árbol T en una gráfica G es un árbol generador si T contiene a todos los vértices de G . Un árbol enraizado es un árbol con un vértice $r \in V(T)$, al cual llamaremos vértice raíz. Un conjunto de $k, k \geq 2$, árboles generadores enraizados en el mismo vértice sobre una gráfica se dice que son ajenos por aristas si cualesquiera dos árboles de este conjunto no tienen aristas en común. El problema de construir árboles generadores ajenos por aristas ha sido estudiado para varias redes.

Los árboles generadores independientes tienen propiedades más fuertes que los árboles generadores ajenos por aristas. Sea T un árbol generador de una gráfica G con raíz en el vértice r . Para un vértice dado $x \in V(T) \setminus \{r\}$, $P(T, x)$ denota la única trayectoria desde r a x en T . Un conjunto de $k, k \geq 2$, árboles generadores T_1, T_2, \dots, T_k sobre una gráfica G enraizada en el vértice r son **independientes** si para cualquier vértice $x \in V(G) \setminus r$ las k trayectorias $P(T_1, x), P(T_2, x), \dots, P(T_k, x)$ son internamente disjuntas.

Los árboles generadores independientes (*IST*) sobre las gráficas tienen aplicaciones importantes en el algoritmo de difusión de la información pues envían mensajes a los vértices de forma segura a través de las aristas.

Ho [14] propone un algoritmo para construir $n + 1$ árboles generadores ajenos por aristas sobre FQ_n . Recientemente, Yang *et al.* [27] mostraron que los árboles generadores ajenos por aristas sobre FQ_n construidos por Ho son independientes y la altura de cada árbol generador es n . Posteriormente, Chen [6] propuso un algoritmo, que mostramos en este capítulo, para construir $n + 1$ árboles generadores independientes óptimos sobre el hipercubo plegado FQ_n , con la característica de que hay una trayectoria más corta entre el único hijo de la raíz y cualquier otro vértice en cada árbol generador, por lo que la altura de cada árbol generador es $\lceil n/2 \rceil + 1$.

Sea G una gráfica y T un árbol generador de G con raíz en el vértice r . Para cualquier vértice $x \in V(G) \setminus \{r\}$, el padre de x es su vecino sobre la única trayectoria $P(T, x)$ desde r a x en T ; sus hijos son los demás vecinos en T y sus padres son los vértices distintos de x sobre la trayectoria $P(T, x)$. Sea ε_i el vértice en el hipercubo plegado, FQ_n , con $\delta_i = 1$ y $\delta_j = 0$ para toda $j \neq i$. Como el hipercubo plegado es transitivo en vértices, suponemos que el vértice O es la raíz de cada árbol generador sobre FQ_n . Denotamos por T_* al árbol generador sobre FQ_n si el vértice raíz O tiene un único hijo $*$ ($= 2^n - 1$), y denotamos por T_i si el vértice raíz O tiene un único hijo ε_i con $0 \leq i \leq n - 1$. Sea $x \neq O$ cualquier vértice en FQ_n . Denotamos al padre de x en T_* por $f_*(x)$ y el padre de x en T_i por $f_i(x)$ para $i = 0, 1, \dots, n - 1$.

Algoritmo para construir n árboles generadores independientes sobre FQ_n

En esta sección mostramos un algoritmo para construir $n + 1$ árboles generadores enraizados en el vértice O en el hipercubo plegado $FQ_n : T_0, T_1, \dots, T_{n-1}$ y T_* . Para construir estos árboles generadores, determinamos los padres de cada vértice x distinto a el vértice raíz O en cada árbol generador. El algoritmo depende únicamente de la cadena binaria del vértice x . Así puede implementarse en sistemas paralelos o distribuidos. A continuación se presenta, en el algoritmo 5.1, el pseudo-código para la construcción de $n + 1$ árboles generadores independientes en FQ_n .

Algorithm 5.1: Algoritmo para construir $n + 1$ árboles generadores independientes en FQ_n

```

1  Datos: cualquier vértice  $x = \delta_{n-1}, \delta_{n-2}, \dots, \delta_0 \in V(FQ_n) \setminus \{O\}$ 
2  Resultado:  $f_0(x), f_1(x), \dots, f_{n-1}(x)$  y  $f_*(x)$ 
3  foreach  $w \in I \cup \{*\}$  do
4      if  $w = *$  then
5          // determina  $f_*(x)$ 
6          if  $s(x) > \lceil (n+1)/2 \rceil$  then
7               $f_*(x) = \bar{x}$ ;
8          if  $s(x) = 0$  then
9               $f_*(x) = O$ ;
10         if  $s(x) = 1$  then
11              $f_*(x) = *$ ;
12         else
13             //  $2 \leq s(x) \leq \lceil (n+1)/2 \rceil$ 
14              $f_*(x) = x + 2^{\text{succ}(\text{sing}(x))}$ ;
15     if  $w = i \in I$  then
16          $h = h(x, \varepsilon_i)$ ;
17         // determina  $f_i(x)$ 
18         if  $h \leq \lceil n/2 \rceil$  then
19             if  $\delta_i = 0$  then
20                  $f_i(x) = x + 2^i$ ;
21             else
22                 //  $\delta_i = 1$ 
23                 if  $h = 0$  then
24                      $f_i(x) = O$ ;
25                 else
26                     //  $t(x) \geq 2$ 
27                      $f_i(x) = x - 2^{\text{succ}(i)}$ ;
28         if  $h > \lceil n/2 \rceil$  then
29             if  $\delta_i = 1$  then
30                  $f_i(x) = x - 2^i$ ;
31             else
32                 //  $\delta_i = 0$ 
33                 if  $s(x) = 1$  then
34                      $f_i(x) = \bar{x} = \varepsilon_i$ ;
35                 else
36                     //  $s(x) \geq 2$ 
37                     if  $i = \text{sing}(x)$  then
38                          $f_i(x) = \bar{x}$ ;
39                     else
40                          $f_i(x) = x + 2^{\text{succ}(i)}$ ;

```

Mostramos a continuación un ejemplo de la aplicación del algoritmo 5.1 para la gráfica de FQ_4 ilustrada en la figura 5.3. Consideremos al conjunto de vértices $V(FQ_4) \setminus \{O\}$.

$V(FQ_4) = \{(0001), (0010), (0011), (0100), (0101), (0110), (0111), (1000), (1001), (1010), (1011), (1100), (1101), (1110), (1111)\}$.

De acuerdo al algoritmo 5.1, en el árbol generador T_* el padre de cada vértice en $V(FQ_4) \setminus \{O\}$, es como se muestra:

(1110), (1101), (0111), (1011), (0111), (1110), (1111), (0111), (1011), (1111), (1101), (1111), (1111), (0000).

En el árbol generador T_O el padre de cada vértice en $V(FQ_4) \setminus \{O\}$, es como se muestra:

(0000), (0011), (0001), (0101), (0001), (1001), (0101), (1001), (0001), (1110), (1001), (1110), (1001), (0001), (1110).

En el árbol generador $T_i (i = 1, 2, 3)$, el padre de cada vértice en $V(FQ_4) \setminus \{O\}$ se obtiene de manera similar. Así, los cinco árboles generadores T_O, T_1, T_2, T_3 y T_* construidos por el algoritmo 5.1 en el hipercubo plegado, FQ_4 se ilustra en las figuras 5.5 a 5.9.

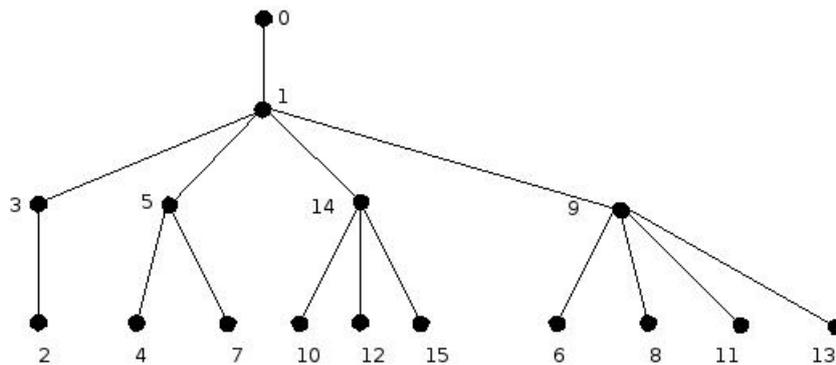


Figura 5.5: Árbol generador T_0 en FQ_4

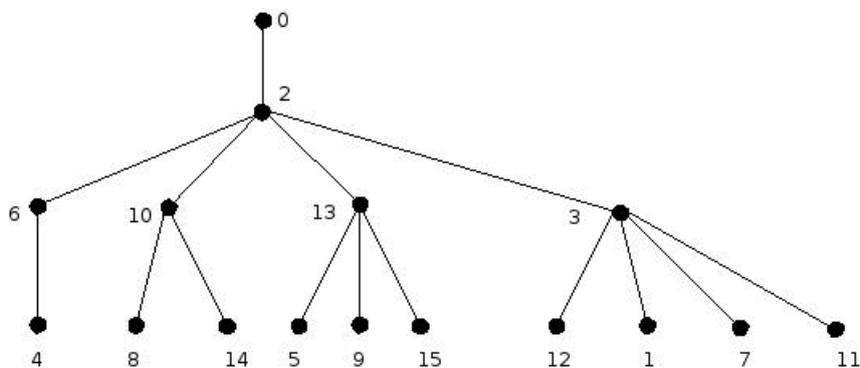
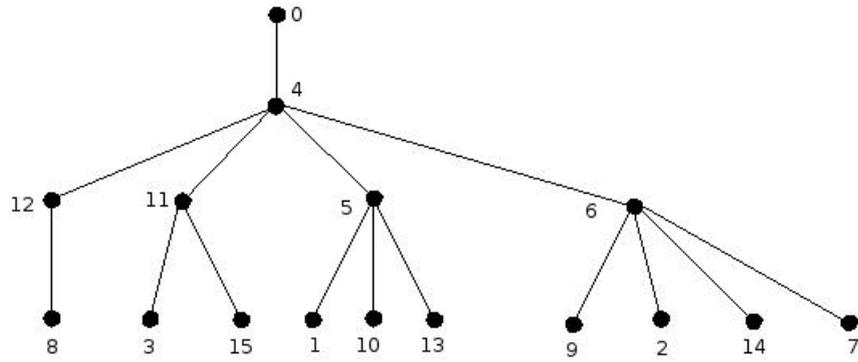
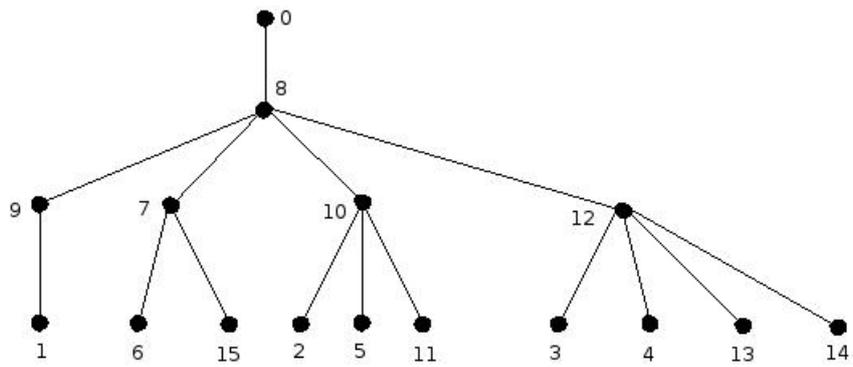
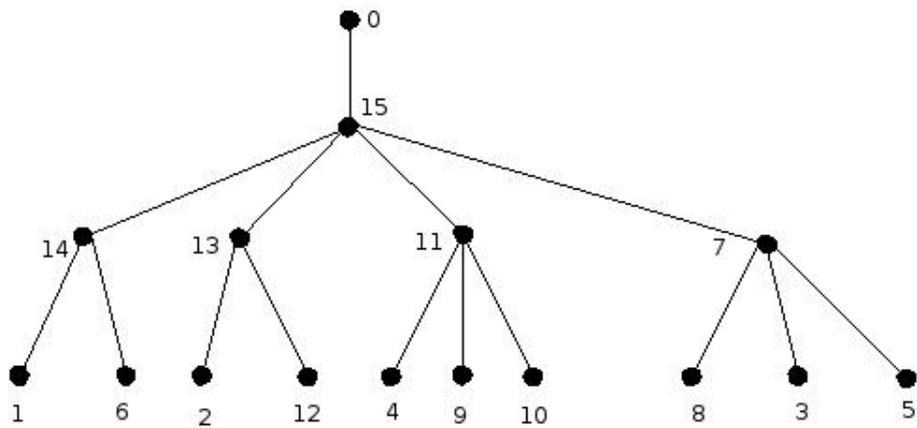


Figura 5.6: Árbol generador T_1 en FQ_4

Figura 5.7: Árbol generador T_2 en FQ_4 Figura 5.8: Árbol generador T_3 en FQ_4 Figura 5.9: Árbol generador T_* en FQ_4

5.3. Algoritmo de enrutamiento

Una red de interconexión es de poco valor si su topología no se emplea de manera eficiente para ejecutar algoritmos cuyos cálculos se distribuyen a través de los nodos de la red. Claramente, algunos algoritmos se basan principalmente en la disponibilidad de las rutas más cortas entre los nodos. Para estos algoritmos, FQ_n es superior en muchas redes, incluyendo la del n -cubo, por su diámetro pequeño. En esta sección presentamos algunos algoritmos de enrutamiento óptimos para FQ_n y mostramos la rapidez obtenida en la comunicación entre los nodos.

Algorithm 5.2: Algoritmo de enrutamiento en FQ_n

- 1 **Datos:** El algoritmo muestra el enrutamiento entre cualquier par de nodos, sean u y $v \in V(FQ_n)$
 - 2 **entrada :** $a(u), a(v)$;
 - 3 $a(w) = a(u) \oplus a(v)$;
 - 4 **if** $\|a(w)\| \leq \lceil n/2 \rceil$ **then**
 - 5 enrutar el mensaje enviado desde u a través de un camino compuesto de aristas con etiquetas que corresponden a las posiciones de bits que tienen 1 en $a(w)$
 - 6 **else**
 - 7 enviar el mensaje a través de u la c -arista; enrutar el mensaje a través de un camino compuesto de aristas con las etiquetas correspondientes a posiciones de bits que son de 0 en una (w)
-

Sin pérdida de generalidad, supongamos que $\|a(w)\| = \tau$ y los bits w_{r-1}, \dots, w_1, w_0 son todos 1 y los bits $w_{n-1}, \dots, w_{r+1}, w_r$, son todos 0. Si $r \leq \lceil n/2 \rceil$, una ruta se forma entre u y v que se compone de cualquier $r!$ permutación de w_{r-1}, \dots, w_1, w_0 . En el caso de $r > \lceil n/2 \rceil$, como se vió en la demostración del teorema 2, un arista c debe ser utilizada en alguna parte a lo largo de la ruta. De ello se deduce que la ruta más corta en este caso es cualquier permutación de $(c, w_{n-1}, \dots, w_{r+1}, w_r)$. Observe que en la formación de una ruta entre u y v , no importa dónde utilizar una arista c , pues la longitud de la trayectoria así obtenida es siempre la misma. Por otra parte, el número de aristas que forman la ruta más corta entre cualquier par de nodos en FQ_n es a lo más $\lceil n/2 \rceil$.

5.4. Difusión de la información

El problema del algoritmo de difusión de la información tanto en el n -cubo como en FQ_n es que dado un nodo s , deseamos encontrar un árbol generador con raíz en s .

El árbol generador con raíz en el nodo 0 de un n -cubo contiene a las aristas que conectan al nodo u con el subconjunto de sus vecinos cuyas direcciones se obtienen al complementar un bit a partir de los ceros en el código binario de u .

Para cualquier nodo origen s , el árbol generador está trasladado por una operación binaria XOR , en todas las direcciones con la dirección del nodo origen, es decir, se forma $v = u \oplus s$. El complemento de los bits de u que corresponden a los ceros iniciales de v define las aristas del árbol generador trasladado. De manera precisa, sea $s = (s_{n-1}, s_{n-2}, \dots, s_0)$, $u = (u_{n-1}, u_{n-2}, \dots, u_0)$ y $v = (v_{n-1}, v_{n-2}, \dots, v_0)$, donde $v_m = s_m \oplus u_m$. Sea $v_k = 1$ y $v_m = 0$ para toda $m > k$ con $k = -1$ para $v = 0^m$, es decir, k es el bit de orden mayor de v que sea un 1. Sean (u, s) el conjunto de los nodos hijos del nodo u en el árbol generador con raíz en el nodo s y $M_{ST}(u \oplus s) = \{k + 1, \dots, n - 1\}$. Entonces, los hijos quedan definidos como:

$$h_{ST}(u, v) = \{(u_{n-1}, u_{n-2}, \dots, u_m, \dots, u_0)\}, \text{ para toda } m \in M_{ST}(u \oplus s)$$

Sea (u, s) el padre del nodo u en el árbol generador, ST , con raíz en el nodo s . Entonces los padres se definen como

$$p_{ST}(u, s) = \begin{cases} \phi, & u = s; \\ u_{n-1}, u_{n-2}, \dots, \bar{u}_k, \dots, u_0, & u \neq s. \end{cases}$$

Se puede observar que las funciones de los vértices hijos y padres son consistentes. La figura muestra un árbol de expansión generado por los hijos (o los padres) en un 4-cubo. En la figura 5.10, los nodos están representados por círculos oscuros. Las aristas se etiquetan y se muestran por líneas rectas.

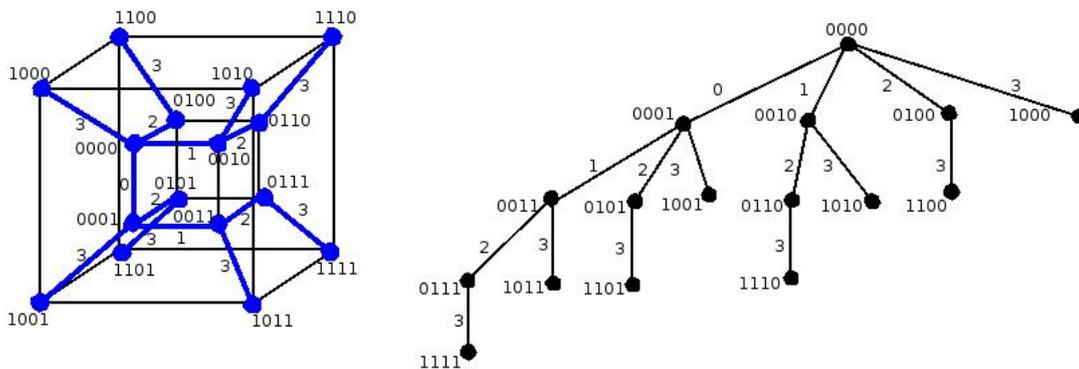


Figura 5.10: Árbol generador para un 4-cubo

Árbol generador¹ para FQ_n : Para encontrar un árbol de difusión en FQ_n , modifica-

¹El autor se refiere a estos árboles como *spanning binary tree* porque sus etiquetas son 0 y 1. Nos referiremos a ellos por sus siglas en inglés como *ST*.

mos todas las direcciones de los nodos mediante la definición de una función U de espacio de direcciones a un nuevo espacio de direcciones. Cada dirección en el nuevo espacio es una $(n + 1)$ -ada. La raíz se encuentra en el nodo s como se supuso antes y $w = u \oplus s$ para un nodo u en FQ_n . Entonces U se especifica de la siguiente manera:

$$U(v_{n-1}, v_{n-2}, \dots, v_1, v_0) = \begin{cases} 0v_{n-1}, v_{n-2}, \dots, v_1, v_0 & \text{si } \|v\| \leq \lceil 2 \rceil; \\ 1\bar{v}_{n-1}, \bar{v}_{n-2}, \dots, \bar{v}_1, \bar{v}_0 & \text{en otro caso} \end{cases}$$

anterior	nueva	anterior	nueva
0000	00000	1111	10000
0001	00001	1110	10001
0010	00010	1101	10010
0100	00100	1011	10100
1000	01000	0111	11000
0011	00011		
0101	00101		
1001	01001		
0110	00110		
1010	01010		
1100	01100		

Cuadro 5.1: Direcciones mapeadas para un 4-cubo

La tabla 5.1 muestra la asignación de dirección para un 4-cubo. Después de haber asignado las direcciones, definimos para FQ_n , $M_{STF}(u, s) = \{k + 1, \dots, n - 1, n\}$ donde k es el bit de orden más alto de $v = u \oplus s$ que sea un 1 tal como se especifica anteriormente. Entonces,

$$h_{STF}(u, s) = \begin{cases} \{(u_n, u_{n-1}, \dots, \bar{u}_m, \dots, u_0)\} \forall m \in M_{SBTF}(u \oplus s) - \{n\} \text{ si } \|a(u)\| = \lceil n/2 \rceil, \\ \text{y } n \text{ es impar;} \\ \{(u_n, u_{n-1}, \dots, \bar{u}_m, \dots, u_0)\} \forall m \in M_{SBTF}(u \oplus s), \text{ en otro caso} \end{cases}$$

Observemos que cuando n es impar, los nodos de peso $\lceil n/2 \rceil$ pueden llegar con o sin recorrido de una arista complementaria en $\lceil n/2 \rceil$ pasos.

Excluyendo $\{n\}$ del conjunto M_{STF} , se garantizará que estos nodos reciben el mensaje sólo a través de la ruta de acceso que consiste en aristas no complementarias.

$$p_{STF}(u, s) = \begin{cases} \phi, & u = s; \\ u_n, u_{n-1}, \dots, \bar{u}_k, \dots, u_0, & u \neq s. \end{cases}$$

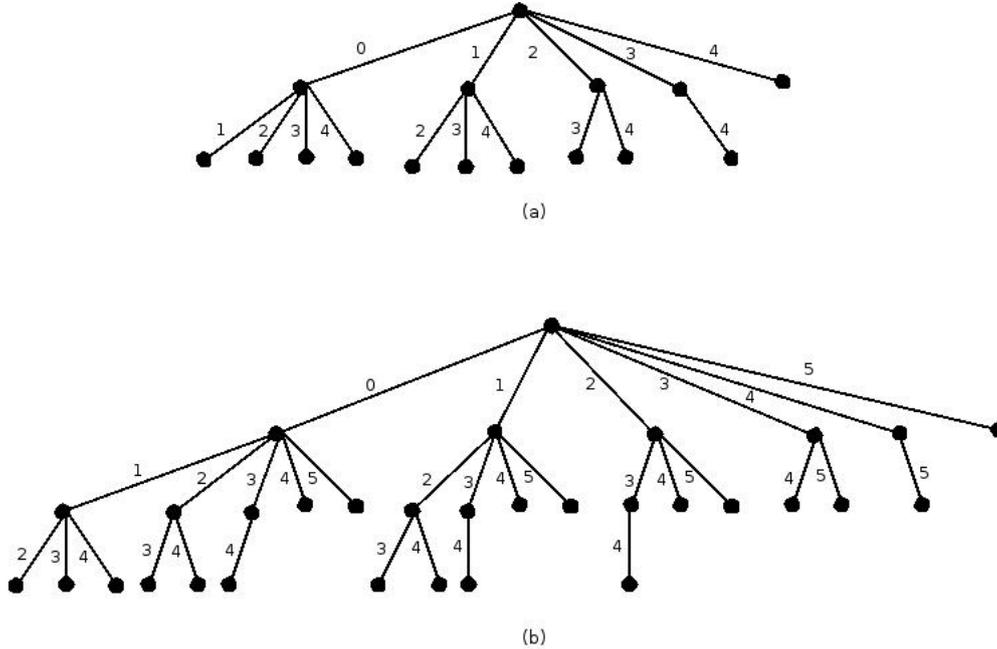


Figura 5.11: Árbol generador de difusión para un FQ_n , (a) $n = 4$, (b) $n = 5$

La figura 5.11 muestra los árboles generadores dados por la función hijo (o padre) para un 4-cubo y un 5-cubo. Notemos que en las figuras 5.10 y 5.11, la etiqueta del nodo origen junto con las etiquetas de las aristas especifica de forma única el árbol de difusión, por lo que, las etiquetas de los otros nodos no se muestran en estas figuras. La altura de un árbol generador para el hipercubo plegado es $\lceil n/2 \rceil$, que es aproximadamente la mitad de la altura de un árbol generador para el hipercubo. Surge el siguiente resultado:

Teorema 5.7. STF es un árbol generador de FQ_n .

Demostración. Comparando la función hijo h_{ST} , o padre p_{STF} , definida por los árboles generadores para el n -cubo y el hipercubo plegado, respectivamente. Podemos ver que todo nodo u con la dirección $a(u) = (0u_{n-1}, u_{n-2}, \dots, u_1, u_0)$ está cubierto por el árbol generador de FQ_n . Esto es cierto puesto que $\|a(u)\| \leq \lceil n/2 \rceil$ y el nodo u será cubierto con certeza después de $\lceil n/2 \rceil$ pasos, comenzando desde la raíz, la cual se supone que es el nodo 0, y cada paso corresponde a atravesar una arista hacia los nodos hoja, los cuales no tiene hijos.

Además, uno de los hijos del nodo u , de acuerdo al algoritmo para el árbol generador del hipercubo plegado, es el nodo u' con dirección $a(u') = (1u_{n-1}, \dots, u_1, u_0)$ y peso $n - \|a(u)\| > \lceil n/2 \rceil$. Como el nodo u es único, el nodo u' también es único. De esta manera, observamos que todos los nodos con peso mayor que $\lceil n/2 \rceil$ están cubiertos de manera única. Así concluimos la demostración. \diamond

En este capítulo se citaron algunas de las propiedades importantes de la gráfica de FQ_n y observamos que comparte con el hipercubo, Q_n , algunas de ellas, por ejemplo

la simetría en vértices y aristas, entre otras. Además se ha mostrado que el hipercubo plegado visto como una red de interconexión es superior en comunicación al hipercubo Q_n pues su diámetro es menor.

Conclusiones

Durante el desarrollo de este documento investigamos propiedades topológicas y algorítmicas de algunas de las variantes del hipercubo, Q_n , entre las que destacan el diámetro, radio, conexidad, simetría. También examinamos resultados sobre hamiltonicidad, conexidad, panciclicidad, panconexidad, transitividad en vértices y en aristas de la red del hipercubo y algunas de ellas en sus variantes.

La estructura topológica de una red de interconexión puede ser modelada por una gráfica cuyos vértices representan procesadores de la red y cuyas aristas representan a los vínculos o conexión entre ellos. Hemos visto que un hipercubo n -dimensional o n -cubo, es una de las redes más eficaces para la computación en paralelo. Tiene muchas características atractivas como la regularidad, la construcción recursiva, transitividad en vértice y aristas, la conexidad máxima, algoritmos de enrutamiento y de difusión de la información, y de esta manera se convierte en la primera opción para la estructura topológica de los sistemas de procesamiento y computación paralela. Sin embargo, el hipercubo tiene algunos inconvenientes respecto a sus propiedades topológicas. Como resultado de ello, se han propuesto diversas variantes para mejorar algunas de estas propiedades tales como el radio y diámetro; algunas de estas variantes se han mencionado en este documento.

En redes de interconexión, la simulación de una arquitectura por otra es importante. El problema de la simulación de una red por otra se modela como un problema de inmersión en gráficas. Las redes de ciclos o trayectorias son adecuadas para el diseño de algoritmos sencillos con bajos costos de comunicación. Estos han motivado a una gran cantidad de investigación en los caminos o ciclos que son sunergibles en otras redes de interconexión. También hemos visto que el problema de la inmersión de trayectorias y ciclos en hipercubos se han estudiado a profundidad. Sin embargo, el mismo problema en las variaciones del hipercubo no se han estudiado mucho, aunque algunos resultados conocidos se mencionaron.

Se ha observado que el hipercubo y sus variantes mencionadas anteriormente tienen estructuras recursivas, y así todas las pruebas proceden de resultados conocidos y aplicando inducción sobre el orden de la gráfica. En el proceso de las demostraciones, para probar la base de inducción en gráficas cuyo orden es pequeño, regularmente se utiliza la observación o verificación directa; en la hipótesis de inducción es la construcción de un camino o de un ciclo requerido por algunas propiedades estructurales de las redes. Así, algunos investigadores consideran que la clave de estudiar problemas de inmersión de ciclos y trayectorias en las variantes del hipercubo es investigar las propiedades estructurales de estas redes.

Índice alfabético

- árbol, 9
 - binario, 10
 - altura, 10
 - completo, 10
 - hoja, 10
 - perfectamente balanceado, 10
 - binomial, 11
 - generador, 10
 - independiente, 72
 - completa, 6
 - completa, 18
 - de Cayley, 16
 - orden, 3
 - pancíclica, 15
 - panconexa, 14
 - producto cuadro, 13
 - tamaño, 3
 - transitiva en aristas, 13
 - transitiva en vértices, 13
- algoritmos de comunicación, 20
 - enrutamiento, 21
 - de Valiant, 21
 - ingenio de enrutamiento mínimo, 22
- automorfismo, 12
- código Gray reflejado, 36
- códigos Gray, 36
- camino, 4
- ciclo, 5, 19
 - hamiltoniano, 5
- conexidad, 7
 - por aristas, 8
 - por vértices, 7
- cuadrado latino, 41
 - distancia Hamming, 41
- cuello, 70
- diámetro, 7
- distancia, 6
 - Hamming, 34
- enrutamiento, 16
- excentricidad, 7
- gráfica, 3
 - bipartita, 5
 - completa, 6
- Hipercubo
 - (n, k) -cubos, 50
 - (n, k) -estrella, 49
 - Balanceado, 48
 - gráfica estrella, 49
 - Localmente Torcidos, 45
 - Moebius, 46
 - Plegado, 67
- hipercubo, 25
 - nodos opuestos, 30
 - operación romper, 26
- longitud de la trayectoria, 5
- nodo
 - hoja, 80
- paridad, 28
- paseo, 5
 - circuito, 5
- radio, 7
- redes de interconexión, 16
 - topologías, 17
 - árboles, 17
 - arreglo lineal, 19

-
- estrella, 18
 - mallá, 17
 - red completa, 18
 - toro, 17
 - ring, 19
 - ruta, 5
 - Subgráfica, 4
 - Teorema de Menger, 8
 - trayectoria, 5
 - ajenas por aristas, 8
 - internamente disjuntas, 8
 - vértice, 3
 - adyacentes, 3
 - grado, 4
 - mínimo, 4
 - máximo, 4
 - terminal, 4

Bibliografía

- [1] B.W. Arden and K.W. Tang. Representation and routing of cayley graphs. Technical report, EL-89-02, Dep. of Electrical Engineering, Univerity of Rochester, 1989.
- [2] B. Barden, R. Libeskind-Hadas, J. Davis, and W. Williams. On edge-disjoint spanning trees in hypercubes. *Information Processing Letters*, (70):13–16, 1999.
- [3] C.P. Chang, J.N. Wang, and J.H. Hsu. Topological properties of twisted cube. *Info Sci*, (113):147–167, 1999.
- [4] G. Chartrand. *Introductory Graph Theory*. Dover Publication, 1977.
- [5] G. Chartrand and O.R. Oellermann. *Applied an Algorithmic Graph Theory*. International Series in Pure and Applied Mathematics. McGraw-Hill, 1993.
- [6] X.B. Chen. Construction of optimal independent spanning trees on folded hypercubes. *information Sciences*, 253:147–156, 2013.
- [7] S.A. Choudum and V. Sunitha. Augmented cubes. *Networks*, 40(2):71–84, 2002.
- [8] P. Cull and S.M. Larson. On generalized twisted cubes. *Information Processing Letters*, (55), 1994.
- [9] P. Cull and S.M. Larson. The Möbius cubes. *IEEE Transactions on Computers*, 44(5):647–659, 1995.
- [10] A. El-Amawy and S. Latifi. Properties and performance on folded hypercubes. *IEEE Transactions on parallel and distributed systems*, 2(1), 1991.
- [11] J. Fan. Hamilton-connectivity and cycle-embedding of the möbius cubes. *Information Processing Letters*, (82):113–117, 2002.
- [12] M.D.L. Gasca and F. Juárez. El análisis amortizado y sus aplicaciones. Vínculos matemáticos no.26-2004, Facultad de Ciencias, UNAM., 2004.
- [13] F. Harary. *Graph Theory*. Addison-Wesley, Philippines, 1969.
- [14] C.T. Ho. Full bandwidth communications on folded hypercubes. *in: Proc. 1990 International Conference on Parallel Processing*, 1:267–280, 1990.

-
- [15] L-H. Hsu and Ch-K. Lin. *Graph Theory and Interconnection Networks*. Taylor and Francis Group, USA, 2009.
- [16] K. Huang and J. Wu. Area efficient layout of balanced hypercubes. *Int. J. High Speed Electron. Syst.*, 6:631–646, 1995.
- [17] W. Imrich and S. Klavzar. *Product Graphs, structure and recognition*. A Wiley-Interscience Publication, 1941.
- [18] G. Kotsis. Interconnection topologies and routing for parallel processing systems. Technical report series, ACPC/TR 92-19, Institut f"ur Angewandte Informatik und Informationssysteme Universit"at Wien Lenaugasse, 1992.
- [19] P.D. Kulasinghe. Conectivity of the crossed cube. *Information Processing Letters*, (61):221–226, 1996.
- [20] S. Latifi and A. El-Amawy. On folded hypercubes. *in Proc.ICPP*, pages 180–187, 1989.
- [21] V. Rantala, T. Lehtonen, and J. Plosila. Network on chip routing algorithms. Technical report,no. 779, University of Turku, Departament of Information Technology Joukahaisenkatu 3-5 B, 20520, Turuku, Finland., 2006.
- [22] Y. Saad and M.H. Schultz. Topological properties of hypercubes. *IEEE Transactions on Computers*, 37(7), 1988.
- [23] J. Werapun, S. Intakosum, and V. Boonjing. An efficient parallel construction of optimal independent spanning trees on hypercubes. *J. Parallel Distrib. Comput.*, (72):1713–1724, 2012.
- [24] J-M. Xu and M-J. Ma. Cycles in folded hypercubes. *Applied Mathematics Letters*, 19(2):140–145, 2006.
- [25] J.M. Xu and M. Ma. Survey on path and cycle embedding in some networks. *Front. Math.*, 40(2):217–252, 2009.
- [26] M. Xu, X.D. Hu, and J.M. Xu. Edge-pancyclicity and hamiltonian laceability of the balanced hypercubes. *Science Direct*, (189):1393–1401, 2007.
- [27] J.S. Yang, J.M. Chang, and H.C. Chan. Independent spanning trees on folded hypercubes. *in: Proceedings of 10th International Symposium on Pervasive Systems Algorithms and Networks*, pages 73–83, 2009.
- [28] J.S. Yang, J.M. Chang, S.M. Tang, and Y.L. Wang. Parallel construction of optimal independent spanning trees on hypercubes. *Parallel Computing*, 33(1):73–79, 2007.