COMPLEJIDAD DE PROBLEMAS COMBINATORIOS

J. DIAZ CORT

Un gran número de problemas entre los que se encuentran problemas tan conocidos como el del viajante de comercio, o el problema isoperimétrico, se muestran como equivalentes en el sentido de que si uno de ellos admite una solución "rápida", en tiempo polinómico respecto el número de variables, todos los restantes también la admiten. Esta teoría, desarrollada en sus principios por S. Cook y R. Karp, usando el modelo de la máquina de Turing, se vuelve a desarrollar aquí usando el modelo más simple de la máquina Combinatoria, debido a K. Harper y J.E. Savage.

1. INTRODUCCION

En 1971, S. Cook demostró que una serie de problemas de lógica matemática eran equivalentes en dificultad al problema de la satisfactoriedad /1/. Karp, /2/, amplió esta clase de "problemas con igual dificultad" a otros problemas del campo de la combinatoria y la optimización lineal. En este mismo trabajo, Karp también indicó la posibilidad de que todos estos problemas "equivalentes" fueran igualmente difíciles de resolver en el sentido de que no tuvieran soluciones en tiempo polinómico respecto el número de variables de input. Esto dividía a los problemas combinatorios en dos clases: Aquéllos con solución obtenida en tiempo polinómico, y aquéllos a los que todavía no se les ha podido encontrar una solución que se obtenga en tiempo polinómico, y cuya mejor solución parece obtenerse en tiempo exponencial respecto el número de variables de input, conocidos respectivamente como problemas de la clase P y de la clase NP. Desde entonces gran número de matemáticos, han separado los problemas pertenecientes o no a la clase NP, haciéndola más y más grande. Así se ha demostrado que problemas de campos tan diversos como la teoría de juegos, circuitos electrónicos, teoría de autómatas y otros, tienen dificultad de resolución equivalente. Cook, Karp y la mayoría de los matemáticos trabajando en este campo, basaron su teoría en el modelo de la máquina de Turing, en sus versiones determinística y nodeterminística (cap. 8, de /3/). En este trabajo, se propone un nuevo desarrollo de esa teoría adoptando el concepto de Máquina Combinatoria, como extensión del trabajo de L. Harper y J.E. Savage, /4/ y /5/.

El uso de máquinas combinatorias simplifica mucho la teoría de la complejidad, sin restarle ni belleza ni utilidad, que como dijo Poincaré son dos conceptos fundamentales en matemáticas. Por ello, el teorema de Cook, pieza clave de la teoría, ha sido reducido apreciablemente, tanto en longitud como en dificultad, (la demostración original viene en /1/).

La notación utilizada a lo largo del artículo, es la notación convencional de la teoría de conjuntos y la teoría de grafos; por ejemplo /6/ y /7/, entre otros muchos, contienen todos los símbolos y conceptos utilizados en este artículo. También en algunos momentos, se necesitarán conceptos muy elementales de lógica elemental, por ejemplo en la demostración del teorema de Cook. Cualquier texto al mismo nivel de /8/ será suficiente para introducirlos.

2. MAQUINAS COMBINATORIAS

Aunque la aplicación de las funciones booleanas a circuitos electrónicos es antigua, el concepto de máquina combinatoria como herramienta en el estudio de la complejidad de --

- J. Diaz Cort de la Universitat de California, California, U.S.A.
- Artículo recibido en Diciembre de 1977.

problemas combinatorios se debe a L.H. Harper y J.E. Savage /4/, /5/ y /3/.

Una Función Booleana es una función que tiene como dominio $\{0,1\}^n$ y como rango $\{0,1\}$, en donde $\{0,1\}^n$ es el producto enésimo cartesiano de $\{0,1\}$ consigo mismo. Un punto arbitrario en el dominio $\{0,1\}^n$ se expresa como $x = (x_1, x_2, \ldots, x_n)$. Las variables $x_1, x_2, \ldots, x_n$ se llaman variables booleanas, y pueden tomar solamente los valores 0 ó 1.

Hay tres funciones booleanas muy conocidas por su uso en electrónica digital:

- La función "Y", $f(x,y) = x \wedge y$

$$x \wedge y = \begin{cases} 1 \text{ si } x=y=1 \\ 0 \text{ en los otros casos} \end{cases}$$

- La función "O", $f(x,y) = x \vee y$

$$x \vee y = \begin{cases} 1 \text{ si } x=1 \text{ ó si } y=1 \\ 0 \text{ en el caso } x=y=0 \end{cases}$$

- La función "NO", $f(x) = \bar{x}$

$$\bar{x} = \begin{cases} 1 \text{ si } x=0 \\ 0 \text{ si } x=0 \end{cases}$$

Estas tres funciones son solamente tres elementos del conjunto de todas las funciones booleanas, pero forman una Base Completa, en el sentido que cualquier otra función booleana puede ser representada como composiciones de estas tres funciones.

Estas tres funciones, por otra parte, cumplen las propiedades asociativa, conmutativa, distributiva y las leyes de Morgan. (Cap. 2 de /3/).

Una Máquina Combinatoria es un grafo orientado y acíclico; en este grafo los nudos que no reciben ningún arco de entrada representan las variables (booleanas) de entrada de la máquina, y los llamaremos Entradas, los nudos que tienen arcos de entrada y arcos de salida representan una de las tres funciones booleanas ($\wedge$, $\vee$, $-$), y serán conocidos como Elementos Funcionales. De este modo, las variables de cada elemento funcional serán los arcos de entrada del correspondiente nudo. -

Dado que ($\wedge$, $\vee$, $-$) constituyen una base completa, con una máquina combinatoria de este tipo se puede representar cualquier función booleana.

A modo de ejemplo de máquina combinatoria, la función

$$f(x,y) = \begin{cases} 0 \text{ si } x=y \\ 1 \text{ si } x \neq y \end{cases}$$

conocida como O-EXCLUSIVO, puede ser representada usando la base ($\wedge$, $\vee$, $-$) por $f(x,y) = (x \vee y) \wedge (\bar{x} \vee \bar{y})$, siendo la correspondiente máquina combinatoria que la computara la de la fig. 1.

El Costo de una máquina combinatoria es el número total de elementos funcionales del tipo $\wedge$ ó $\vee$. El costo de la máquina representada en la figura 1 es 3.

La Complejidad de una función f, C(f), es el costo mínimo de cualquier máquina combinatoria que compute f. O, dicho de otra forma, la complejidad de f es el mínimo número de variables lógicas ($\wedge$, $\vee$, $-$) necesarias para representarla.

Dada una máquina combinatoria existe un teorema que da una acotación superior con respecto al número mínimo de elementos funcionales necesarios para representar la función que compute dicha máquina.

Teorema de Shannon-Lupanov: Si f es una función booleana de n variables, y n es suficientemente grande, entonces $C(f) \leq 2^n/n$.

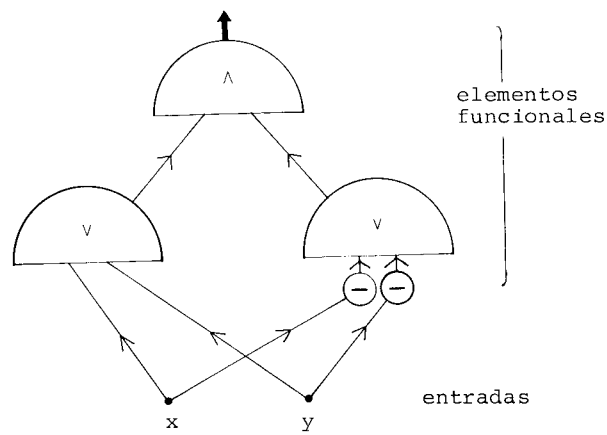


Fig. 1
Máquina combinatoria de la función O-exclusivo.

Para la demostración y para extenderse más sobre la teoría de máquinas combinatorias, véase el cap. II y III de /3/ y /5/.

3. LA CLASE P

Para estudiar la complejidad de los problemas combinatorios, es más conveniente su planteamiento en forma "decisional" en vez de una búsqueda del valor óptimo, que suele ser su enunciado más común. No es difícil ver que ambas formas son equivalentes en el sentido de que una solución para un problema en forma decisional será también una solución para el mismo problema en forma de valor óptimo, y viceversa. Veamos a modo de ejemplo el problema de la Enumeración Mínima de un Grafo. Este es el siguiente: dado un grafo G(V,A), en donde V es el conjunto de vértices del grafo, con cardinalidad n, expresado $|V|=n$, y A es el conjunto de aristas. Se define una enumeración de G, como una función isomórfica $f:V \to \{1,2,.....,n\}$, en donde como se ha señalado, $n=|V|$. Denotando por F el conjunto $\{f_1,f_2,f_3,...,f_{n!}\}$ de todas las posibles enumeraciones del grafo G(V,A), la cardinalidad de F es $|F|=n!$. El problema de la enumeración mínima de G, consiste en hallar la enumeración $f_i \in F$ tal que minimice la suma de todas las diferencias entre los valores asignados por f_i a los vértices conectados por aristas entre sí. En forma más matemática, el problema de la enumeración mínima en G(V,A) es hallar la $f_i \in F$ tal que minimice

$$\sum_{\substack{u,v \in V \\ \{uv\} \in A}} |f_i(u) - f_i(v)|.$$

El mismo problema en su versión decisional sería, dado el grafo G(V,A) y un entero positivo K, encontrar la enumeración de G; $f_i \in F$, tal que

$$\sum_{\substack{u,v \in V \\ \{uv\} \in A}} |f_i(u) - f_i(v)| \leq K$$

Al enunciar un problema en forma decisional, facilita la codificación del input del problema como secuencias de 0 y de 1, es decir en forma de secuencias de variables booleanas. El output del problema, también será una variable booleana, 0 si el resultado de la comparación es negativo, (en el caso del problema de la enumeración mínima, por ejemplo, sería si la suma de las diferencias NO fuera menor o igual que K), y el valor sería 1 en el caso que el resultado de la comparación sea positivo. Entonces el resolver un problema decisional, no es más que el resolver una secuencia de funciones booleanas, y por lo tanto para esto podemos utilizar máquinas combinatorias.

Vamos a definir una variante de la máquina combinatoria:

Una Máquina Combinatoria Determinística, abreviadamente MCD, es una máquina combinatoria tal que sus entradas consisten en:

a) el imput del problema codificado como un conjunto de variables booleanas, con una variable por cada entrada de la máquina,

b) el resto de las entradas serán todas las posibles soluciones del problema. La MCD hace las correspondientes comparaciones con cada una de las soluciones generadas, y da como salida final 1, si existe una solución que sea "compatible" con el input del problema, y 0 en caso contrario.

Vamos a ver un ejemplo que clarifique la definición. Definamos el problema de la banda[1].

PROBLEMA DE LA P-BANDA: Dado un grafo G(V,A) se trata de encontrar un subgrafo completo de orden P. En otras palabras, dado G(V,A), encontrar si existe o no un subgrafo S(W,E) con $W \subseteq V$, $E \subseteq A$, y que $|W|=P$, siendo S completo (todos los vértices de S conectados entre sí por su correspondiente arista). Por tanto, si S(W,E) es completo, el número de aristas de S, es decir |E|, es igual a $\frac{P(P-1)}{2}$ (/7/).

MCD que resuleva el problema de la P-Banda

Tal MCD tendrá las siguientes entradas:

1) $\frac{n(n-1)}{2}$ entradas, cada una correspondiendo a una posible arista del grafo F(V,A). Estas entradas serán enumeradas como a_{12}, a_{13}, a_{14}, ..., a_{1n}, a_{23}, a_{24}, ... $a_{(n-1)n}$, en donde los subíndices ij expresan la posible arista entre los vértices i y j. Si realmente la arista existe

entre esos dos vértices de G(V,A), entonces $a_{ij}=1$, de otra forma $a_{ij}=0$.

2) $\log_2 \frac{n(n-1)}{2}$ entradas que especificarán el valor de P, por el número de aristas entre los P vértices de la banda, en notación binaria.

3) La parte determinística de la máquina consistirá en generar todas las posibles bandas de orden 3, de orden 4, etc., hasta de orden n. Para esto se hacen copias iguales de n entradas de la MCD por cada copia, denominadas $v_1, v_2, \ldots, v_n$, que representan los n vértices de G. Para representar una P-banda, se darán a P de esas entradas el valor 1, y al resto, las (n-P) entradas, se le dará el valor 0. Por ejemplo, la primera 3-banda que se generará será el 111000...0, lo que significa que v_1, v_2 y v_3 pertenecerán a la banda. La segunda 3-banda generada será el 110100...0, y así sucesivamente. El número de 3 bandas que se necesitará generar, serán todas las posibilidades de tomar n elementos (los n vértices de G), de tres en tres, es decir $\binom{n}{3}$ y cada uno de esas 3-bandas necesitará a su vez n entradas de la máquina combinatoria; entonces el número total de entradas de la MCD que serán necesarias para representar todas las posibles 3-bandas, serán $\binom{n}{3}\cdot n$. De la misma manera, el número de entradas de la MCD necesarias para representar todas las 4-bandas serán $\binom{n}{4}\cdot n$, y así sucesivamente hasta que lleguemos a que el número de entradas de la MCD necesarias para generar las n-bandas (en caso de que todo el grafo G sea completo) será $\binom{n}{n}\cdot n = n$.

Entonces el número total de entradas de que dispondrá la MCD será

$$\frac{n(n-1)}{2} + \log_2 \frac{n(n-1)}{2} + \sum_{i=3}^{n} \binom{n}{i}\cdot n$$

Estas entradas de la MCD se representan en la fig. 2.

Se puede descomponer esta MCD en pequeños módulos, para mayor claridad.

Las entradas correspondientes a las soluciones de las posibles bandas usan los vértices de G, mientras que las entradas correspondientes al input del problema, usan las aristas de G. En este caso el primer módulo que se necesita, sea M1, toma las n entradas correspondientes a cada banda, y las compara cada una de ellas con todas las restantes, dos a dos, mediante un elemento funcional $\wedge$, si dos de las entradas son 1, el resultado de la comparación será 1, lo cual significa que como los dos vértices que se están comparando, pertenecen a la banda, la arista correspondiente existirá (por la definición de banda) y por lo tanto el resultado será 1. En total por cada módulo M1 se tienen $\binom{n}{2}$ salidas, que corresponderán a las posibles aristas de G, las salidas con 1 serán las aristas que conectan los vértices de la banda correspondiente. Ver la fig. 3.

Pero se necesita un módulo M1 por cada n entradas correspondientes a una P-banda, es decir que se necesitarán $\binom{n}{p}$ módulos M1 para expresar todas las posibles P-bandas. Pero se están generando todas las bandas desde P=3 hasta P=n, entonces en total se necesitarán

$$\sum_{i=3}^{n} \binom{n}{p} \text{ módulos M1.}$$

El costo de cada M1 será $C(M1) = \frac{n(n-1)}{2}$. El número de salidas de cada módulo M1 será $\binom{n}{2}$, una por cada elemento funcional.

El módulo M2 consistirá en un módulo M1 aplicado a uno de los conjuntos $\{v_1, v_2, \ldots, v_n\}$ que representan las soluciones generadas de la banda. Las salidas de este M1 han de ser comparadas con el input del problema, representado en las primeras $\frac{n(n-1)}{2}$ entradas de

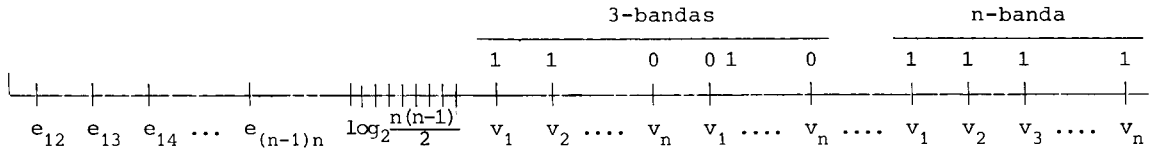


Fig. 2
Entradas de la MCD para la n-banda

la MCD en cuestión. Uniendo cada salida V_{ij} de M1 con la correspondiente entrada de la MCD, e_{ij}, mediante un elemento funcional $\wedge$, si ambos son 1, lo que significa que el lado e_{ij} existe en la banda, entonces la salida del elemento funcional será 1. Así se formarán $\frac{n(n-1)}{2}$ elementos funcionales del tipo $\wedge$, uno por cada par (V_{ij},e_{ij}) y por lo tanto se tendrá el mismo número de salidas. Estas salidas se llevarán a un módulo CT, que consiste en un contador binario que cuenta el número de entradas con valor 1, y como salida expresa esa suma de unos en notación binaria. Como el máximo número de unos posible es el mismo que de entradas al CT, para representarlo en notación binaria se precisan $\log_2\frac{n(n-1)}{2}$ salidas de CT. No es necesario el especificar el contador CT, pues existe abundante literatura al respecto (/3/), baste con decir que es posible construir CT con un costo menor o igual a la cuarta potencia del número de entradas, es decir polinómido de orden 4. Finalmente el módulo M2 se completará observando que el número de lados de la banda, coincide con la especificación del número de aristas entre los P vértices que entran en la MCD. Para esto se compara cada salida de CT con las entradas correspondientes al input del problema, por medio del conjunto de elementos funcionales indicado en la fig. 5.

Si la salida es 1, el correspondiente dígito binario es el mismo. Para comprobar si todos los $\log_2\frac{n(n-1)}{2}$ dígitos son igual, o sea representan el mismo número, se puede poner una cascada de elementos funcionales del tipo $\wedge$. En total el módulo M2 quedará como en la figura 4.

El costo de cada M2 será:

$$C(M2) \leq C(M1)+C(CT)+\frac{n(n-1)}{2}+3\cdot\log_2\frac{n(n-1)}{2}+$$

$$+\log_2\frac{n(n-1)}{2} \leq n(n-1)+4(\log_2 n(n-1)-$$

$$-\log_2 2)+\left(\frac{n^2(n-1)^2}{2}\right)^4 =$$

$$= n(n-1)+\frac{n^4(n-1)^4}{16}+4\log_2 n(n-1)-4$$

Como se ha visto cada M2 funciona con cada una de las bandas generadas por las entradas de la MCD. De este modo el módulo M3 consistirá en la serie de módulos M2 para todas las bandas de un mismo valor P, con $3\leq P\leq n$. Ver fig. 6. Si uno de los módulos M2 tiene como salida un 1, es que ese módulo contiene en la entrada el valor generado que es solución al input del problema representado en las primeras $\frac{n(n-1)}{2}$ entradas de la MCD. Entonces se completará M3 enlazando las salidas de los $\binom{n}{p}$ módulos M2 que contiene, mediante elementos funcionales $\vee$.

El costo de M3 será $C(M3)\leq\binom{n}{p}\cdot C(M2)+\binom{n}{p}$.

Si uno de los n-3 módulos M3, uno por cada valor P de las P-bandas que generamos, tiene como salida 1, significará que habrá al menos una P-banda en nuestro grafo, por lo que la versión final de la MCD será la serie de (n-2) módulos M3 y las salidas conectadas mediante elementos funcionales del tipo $\vee$. La versión final de la MCD M, es la

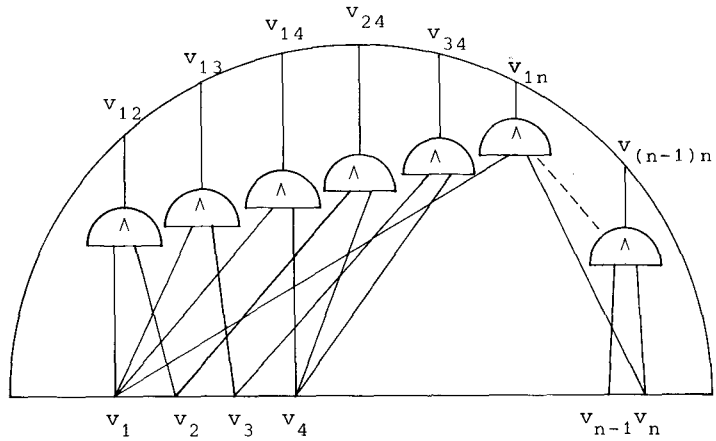


Fig. 3
Salidas de módulo M1

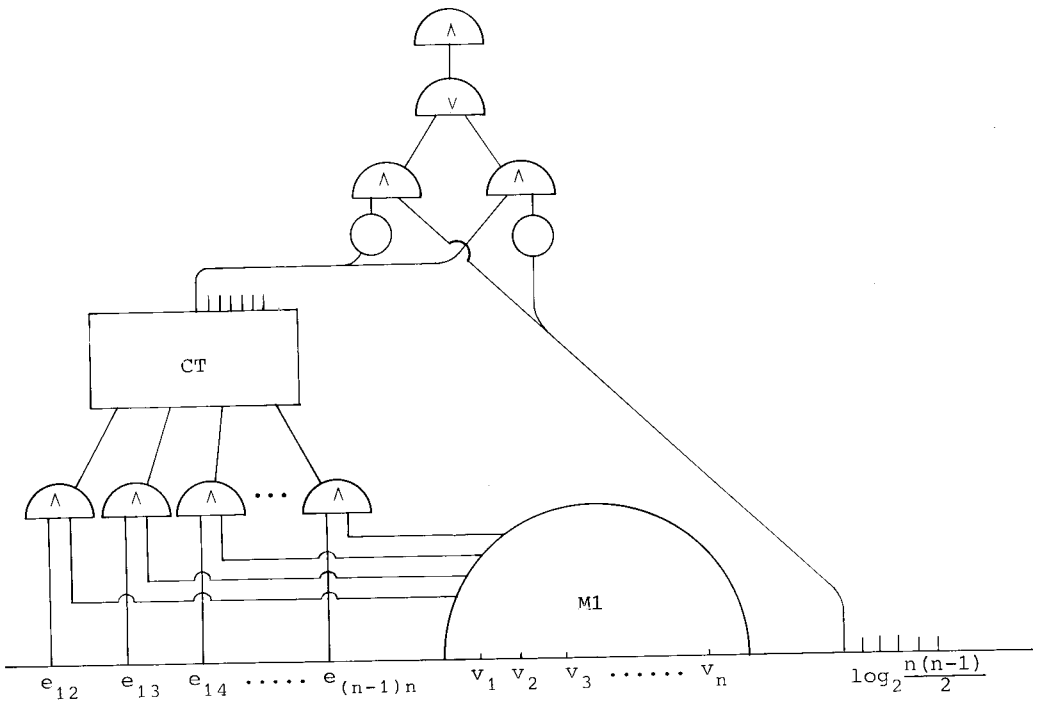


Fig. 4
Módulo 2

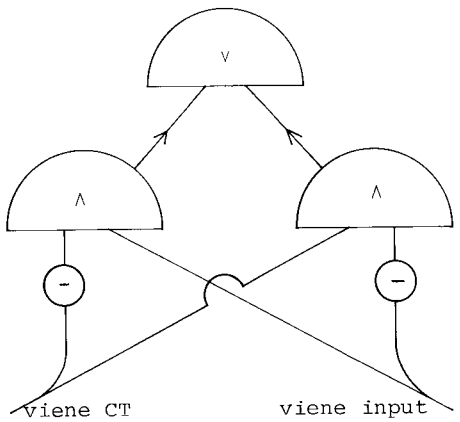


Fig. 5
Comparación salida CT con entrada

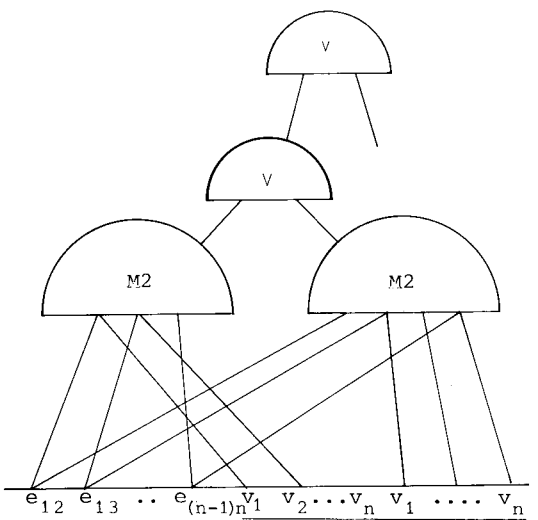


Fig. 6
Módulo 3

de la fig. 7.

El costo total de la MCD será:

$$C(M) \leq (n-2) + \sum_{P=3}^{n} \left[\binom{n}{p} \cdot \left(n(n-1) - 4 + 4\log_2 n(n-1) + \right. \right.$$

$$\left. \left. + \frac{n^4(n-1)^4}{16} \right) + \binom{n}{p} \right]$$

simplificando

$$C(M) \leq (n-1) + \left[(n^2-n) + 4\log_2 n(n-1) + \right.$$

$$\left. + \frac{n^4(n-1)^4}{16} \right] \sum_{P=3}^{n} \binom{n}{p}$$

pero teniendo en cuenta la fórmula de Stirling en el caso intermedio en que $P=\frac{n}{2}$, se tiene:

$$\binom{n}{p} \simeq \frac{n!}{\frac{n}{2}! \cdot \frac{n}{2}!} = \frac{\sqrt{2n\pi} \cdot n^n \cdot e^{-n}}{2 \cdot \sqrt{\pi n} \cdot n^{n/2} \cdot (2e)^{-(n/2)}}$$

entonces será igual:

$$\binom{n}{n/2} \simeq \sqrt{\frac{2}{n \cdot \pi}} \cdot 2^n$$

Es decir, que $\binom{n}{p}$ puede ser exponencial, y si se substituye este valor en la fórmula que se ha deducido para el C(M), la fórmula que resulta puede llegar a ser de tipo exponencial.

La CLASE P, es la clase de secuencias de funciones booleanas, (clase de problemas), que se pueden computar usando máquinas combinatorias determinísticas, que trabajan con costo polinómico.

Una lista de problemas pertenecientes a esta clase, vienen en /2/. Algunos de los problemas en esta lista son:

a) Dado un grafo G(V,A), hallar la ruta más corta entre dos vértices. (Resuelto por Dijkstra en 1959).

b) El problema de la SATISFACTORIEDAD con un máximo de dos variables por cláusula (Cook 1971).

c) Si Z_p denota el conjunto de los p primeros enteros, dado el subconjunto $S \subseteq Z_p \times Z_p$ (producto cartesiano), determinar si hay o no, p elementos de S tales que dos cualesquiera no tienen componentes iguales.

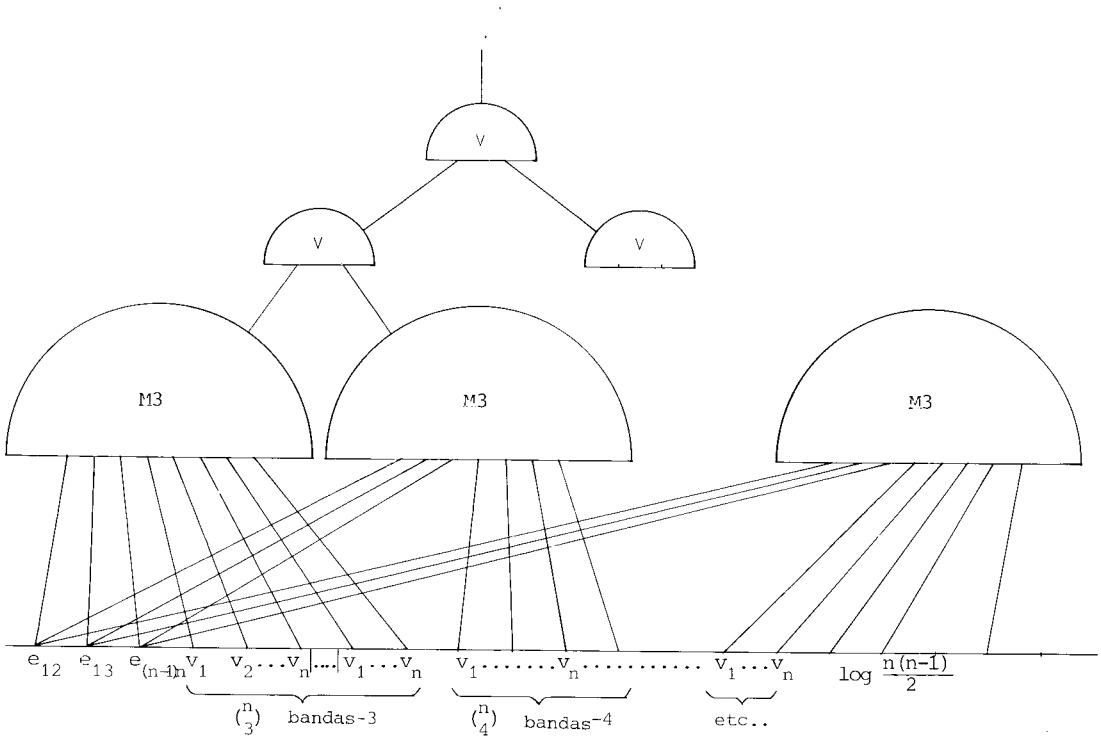


Fig. 7
Esquema final de la MCD

4. LA CLASE NP

Al igual que para definir la clase P, tuvimos que definir lo que era una MCD, ahora vamos a definir su equivalente nodeterminístico.

Una Máquina Combinatoria Nodeterminística, abreviadamente MCND, es una máquina combinatoria tal que las entradas consisten en:

a) el input del problema a resolver, codificado como un conjunto de variables booleanas, como en el modelo de la MCD, y

b) en el resto de las entradas se tendrá una única solución conjeturada, que además será la correcta.

El resto de la MCND será similar a la MCD, es decir comparará el input con la solución correcta conjeturada y como salida dará 1 ó 0 según la solución exista en el input o no.

Así se ve que la gran diferencia entre ambas máquinas es que mientras en el modelo determinístico se generan todas las soluciones posibles, en el modelo nodeterminístico se "adivina", heurísticamente, la solución correcta y se reduce en mucho el número de entradas de la máquina nodeterminística con respecto a la determinística.

MCND que resuelva el problema de la P-banda

Tal MCD tendrá las siguientes entradas:

1) $\frac{n(n-1)}{2}$ entradas que especificarán las aristas del grafo G(V,A), exactamente igual que en el caso de la MCD.

2) $\log_2 \frac{n(n-1)}{2}$ entradas que especificarán el valor de P, contando el número de aristas entre los P vértices (en notación binaria).

3) n entradas que especificarán la solución conjeturada de la P-banda.

El resto de la MCND será igual al módulo M2 de la máquina determinística para la P-banda que hemos construido antes; los valores v_1, $v_2, v_3, \ldots, v_n$, en cada M2, que en el caso de la MCD correspondían a una de las soluciones generadas, en la MCND corresponderán a los valores de la solución única conjeturada.

Por un procedimiento análogo al seguido previamente para hallar el costo de M2, el costo de esta MCND para la P-banda será:

$$C \leq \frac{n(n-1)}{2} + \frac{n(n-1)}{2} + C(CT) + 3n(n-1) =$$

$$= 4(n-1) + C(CT)$$

Como se ve el costo es polinómico. Esto nos dice que el problema de la P-banda tiene complejidad polinómica cuando se computa con una MCND.

La CLASE NP, es la clase de secuencias de funciones booleanas, que se pueden computar usando máquinas combinatorias nodeterminísticas, con costo polinómico.

Cualquier MCD que trabaje con costo polinómico, implica que la correspondiente MCND construida "adivinando" la solución correcta y eliminando las otras, tendrá también costo polinómico; de hecho, el costo de la MCND será menor que el de la correspondiente MCD. Esta observación indica que todo problema en la clase P pertenece también a la clase NP, por lo tanto $P \subseteq NP$. Como se verá más adelante en detalle, la gran interrogación en este campo es ver si $P \subset NP$, o si $P = NP$. Hay indicios de tipo heurístico para creer que la respuesta será $P \subset NP$, como indicó Karp /2/.

5. REDUCIBILIDAD

En la teoría de la complejidad, la técnica de transformar un problema en otro es crucial, pues dos problemas equivalentes tendrán complejidades equivalentes, es decir pertenecerán a la misma clase.

Dados dos problemas L y M, recordemos que se pueden representar respectivamente por dos secuencias de máquinas combinatorias l_s y m_j. Diremos que el problema L es REDUCIBLE al problema M, $L \propto M$, si existe una secuencia de funciones booleanas f_k, en donde cada f_k tiene como dominio I_s las entradas de cada máquina l_s, y como rango I_j las entradas de cada m_j, y que se cumpla que $l_s = m_j \cdot f_k$, y además que cada f_k se pueda computar con una MCD de costo polinómico.

Esta definición dice que para que un problema sea reducible a otro, basta encontrar una secuencia de funciones, con complejidad polinómica, que transformen las entradas de las MC de un problema en las entradas de las MC del otro problema. En la práctica, es suficiente encontrar una función que transforme el input de L en el input de M y que dicha función tenga complejidad polinómica. Vamos a definir un problema de lógica elemental que es fundamental en el estudio de la clase NP.

PROBLEMA DE LA SATISFACTORIEDAD: dado un conjunto de n variables booleanas y sus negaciones, $X=\{x_1,x_2,\ldots x_n, \bar{x}_1, \bar{x}_2,\ldots\bar{x}_n\}$ podemos formar unas ciertas cláusulas C_1, C_2, $C_3,\ldots$, C_k, en donde cada cláusula es la disyunción de elementos de X; $C_i=\vee x_j$. El problema consiste en encontrar un subconjunto $S \subseteq X$, tal que haga satisfactoria la conjunción de todas las cláusulas, $C_1 \wedge C_2 \wedge C_3 \wedge C_k$. Dicho de otra forma, el problema consiste en encontrar un $S \subseteq X$, tal que si damos valor 1 a todos los elementos de S, la conjunción de todas las cláusulas también dé 1. Para que esto ocurra, es suficiente que S cumpla dos condiciones:

1) $|S \cap \{x_1,x_j\}| \leq 1$. (Claramente S no puede contener una variable y su complemento, puesto que tenemos que hacer 1 todos los elementos de S).

2) $\{S \cap C_i\} \neq \emptyset$, para todo i tal que $1 \leq i \leq k$. (Si cada cláusula contiene al menos un elemento de S, eso significará que cada cláusula tomará el valor 1, y por lo tanto la conjunción de todas las cláusulas será 1).

SATISFACTORIEDAD $\propto$ P-BANDA: dado el input para la satisfactoriedad, k cláusulas y el conjunto X, se debe encontrar una transformación F, que transforme el input anterior en el input de la P-banda, o sea en un grafo G(V,A) y en el valor P; demostrar que F tiene complejidad polinómica y que el problema de la satisfactoriedad será verdad con el mencionado input si, y solo si, el problema de la banda es verdad con el input correspondiente a la transformación.

La transformación será:

$$F:\begin{cases} V=\{(x,C_i);\ \text{donde } x\in X \text{ y } x\in C_i\} \\ A=\{\{(x,C_i),(y,C_j)\};\ \text{donde } i\neq j,\ x\neq y,\ x,y\in X\} \\ P=k \end{cases}$$

F transformará las cláusulas $C_1, C_2, C_3,\ldots.., C_k$, y las variables $X=\{x_1,x_2,..x_n,x_1,\ldots x_n\}$, en el grafo de la fig. 8. Los ejes verticales representarán los elementos de X, alternando las variables y las negaciones, y el eje horizontal representará las cláusulas $C_1\ldots..C_k$. Poniendo un vértice de G, en todas las intersecciones de filas y columnas, tales que la x (ó $\bar{x}$), pertenezca a la cláusula de la columna en consideración. Por ejemplo en la fig. 8 está representada la transformación de $C_1=\{x_2,x_n\}$, $C_2=\{\bar{x}_n\}$, $C_3=\{x_1,\bar{x}_2,\bar{x}_n\}$, ... $C_k=\{x_1\}$. Ahora unimos los vértices entre sí excepto:

a) cuando los vértices estén en la misma columna

b) cuando los vértices estén en filas complementarias $(x_i,\bar{x}_i)$.

El grafo G(V,A) obtenido en esta forma contiene una P-banda si, y solo si, la conjunción de las cláusulas es satisfactoria.

Demostración: Suponiendo que la satisfactoriedad se cumple, es decir, existe un $S \subseteq X$ tal que

1) $\{S \cap C_i\} \neq \emptyset$

2) $|S \cap \{x_i,x_j\}| \leq 1$

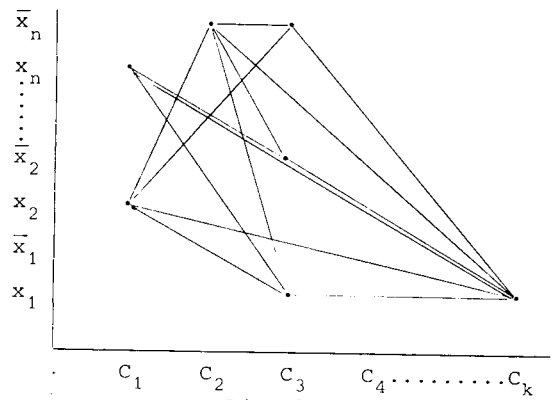


Fig. 8
Transformación cláusulas → variables

La condición 1) implica que podemos formar un subconjunto de los vértices de G, $W \subseteq V$, tal que exista un vértice por cada columna, al menos. Por la condición 2), no existen dos elementos en W que estén en filas complementarias, (pues eso rompería la hipótesis). Si $|W|=k$, eso significa que sólo existe un vértice por columna, por lo tanto por la forma como hemos construido el grafo, contiene un subgrafo completo en los W vértices, es decir contiene un a k-banda. Si $|W|>k$, significa que hay una o más columnas que contienen más de un vértice; en esas columnas escogemos sólo un vértice por columna, formando un nuevo subconjunto $W' \subseteq W$, tal que $|W'|=k$, con lo que se reduce al caso anterior.

Por otra parte, partiendo de un grafo creado del modo anteriormente expresado, mediante la transformación F, con una k-banda, se demostrará que en este caso debe existir una solución $S \subseteq X$, que haga las cláusulas satisfactorias. Por construcción del grafo, si existe un subgrafo completo en k vértices, existe un vértice por cada columna, es decir si se denomina S al conjunto de las x que forman los vértices de la k-banda, se tiene que $\{C_j \cap S\} \neq \emptyset$, (pues para que exista la banda, deben haber un vértice por columna). Así pues la condición 1) se cumple. La condición 2) deberá cumplirse pues si no, tendría que haber dos elementos de S que serían complementarios, x_i e $\bar{x}_i$, lo cual implicaría que los vértices correspondientes no podrían estar conectados entre sí por una arista, por construcción del grafo, lo que va contra la hipótesis de que existe una k-banda.

Esto prueba que cuando la satisfactoriedad opera, la k-banda construida a partir del input de satisfactoriedad, también actúa, y viceversa. Por tanto solo queda por probar que F debe tener complejidad polinómica.

Construyamos la siguiente MCD: las entradas de la máquina serán variables del siguiente tipo

$$y_{ij}^{n} = \begin{cases} 1 \text{ si} \begin{cases} x_i \in C_j \text{ y además } n=1, \text{ esto último significa que es } \bar{x}_i \text{ y también} \\ x_i \in C_j \text{ y además } n=0, \text{ no es el complemento.} \end{cases} \\ 0 \text{ En todos los otros casos.} \end{cases}$$

El significado de estas entradas es que, si es del tipo y_{ij}^{1} representa un vértice $(\bar{x}_i, C_j)$ del grafo que hemos construido antes, y si es del tipo y_{ij}^{0} representa un vértice (x_i, C_j), esto será verdad cuando $y_{ij}^{0}=1$ o cuando $y_{ij}^{1}=1$.

Las salidas de la MCD serán de la forma

$$z_{ijkl}^{n\tilde{n}} = \begin{cases} 1 \text{ si existe una arista entre } y_{ij}^{n} \text{ e } y_{kl}^{\tilde{n}}. \\ 0 \text{ si no existe una arista entre } y_{ij}^{n} \text{ e } y_{kl}^{\tilde{n}} \end{cases}$$

pero como la condición para que exista una arista debe de ser que $y_{ij}^{n}=y_{kl}^{\tilde{n}}=1$, la máquina estará constituída por elementos funcionales que comparen las entradas dos a dos, y den 1 cuando ambas entradas sean 1, es decir $\wedge$. Ver fig. 9.

Como la satisfactoriedad es válida, no es preciso cuidar de que caigan o no en la misma columna, o de que estén o no en filas complementarias.

El costo de la MCD será el número total de entradas tomadas dos a dos. El número de entradas será 2nP, entonces el costo será menor o igual que

$$\binom{2nP}{2} = \frac{2nP \cdot (2nP-1)}{2} \leq 2n^2P^2 - nP.$$

Lo que quiere decir que la complejidad de F será $C(F) \leq 2n^2P^2-nP$, lo cual es complejidad polinómica, (cuadrática).

Conclusión: SATISFACTORIEDAD $\propto$ P-BANDA.

6. LA CLASE NP-COMPLETA

La reducibilidad, $\propto$, es una operación transitiva: si L, M y T, son tres problemas, $L \propto M$ y $M \propto T$, implican $L \propto T$. Esta propiedad permitirá demostrar la equivalencia dos problemas usando uno intermedio.

Teoremas relativas a las clases P y NP

Teorema 1: Si $L \propto M$ y si $M \in P$ entonces $L \in P$.

Demostración: Por la definición de reducibilidad, existe una secuencia de funciones -- con complejidad polinómica, f_k, tales que - $l_s = m_j \cdot f_k$. Como las m_j tienen complejidad polinómica, eso significa que las l_s la tienen también, lo que a su vez significa que L∈P. q.q.d.

El problema de la SATISFACTORIEDAD es el -- ejemplo principal de la clase NP. Los siguientes teoremas, dan una medida de la importancia de dicho problema.

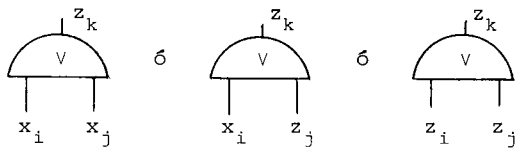
Teorema 2, Teorema de Cook: Todo problema en la clase NP es reducible al problema de la - SATISFACTORIEDAD.

Demostración: Si L∈NP, implica ∃ una MCND M tal que si se enuncia L como problema decisional, como se ha venido haciendo a lo largo de este artículo, se tendrá como entradas de M un conjunto de variables $x_1, x_2, x_3, \ldots x_n$, y un conjunto de variables auxiliares $y_1, y_2, \ldots, y_t$, que darán el valor conjeturado. El número de elementos funcionales de M tiene que ser ≡ p(n), (polinómico en n).

La línea general de la demostración va a ser el representar todos los elementos funcionales en M, usando sólo ∨.

Designando por z_k la salida del elemento funcional número k, en M, se tiene que un elemento funcional de la forma $z_k = x_i \wedge x_j$ se puede transformar al tipo ∨ usando las leyes de Morgan, es decir quedará como $z_k = (\overline{\bar{x}_i \vee \bar{x}_j})$.

Asignando valores fijos a las variables de - entrada $x_1 \ldots x_n$, se tendrá que en M habrá -- tres tipos de elementos funcionales:



Clase 1 Clase 2 Clase 3

Una relación de lógica que es imprescindible para el desarrollo de la demostración es -- $(x \Rightarrow y) \Leftrightarrow (\bar{x} \vee y)$. (Cap. II, /8/).

Para representar un elemento funcional de la clase 1, se necesitará el siguiente número - de cláusulas: $x_i \vee x_j \Leftrightarrow z_k$, o sea $((x_1 \vee x_2) \Rightarrow z) \wedge$ $\wedge ((\overline{x_1 \vee x_2}) \Rightarrow \bar{z})$. Entonces se necesitarán dos -- cláusulas por elemento de la clase 1, ver - tabla 1.

Para cada elemento funcional de la clase 2, será preciso que todos los elementos puedan ser expresados como $x_j \vee z_i \Leftrightarrow z_k$. (Fig. 10). Ahora bien, si $z_i=1$, sin importar el valor de x_j, el resultado será $z_k=1$, por lo que - se tendrá $z_i \Rightarrow z_k \equiv \bar{z}_i \vee z_k$. Esto será una de las cláusulas necesarias para expresar - un elemento funcional de la clase 2.

Si $z_i=0$, el valor de z_k depende de x_j. Si - $x_j=1$ tenemos que $x_j \Rightarrow z_k$, y si $x_j=0$, $\bar{x}_j \Rightarrow z_k$. (Tabla 2).

Se ve pues que para representar un elemento funcional de la clase 2, se necesitan tres cláusulas.

Por su parte, los elementos de la clase 3 - serán de la forma $z_i \vee z_j \Leftrightarrow z_k$. Se ha de ver $(z_i \vee z_j \Rightarrow z_k) \wedge (z_i \vee z_j \Leftrightarrow z_k)$, lo que es equivalente a $((\overline{z_i \vee z_j}) \vee z_k) \wedge ((z_i \vee z_j) \vee \bar{z}_k)$. Usando la distributividad se puede ver que esto es equivalente a $((\bar{z}_i \wedge \bar{z}_j) \vee z_k) \wedge (z_i \vee$ $\vee z_j) \vee \bar{z}_k$, que a su vez es equivalente a -- $(\bar{z}_i \vee z_k) \wedge (\bar{z}_j \vee z_k) \wedge (z_i \vee z_j \vee \bar{z}_k)$, y por lo -- tanto ha de ser expresada también con tres cláusulas. La tabla correspondiente a los - elementos funcionales de la clase 3, es la tabla 3.

Se ve pues que para representar cualquier - elemento funcional en M, se puede hacer con un máximo de 3 cláusulas. El número total - de cláusulas, estará acotado por arriba por 3(p(n)), y claramente es polinómico en el - número de cláusulas. Como hay en las tres - tablas en cada columna un elemento diferente de 0, y no hay dos elementos distintos - de cero en filas complementarias, en la misma columna, y como bajo está condición las cláusulas son satisfactorias, hemos hallado un endomorfismo del input de un problema -- L∈NP, al input de satisfactoriedad, y las - cláusulas serán satisfactorias si, y sólo -

Tabla 1.

	C_1	C_2
z	$x_1\ x_2$	0
$\bar{z}$	0	$\overline{x_1\ x_2}$

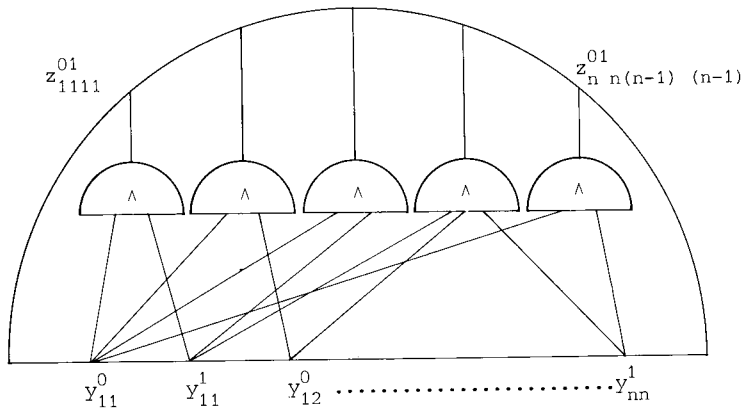


Fig. 9
MCD

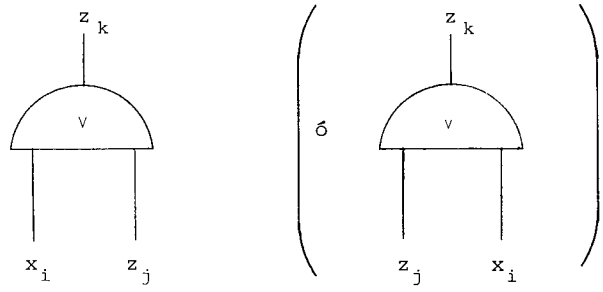


Fig. 10

Tabla 2.

	C_1	C_m	C_n
z_i	0	0	0
$\overline{z}_i$	1	0	0
z_k	1	x	0
$\overline{z}_k$	0	0	$\overline{x}$

Tabla 3.

	C_1	C_m	C_n
z_i	0	0	1
$\overline{z}_i$	1	0	0
z_j	0	0	1
$\overline{z}_j$	0	1	0
z_k	1	1	0
$\overline{z}_k$	0	0	1

si, la salida de la máquina M es 1. q.q.d.

Un corolario inmediato del teorema de Cook es el siguiente,

Corolario: P=NP, si y sólo si SATISFACTORIEDAD $\in$ P.

Demostración: Si satisfactoriedad $\in$ P, entonces por cada problema L tal que L$\in$NP, por el teorema de Cook L$\in$P, y por lo tanto NP $\subseteq$ P. Como P $\subseteq$ NP, P=NP. q.q.d.

Por tanto el problema de la satisfactoriedad es tan complejo como cualquier otro problema de la clase NP.

Un problema L pertenece a la clase NP-COMPLETA (o se dice ser NP-COMPLETO) si:

a) L $\in$ NP

b) SATISFACTORIEDAD $\propto$ L

Entonces el problema de la P-banda, como se ha expuesto anteriormente, satisface las dos condiciones de la definición, y por lo tanto es NP-completo.

Teorema 3: Si un problema NP-completo pertenece a la clase P, esto implica que todos los problemas NP-completos pertenecen a la clase P, y además P=NP. Por el contrario, si existe un problema NP-completo que no pertenezca a la clase P entonces ningún problema NP-completo pertenece a la clase P.

Demostración: Si L$\in$NP-completo y también L$\in$P, entonces por el teorema 1, satisfactoriedad$\propto$P, y por el corolario al teorema de Cook P=NP. Por otra parte, si hay un problema L que es NP-completo y que además L$\notin$P, entonces si hubiese otro problema NP-completo M, tal que M$\in$P, por el teorema 1 satisfactoriedad$\propto$P, y por la definición de NP-completo y la transitividad de $\propto$, se tendría que L$\in$P, contradicción. O sea que si un problema de la clase NP-completo no pertenece a la clase P, ninguno puede pertenecer. q.q.d.

El intento de probar que un cierto problema es o no NP-completo, ha creado algoritmos muy ingeniosos para casos particulares de esos problemas. Por ejemplo Garey y Johnson /9/ han probado que el problema de la enumeración mínima de un grafo es NP-completo, pero Y. Shiloach sacó un algoritmo para ese mismo problema, cuando el grafo es un árbol, que tiene complejidad polinómica, es decir pertenece a la clase P, en ese caso específico, (/10/). Otras veces, se han creado algoritmos llamados de aproximación, que "casi" resuelven el problema, y pertenecen a la clase P, mientras que el problema en si es NP-completo, (/11/). Una de las excepciones es el problema del viajante de comercio; la solución aproximada a este problema también pertenece a la clase NP-completa, (cap. VIII de /3/).

La gran pregunta, como se ha apuntado, es ¿P=NP?. Si fuera así, implicaría que todo tipo de problemas que se pueden resolver por métodos de "backtracking", (complejidad 2^n), se podrían resolver usando métodos con complejidad polinómica, lo cual choca contra la intuición matemática. Pero todavía nadie ha ofrecido una prueba formal a favor o en contra.

Sin embargo, quedan problemas de menor amplitud que el anterior, todavía por resolver. El autor ha probado que el problema de la enumeración mínima a la n, para cualquier n$\in$$Z^+$, es decir hallar la enumeración f_i de un grafo, tal que minimice $\Sigma|f_i(u)-f_i(v)|^n$, es también NP-completo. Cuando ese grafo tiene la estructura particular de un árbol, ¿es el problema P ó NP-completo? (se ha mencionado que cuando n=1 es P).

El siguiente teorema es una versión ampliada y puesta al día, del teorema de Karp, (/2/). Se enunciarán una serie de problemas, cada uno de ellos lleva la referencia en donde se demostró que son NP-completos. Esta lista no pretende ser completa, si no tan solo una muestra. Si algún lector está interesado en parte, o en todo el teorema, puede ponerse en contacto con el autor del artículo, o con el autor de la referencia.

Teorema 4: Los siguientes problemas pertenecen a la clase NP-completa:

SATISFACTORIEDAD /1/; SATISFACTORIEDAD CON UN MAXIMO DE 3 VARIABLES POR CLAUSULA /1/; BANDA /1/; SATISFACTORIEDAD DE K O MAS CLAUSULAS CON 2 VARIABLES POR CLAUSULA /9/; PROGRAMACION ENTERA BIVALENTE /2/; FLUJOS ENTE

ROS CON ARCOS HOMOLOGOS /12/; VER SI UN CIRCUITO COMBINATORIO ES NO REDUNDANTE /13/; - VER SI UN ERROR EN UNA LINEA DE ENTRADA, EN UN CIRCUITO COMBINATORIO PUEDE SER DETECTADO POR EXPERIMENTOS DE (INPUT/OUTPUT) /13/; VER SI UN CIRCUITO SATISFACE LA FUNCION BOOLEANA F /13/; ENCONTRAR EL CIRCUITO COMBINATORIO MINIMO QUE SATISFAGA F /13/; VER SI - UN GRAFO ES 3-COLOREABLE /9/; PUNTO DE EQUILIBRIO EN UN JUEGO-N /12/; NUMERO CROMATICO DE UN GRAFO /2/; ORDENACION LINEAL /14/; ORDENACION LINEAL RESTRINGIDA /14/; CORTE MAXIMO DE UN GRAFO /9/; LA ORDENACION LINEAR OPTIMA /9/; EL CIRCUITO HAMILTONIANO ORIENTADO /2/; EQUIVALENTE MINIMO DE UN GRAFO -- ORIENTADO /12/; LA MOCHILA /2/; PROGRAMACION CUADRATICA /12/; EL ARBOL RECTILINEO - DE STEINER /15/; PROBLEMA DE LA ENUMERACION MINIMA /9/; ENUMERACION MINIMA-N /16/; DADO UN GRAFO ORIENTADO HALLAR SOLUCION AL PROBLEMA DEL VIAJANTE DEL COMERCIO /17/; EL -- VIAJANTE DE COMERCIO EN UN GRAFO NO ORIENTADO /17/; SOLUCION APROXIMADA DEL VIAJANTE - DE COMERCIO /11/; MINIMIZAR $2^{d(S_i)}$, DONDE - d(S) ES LA FRONTERA DE S, SOBRE TODAS LAS - POSIBLES SECUENCIAS $S_0 \subseteq S_1 \subseteq S_2 \subseteq S_n$, DE - SUBCONJUNTOS DE LOS VERTICES DE UN GRAFO -- /16/.

7. RECONOCIMIENTO

El autor agradece las múltiples sugestiones y consejos recibidos del profesor Lawrence H. Harper, durante la realización de este - trabajo.

8. NOTAS

[1]El término original inglés es "clique". El Consejo Editor de QÜESTIIÓ ha optado por la traducción "banda", que ilustra la idea de elementos perfectamente comunicados unos -- con otros, sin jerarquías (por lo menos aparentes) y formando un grupo compacto frente al entorno.

9. REFERENCIAS

/1/ COOK, S., "The Complexity of Theorem-Proving Procedure". Theory of Computing, 1971, pp. 151-158.

/2/ KARP, R., "Reducidility among Combinatorial Problems. Complexity of Computer - Computations", Plenum Press, N.Y. 1976.

/3/ SAVAGE, J.E., "The Complexity of Computing", J. Wiley, 1976.

/4/ HARPER, L.H. y SAVAGE, J., "The Complexity of the Marriage Problem", Adv. in Math. 9, n° 3, 1972, pp. 299-312.

/5/ HARPER, L.H. y SAVAGE, J., "Complexity Made Simple", Proc. Colloquio Int. sulle Teorie Combinat. Roma. 1972. pp.555 -565.

/6/ HALMOS, P., "Naive Set Theory" Van Nostrand, 1960.

/7/ BERGE, C., "Théorie des Graphes et Applications", Dunod, 1958.

/8/ KALISH & MONTAGUE, "Logic, Technique et Formal Reasoning", Harcourt, Brace and World Inc. 1964.

/9/ GAREY, JOHNSON & STOCKMAYER, "Some Simplified NP-Complete Problems", Theoretical Computer Science 1, 1976, pp.237--267.

/10/ SHILOACH, Y., "Linear and Planar Arrangements of Graphs", Ph.D. Thesis, Dept. Applied Match., Weizmann I. Sc. 1976.

/11/ SAHNI, S. & GONZALEZ, T., "NP-Complete Approximation Problems", J.A.C.M. 23,3 1976, pp. 555-565.

/12/ SAHNI, S., "Computationally Related Problems", SIAM J. on Computing 1, 4, 1974, pp. 835-859.

/13/ IBARRA, O. & SAHNI, S., "NP-Complete -- Fault Detection Probl.", IEEE Transactions on Computers. 1974.

/14/ PAPADIMITRIU, Ch., "The NP-Completeness of the Bandwith", Computing, June 1976, pp. 263-270.

/15/ GAREY & JOHNSON, "The rectilinear Steiner Tree is NP-Complete", SIAM J. on -- Appl. Math. 32, 4. 1977. pp. 835-859.

/16/ DIAZ, J., Manuscrito en Preparación, 1978.

/17/ HARPER, L.H., "Comunicación Personal", 1976.