

**INSTITUTO TECNOLOGICO Y DE ESTUDIOS
SUPERIORES DE MONTERREY**



CAMPUS MORELOS

DIVISION DE INGENIERIA Y CIENCIAS

**HERRAMIENTAS DE VISUALIZACION PARA
INSTANCIAS DEL PROBLEMA DE
SATISFACIBILIDAD**

T E S I S
QUE PARA OBTENER EL GRADO DE
MAESTRO EN CIENCIAS DE LA
C O M P U T A C I O N
P R E S E N T A :
JUAN MANUEL GARCIA ORTEGON

QA
402.5
.G37
2000
c.2

ASESOR: DR. JOSE TORRES JIMENEZ

NOVIEMBRE, 2000

INSTITUTO TECNOLÓGICO Y DE ESTUDIOS
SUPERIORES DE MONTERREY



CAMPUS MORELOS

DIVISIÓN DE INGENIERÍA Y CIENCIAS

HERRAMIENTAS DE VISUALIZACIÓN PARA
INSTANCIAS DEL PROBLEMA DE
SATISFACIBILIDAD

TESIS
QUE PARA OBTENER EL GRADO DE MAESTRO
EN CIENCIAS DE LA COMPUTACIÓN

PRESENTA

JUAN MANUEL GARCÍA ORTEGÓN

NOVIEMBRE DE 2000

ASESOR:
DR. JOSÉ TORRES JIMÉNEZ

HERRAMIENTAS DE VISUALIZACION PARA INSTANCIAS DEL PROBLEMA DE SATISFACTIBILIDAD

por

Juan Manuel García Ortegón

Sometida al Programa de Graduados en Informática
en Cumplimiento Parcial de los Requerimientos para el Grado de

Maestro en Ciencias Computacionales

ITESM CAMPUS MORELOS Noviembre 2000

Aprobada por
Dr. José Torres Jiménez
Investigador ITESM Campus Morelos, Programa de Graduados en Informática
Asesor de Tesis

Revisada por
Dr. Carlos Gomez-Mont Avalos
Catedrático del ITESM Campus Morelos, Programa de Graduados en Informática
Revisor de Tesis

Revisada por
Dr. Juan Frausto Solís
Investigador ITESM Campus Morelos, Programa de Graduados en Informática
Revisor de Tesis

HERRAMIENTAS DE VISUALIZACION PARA INSTANCIAS DEL PROBLEMA DE SATISFACTIBILIDAD

por

Juan Manuel García Ortegón

Sometida al Programa de Graduados en Informática
en Cumplimiento Parcial de los Requerimientos para el Grado de

Maestro en Ciencias Computacionales

ITESM CAMPUS MORELOS Noviembre 2000

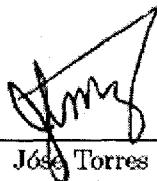
Aprobada por
Dr. José Torres Jiménez
Investigador ITESM Campus Morelos, Programa de Graduados en Informática
Asesor de Tesis

Revisada por
Dr. Carlos Gomez-Mont Avalos
Catedrático del ITESM Campus Morelos, Programa de Graduados en Informática
Revisor de Tesis

Revisada por
Dr. Juan Frausto Solís
Investigador ITESM Campus Morelos, Programa de Graduados en Informática
Revisor de Tesis


Herramientas de Visualización Para Instancias del Problema de Satisfactibilidad

Aprobada por:



Dr. José Torres Jiménez

Investigador ITESM Campus Morelos, Programa de Graduados en Informática
Asesor de Tesis



Dr. Carlos Gómez Mont Ávalos

Catedrático ITESM Campus Morelos, Programa de Graduados en Informática
Revisor de Tesis



Dr. Juan Faustino Solís

Investigador ITESM Campus Morelos, Programa de Graduados en Informática
Revisor de Tesis

Dedicatoria

A mis padres y hermana que siempre han estado a mi lado apoyándome en mis proyectos académicos. Gracias (una vez más).

Agradecimientos

Al Dr. José Torres, por su dirección en el desarrollo de esta tesis y en general por su apoyo durante toda la maestría.

A los Doctores Juan Frausto y Carlos Gomez-Mont, por su ayuda en la revisión de esta tesis y sus valiosos comentarios que la enriquecieron.

A mis compañeros estudiantes del Programa de Graduados en Informática, por su ayuda y comentarios respecto de este proyecto.

Finalmente, un agradecimiento muy especial para mis padres, que me apoyaron de todas las formas imaginables durante el curso de este posgrado.

HERRAMIENTAS DE VISUALIZACION PARA INSTANCIAS DEL PROBLEMA DE SATISFACTIBILIDAD

por

Juan Manuel García Ortégón

Sometida al Programa de Graduados en Informática
en Noviembre de 2000, en Cumplimiento Parcial de los
Requerimientos para el Grado de
Maestro en Ciencias Computacionales

Resumen

El Problema de Satisfactibilidad Proposicional (SAT) es un tópico frecuentemente referido en trabajos relacionados con complejidad computacional en parte porque es el ejemplo clásico de problema NP. Debido a su relación con problemas cruciales en disciplinas como la Inteligencia Artificial y la Optimización, se ha realizado un esfuerzo considerable para desarrollar métodos de solución con desempeños razonables. SAT es un problema NP-Completo, sin embargo puede ocurrir que cuando se realizan evaluaciones de desempeño con algunos algoritmos de solución, se observen tiempos de solución relativamente cortos. Esto se debe a una incorrecta selección de instancias de prueba. Para evaluar el real desempeño de un algoritmo que soluciona SAT, se debe poner especial atención en la selección de las instancias para asegurar que estas no tengan soluciones triviales, es decir, que sean duras. Tradicionalmente se han utilizado medidas de dureza resultado de estudios estadísticos. Tal es el caso, cuando se utilizan instancias SAT aleatorias, del cociente M/N (donde M es el número de cláusulas y N el número de variables). Sin embargo, una medida como la anterior nada dice de aspectos como las relaciones entre las variables que componen la instancia, entendiendo por variables relacionadas a aquellas que aparezcan juntas en una o más cláusulas. En esta tesis se explora la viabilidad de la utilización de herramientas de visualización en la determinación de la dureza de instancias SAT. Como resultado de este trabajo de investigación se presenta un conjunto de herramientas de visualización que pueden ser útiles en la comprensión de las condiciones que hacen dura a una instancia 3-SAT. Las técnicas de visualización contenidas en este trabajo están orientadas a evidenciar las relaciones entre las variables de un instancia SAT. Adicionalmente se incluyen técnicas que muestran la distribución de las variables y los signos en la instancia, así como una representación gráfica de la instancia misma. Usando esta información, el usuario puede identificar patrones asociados con la dureza de una instancia SAT, lo que ofrece parámetro extra en la identificación de la dureza además de la muy utilizada relación entre el número de cláusulas (M) y variables (N). La visualización de las relaciones entre variables puede incluso auxiliar en la selección del algoritmo de solución para instancias SAT específicas.

Índice General

1	Introducción	8
1.1	Estructura de la Tesis	9
1.2	Resumen del Capítulo	10
2	El Problema SAT	12
2.1	Definición del Problema SAT	12
2.1.1	Problema K-SAT	13
2.1.2	Problema Horn-SAT	13
2.2	Trascendencia del problema SAT	13
2.3	Métodos de Solución	14
2.3.1	Métodos Completos	14
2.3.2	Métodos Incompletos	16
2.4	Generadores de Instancias SAT	17
2.4.1	Generador de Selman	18
2.5	Resumen del Capítulo	18
3	Representaciones Gráficas del Problema SAT	19
3.1	Instancias SAT como Grafos	19
3.1.1	Grafos de instancias 2-SAT	19
3.1.2	Ambigüedad en Instancias 3-SAT	22
3.2	Instancias SAT como Hipergrafos	23
3.2.1	El modelo de Gallo	24

3.2.2	Cláusulas en las que se Repiten Combinaciones de Variables	26
3.3	Resumen del Capítulo	27
4	Visualización de Información	28
4.1	Cognición externa	28
4.2	Visualización de Información	29
4.2.1	Visualización Científica y Visualización de Información	30
4.2.2	Orígenes y evolución de la Visualización de Información	32
4.3	Amplificación Cognitiva	34
4.4	Mapeo de Datos a una Representación Visual	35
4.4.1	Características Deseables de una Representación Visual	35
4.4.2	Tablas de Datos	36
4.4.3	Estructuras Visuales	38
4.4.4	Transformaciones de Vistas	41
4.4.5	Interacción Humana	41
4.5	Resumen del Capítulo	41
5	Técnicas de Visualización Aplicadas al Problema SAT	43
5.1	Preliminares	43
5.2	Distribución de Signos y Variables	44
5.3	Relaciones entre dos Variables en una Cláusula	45
5.3.1	Versión 2D	47
5.3.2	Versión 3D	49
5.4	Relaciones entre tres Variables	49
5.5	Relaciones entre tres Variables Considerando Signos	50
5.6	Distribución de Combinaciones de Signos	54
5.7	Representación Gráfica de Instancias 3-SAT	54
5.8	Resumen del Capítulo	59

6 Ejemplos de Aplicación y Resultados	61
6.1 Identificación de Peculiaridades en los Generadores de Casos de Prueba	61
6.2 Herramientas de Visualización como Auxiliar en la Selección de Algoritmos de Solución	62
6.2.1 Algoritmos Genéticos y SAT	62
6.2.2 Instancias SAT adecuadas para AG	62
6.3 Resumen del Capítulo	64
7 Trabajos Futuros y Conclusiones	66
7.1 Ampliaciones y posibles modificaciones	66
7.2 Conclusiones	66

Índice de Figuras

3-1	Grafo equivalente para la instancia $(A \vee B) \wedge (A \vee C) \wedge (B \vee D) \wedge (C \vee D)$. . .	21
3-2	Grafo equivalente para la instancia $(A \vee -B) \wedge (-A \vee C) \wedge (-B \vee D) \wedge (C \vee -D)$	22
3-3	Ambigüedad en instancias K-SAT, cuando K es mayor que 2	23
3-4	Conexión de vértices en un A) Grafo, y en un B) Hipergrafo	24
3-5	Hipergrafo equivalente a una instancia SAT	25
3-6	Problemas de interpretación en un hipergrafo	26
4-1	Debilitamiento de la capa de ozono en el polo Norte (NASA).	30
4-2	El fenómeno de "El Niño" (NASA).	31
4-3	Flujo turbulento (SDSC).	31
4-4	Evolución de la economía de los Estados Unidos (SDSC).	32
4-5	Visualización de conversaciones basadas en WWW (MIT Media Lab)	33
4-6	Flujo de tráfico en Internet (Laboratorios Bell)	34
4-7	Esquema general de un sistema de visualización	37
4-8	Tabla de datos	37
4-9	Propiedades Retinales	40
5-1	Distribución de variables	45
5-2	Distribución de variables	46
5-3	Matriz de relaciones	46
5-4	Representación de relaciones entre dos variables	47
5-5	Relación entre dos variables utilizando la configuración de rombo	48
5-6	Direcciones de los ejes en la representación de rombo	48

5-7	Relaciones entre dos variables utilizando la representación 2D	50
5-8	Relaciones entre dos variables utilizando la versión 3D	51
5-9	Ejes correspondientes a la primera, segunda y tercera variable de una instancia 3-SAT	52
5-10	Relaciones de tres variables en una instancia, sin considerar los signos	53
5-11	Regiones (octantes) correspondientes dos combinaciones de signos	55
5-12	Relaciones entre tres variables considerando el signo	56
5-13	Distribución de signos en una instancia 3-SAT	57
5-14	Representación gráfica de una instancia SAT	58
5-15	Relocalización de los vértices del modelo de hipergrafo modificado.	59
6-1	Cadena binaria asociada a valores de verdad de variables proposicionales	63
6-2	Variables numéricamente cercanas	63
6-3	Variables numéricamente cercanas	64
6-4	Variables numéricamente cercanas (versión 3D)	65
6-5	Variables numéricamente cercanas (versión 3D con signos)	65

Capítulo 1

Introducción

Es difícil encontrar un problema NP-Completo más referenciado que el problema de Satisfactibilidad Proposicional (SAT). Es notable la gran cantidad de investigaciones realizada alrededor de SAT, lo que es evidencia de la importancia de este problema. Dicha importancia radica principalmente en dos aspectos:

- El problema SAT a probado ser un tópico de estudio crucial en el área de complejidad computacional [Garey79].
- El problema SAT tiene una estrecha relación con diversas áreas de las ciencias computacionales, como la Inteligencia Artificial, búsqueda y optimización, síntesis de circuitos electrónicos etc.

Por lo anterior se ha realizado un esfuerzo importante de investigación para encontrar métodos de solución con desempeños aceptables, es decir, que su costo para el promedio de los casos esté dentro de los rangos de aplicación. Así se han propuesto, y siguen apareciendo, algoritmos para resolver SAT.

No todos los algoritmos que resuelven SAT muestran el mismo nivel de desempeño. Para verificar la valía de un nuevo algoritmo para resolver SAT, la metodología común es comparar su desempeño contra el de otros algoritmos conocidos, mediante la realización de experimentos sobre un conjunto de instancias de prueba. Sin embargo, no todas las instancias del problema

SAT demuestran ser igualmente duras. El no prestar atención al proceso de selección de instancias para los experimentos, puede ocasionar que los resultados obtenidos no reflejen el real desempeño del algoritmo de solución analizado.

¿Entonces, como seleccionar las instancias para asegurar que sean suficientemente difíciles?. Tradicionalmente se han utilizado medidas de dureza resultado de estudios estadísticos[Cook97], tal es el caso del cociente del número de variables entre número de cláusulas (M/N)[Selman95]. Sin embargo, esta medida de dureza solo es aplicable a instancias producto de generadores de casos de prueba, ¿que ocurre con instancias que son resultado de problemas reales?. Una aplicación de las herramientas resultado de este trabajo, es la de servir como un medio adicional al cociente M/N , para inferir la dureza de una instancia.

La hipótesis de la que se parte es: la manera en la que están relacionadas las variables en una instancia SAT, tiene relación con su dureza. Identificar estas relaciones (y especialmente los patrones que forman) automáticamente no es una tarea sencilla. Esta es la motivación para utilizar herramientas de visualización.

El cerebro humano muestra una alta capacidad para extraer información de datos, y relaciones entre ellos, cuando le son presentados visualmente. El objetivo primordial de esta tesis es construir herramientas que le permitan al usuario utilizar su habilidad natural para interpretar información visual, en la búsqueda de patrones. Estos patrones pueden ser asociados con la dureza de una instancia, e incluso pueden ser un medio para determinar con que algoritmo de solución debe atacarse un conjunto de problemas específicos, lo que es especialmente atractivo cuando se trabaja con instancias SAT utilizadas para representar problemas reales.

1.1 Estructura de la Tesis

El presente reporte se compone de siete capítulos, a continuación se dará una breve descripción de los tópicos que cada uno de ellos cubre.

Capítulo 2. El Problema SAT

Se discuten aspectos generales del problema SAT, tales como una definición formal del problema, la conexión de SAT con diferentes disciplinas de las Ciencias Computacionales así

como los tipos de métodos de solución con sus respectivos ejemplos. También se explica el concepto de dureza y se revisa un ejemplo de generador de casos de prueba (generador de Selman), que en la literatura se reconocen como duro y que es comúnmente utilizado.

Capítulo 3. Representaciones Gráficas del Problema SAT

Se justifica la utilidad de tener una representación gráfica para instancias del problema SAT. Se discute la implementación de dicho modelo de representación utilizando grafos. Se ponen en evidencia las limitaciones del modelo basado en grafos y se presenta como alternativa el uso de hipergrafos.

Capítulo 4. Visualización de Información

En este capítulo se presenta un panorama del área de Visualización de Información. Se discuten aspectos tales como, la utilización de representaciones visuales para extender la capacidad cognocitiva del ser humano, el desarrollo histórico de la visualización, diferencias entre visualización científica y visualización de información. Se concluye el capítulo con una breve introducción de los fundamentos de la visualización de la información.

Capítulo 5. Técnicas de Visualización Propuestas

Es el capítulo central de la tesis y en él se presentan las técnicas de visualización desarrolladas durante este proyecto de investigación. Se describe tanto el fundamento de las técnicas como su implementación. También se menciona el tipo de información que se puede extraer de las representaciones visuales.

Capítulo 6. Ejemplos de Aplicación y Resultados

Aún cuando el objetivo de la tesis no es el descubrimiento de patrones, sino el desarrollo de herramientas con las que éstos puedan evidenciarse, en este capítulo se presentan algunos ejemplos de aplicación que se han identificado durante el desarrollo del proyecto.

Capítulo 7. Trabajo Futuros y Conclusiones

Se presentan propuestas de extensiones al proyecto además de las conclusiones de la tesis.

1.2 Resumen del Capítulo

Se ha realizado un gran esfuerzo en la búsqueda de soluciones apropiadas a instancias del problema SAT. Parte fundamental de este esfuerzo es la caracterización, mediante experimentos,

del desempeño de algoritmos de solución. Ya que los tiempos de solución de los problemas son dependientes de la dureza de la instancia con que se pruebe, es deseable tener una estimación de la dureza. Se pueden utilizar técnicas de visualización para descubrir patrones en instancias SAT y estos patrones podrían ser útiles para estimar la dureza.

Capítulo 2

El Problema SAT

2.1 Definición del Problema SAT

El Problema de Satisfactibilidad Proposicional (SAT) se define como: Dado un conjunto de cláusulas del cálculo proposicional C_1, C_2, \dots, C_M sobre las variables V_1, V_2, \dots, V_N , el problema de Satisfactibilidad es el determinar si para la fórmula $C_1 \wedge C_2 \wedge \dots \wedge C_M$ expresada en Forma Normal Conjuntiva (FNC), existe una asignación de valores de verdad para sus variables que haga verdadera la expresión completa. SAT fue el primer problema identificado como NP-Completo[Cook71]. Es por esto que, para el peor caso, los tiempos de solución utilizando los algoritmos conocidos crecen más rápidamente que cualquier polinomio respecto al tamaño de la entrada.

En el resto de este reporte se referirá con la letra N al número de variables de una instancia, con M se hará referencia al número de cláusulas, y con K al número de literales por cláusula. Si no se especifica otra cosa, se asumirá que $K = 3$. En los próximos capítulos se hará referencia al tamaño de la entrada en un problema SAT, dicho tamaño está definido por el número de variables y el número de cláusulas que forman la instancia.

Existen versiones restringidas del problema SAT. Debido a su relación con las herramientas de visualización que se presentan en esta tesis, se definen dos de ellas.

2.1.1 Problema K-SAT

La restricción de SAT a instancias donde todas las cláusulas tienen la misma longitud (número de literales) se denomina K -SAT. Es interesante destacar que 3 es el valor más pequeño de K para el cual el problema K -SAT sigue siendo NP-Completo[Cook71]. Es también importante notar que cualquier instancia de problema SAT puede ser transformada a una correspondiente instancia 3-SAT en tiempo polinomial[Garey79].

2.1.2 Problema Horn-SAT

Horn-SAT es la restricción del problema a instancias donde cada cláusula tiene como máximo una literal no negada. Horn-SAT es interesante ya que existen métodos para resolverlo en tiempo polinomial [DOWLING84].

2.2 Trascendencia del problema SAT

Son muchos los aspectos que hacen de SAT un problema de interés en el ámbito de las Ciencias Computacionales. En las áreas de computación teórica, SAT es muchas veces utilizado como el ejemplo clásico de problema NP-Completo, ya que fue el primero en ser identificado como tal, además de ser uno de los más simples. Esto lo coloca en una posición central en el estudio de la complejidad computacional[Garey79]. SAT tiene una conexión estrecha con problemas cruciales de la Inteligencia Artificial (IA), debido a su relación con el razonamiento deductivo. Por ejemplo, considérese lo siguiente: Dada una colección de hechos base Σ , una expresión α puede ser deducida de Σ si y solo si $\Sigma \cup \{\neg\alpha\}$ es no satisfactible. Finalmente, muchos problemas reales de áreas como la inferencia, aprendizaje, satisfacción de restricciones y la síntesis de circuitos VLSI[Devadas89], por mencionar algunos ejemplos, son NP-Completo y pueden ser transformados de manera eficiente a SAT.

Por todo lo anterior se ha realizado un gran esfuerzo para desarrollar métodos de solución con desempeños razonables. Para estos problemas, razonable significa que su desempeño sea bueno en promedio o con alta probabilidad. Razonable también puede significar que el método de solución trabaje bien con un subconjunto interesante de instancias (instancias generadas por

problemas reales, por ejemplo). No se ha dicho la última palabra en cuanto a métodos de solución del problema SAT, se siguen presentando nuevos métodos y se espera que la investigación al respecto continúe.

2.3 Métodos de Solución

Desde el punto de vista de la funcionalidad, los métodos de solución de SAT se pueden agrupar en dos categorías, completos e incompletos. Un método es completo si utilizándolo se puede determinar, en todos los casos, si una instancia es satisfactible o no lo es. Los métodos incompletos solo pueden determinar la condición de satisfactibilidad en algunas instancias, es decir, si no se encuentra solución aplicándolo, no significa que para la correspondiente instancia no exista solución alguna. La utilización de una clase de método u otro depende usualmente del tamaño de los problemas en los que se aplicará y del tiempo disponible para obtener una respuesta, ya que el desempeño de los métodos completos se ve rápidamente afectado con el aumento en el tamaño de la entrada.

2.3.1 Métodos Completos

A continuación se presenta dos ejemplos de métodos completos de solución de SAT.

El Método de Davis y Putnam

Davis y Putnam es uno de los métodos de solución más utilizados. Consiste en el uso de cuatro reglas que permiten realizar la comprobación de la insatisfactibilidad de un conjunto de cláusulas. Aplicando las cuatro reglas es posible reducir la cantidad de cláusulas hasta llegar a tener una fórmula que es válida o que es insatisfactible y que contiene el mínimo número de cláusulas [Davis60].

Considérese lo siguiente:

Sea S un conjunto de cláusulas.

1. *Regla de la Tautología:* Elimine todas las cláusulas de S que son tautologías. El conjunto resultante S' es insatisfactible si y sólo si S lo es.

2. *Regla de la Literal Unitaria:* Si existe una cláusula unitaria L en S , obtenga S' de S al eliminar aquellas cláusulas en S que contengan L . Si S' está vacío, S es satisfactible. De lo contrario, obtenga un conjunto S'' de S' al eliminar $\sim L$. S'' es insatisfactible si y sólo si S lo es. Note que si $\sim L$ es una cláusula unitaria, entonces la cláusula se convierte a una cláusula vacía cuando $\sim L$ es eliminada de la cláusula.
3. *Regla de la Literal Pura:* Una literal L en una cláusula de S , se dice que es pura en S si y sólo si la literal $\sim L$ no aparece en ninguna cláusula en S . Si una literal L es pura en S , elimine todas las cláusulas que contengan L . El conjunto S' que resulta de esto, es insatisfactible si y sólo si S lo es.
4. *Regla de Expansión:* Si el conjunto S puede ser puesto en la forma: $(A_1 \vee L) \wedge \dots \wedge (A_m \vee L) \wedge (B_1 \vee \sim L) \wedge \dots \wedge (B_n \vee \sim L) \wedge R$, donde A_i, B_i , y R están libres de L y $\sim L$, entonces se obtienen los conjuntos $S_1 = A_1 \wedge \dots \wedge A_m \wedge R$ y $S_2 = B_1 \wedge \dots \wedge B_n \wedge R$. S es insatisfactible si y sólo si $(S_1 \vee S_2)$ es insatisfactible, esto significa que S_1 y S_2 también lo son.

Las reglas son aplicadas iterativamente hasta determinar si la instancia es satisfactible o no.

Principio de Resolución

El principio de resolución se basa en la verificación de que un conjunto de cláusulas S contiene a la cláusula vacía. Si efectivamente S contiene la cláusula vacía, entonces S es insatisfactible. Si no la contiene, entonces se debe determinar si a partir de S se puede derivar la cláusula vacía.

De acuerdo con el teorema de Herbrand, verificar la presencia de una cláusula vacía es equivalente a contar el número de nodos de un árbol semántico cerrado para S [Chang87]. Este teorema establece que S es insatisfactible, si y sólo si existe un árbol semántico cerrado finito T para S . Se puede deducir fácilmente que S contiene la cláusula vacía, si y sólo si T consiste de un sólo nodo (el nodo raíz). Si S no contiene la cláusula vacía, entonces T , tiene más de un nodo. Sin embargo, si se reduce el número de nodos a uno, en algún momento la cláusula vacía puede ser forzada a aparecer. Esto es exactamente lo que el principio de resolución permite realizar.

Se puede ver al principio de resolución como una regla de inferencia que permite generar nuevas cláusulas a partir de S . Si se colocan nuevas cláusulas dentro de S , algunos nodos del conjunto original se estarán forzando a convertirse en nodos con falla. De esta forma el número de nodos en T puede ser reducido a uno y por lo tanto la cláusula vacía se presentará.

El principio de resolución, es en esencia una extensión de la regla de la literal unitaria, que propusieron Davis y Putnam. Esta regla lo que indica es verificar si existen dos literales en las cláusulas que formen un par complementario, para posteriormente eliminarlo.

Lo que realiza el principio de resolución es extender la regla anterior para cualquier par de cláusulas (no necesariamente cláusulas unitarias), con lo que surge la siguiente regla en la cual se expresa formalmente el Principio de Resolución:

Para cualquier par de cláusulas C_1 y C_2 , si hay una literal L_1 en C_1 que es complementaria a una literal L_2 en C_2 , entonces debe eliminarse L_1 y L_2 de C_1 y C_2 respectivamente, y construir la disyunción de las cláusulas restantes. La cláusula resultante es un resolvente de C_1 y C_2 .

Si se tienen dos cláusulas unitarias, entonces el resolvente de ellas es la cláusula vacía. Más importante aún, si un conjunto S de cláusulas es insatisfactible se puede, a partir del principio de resolución, generar la cláusula vacía a partir de S .

El principio de resolución es una poderosa regla de inferencia, y se aplica en muy diversas áreas de investigación. Un punto muy importante es que el principio de resolución es completo al probar la insatisfactibilidad de un conjunto de cláusulas por refutación. Esto significa, que siempre se podrá generar la cláusula vacía a partir de un conjunto insatisfactible de cláusulas.

2.3.2 Métodos Incompletos

A continuación se presenta un ejemplo de método incompleto de solución de SAT.

Algoritmo G-SAT

G-SAT es un procedimiento de búsqueda local para resolver problemas de satisfactibilidad [Selman92]. G-SAT se inicia con una asignación aleatoria de valores de verdad, se cuantifica el número de cláusulas satisfechas con estas condiciones y se cambia el valor de verdad de la variable que

provoque que el mayor aumento en el número de cláusulas satisfechas. Estos cambio son repetidos un número máximo de veces (Max_Cam), después de los cuales se genera una asignación de valores de verdad totalmente nueva. Esto se repite hasta que se encuentre una asignación que satisfaga la expresión o hasta que se alcance un número máximo de reinicios (Max_Ini).

Procedimiento G-SAT

Entradas: Un conjunto de cláusulas en FNC, Max_Cam , Max_Ini

Salidas: Una asignación de valores de verdad que satisfagan la expresión (si la encuentra)

Desde $i=1$, **hasta** Max_Ini

T= Asignación aleatoria de valores de verdad

Desde $i=1$, **hasta** Max_Cam

Si T satisface la expresión, termina

p = Una variable proposicional cuyo cambio de valor provoca el mayor aumento de cláusulas satisfechas en T

T = T con el cambio en el valor de verdad de p

Fin del ciclo

Fin del ciclo

No se encontró asignación de valores que satisfagan la expresión.

Selman reporta que utilizando G-SAT es posible atacar problemas SAT un orden de magnitud más grandes que los problemas más grandes que pueden ser resueltos con métodos como el de Davis y Putnam o el principio de Resolución.

2.4 Generadores de Instancias SAT

Como se ha mencionado, 3-SAT es un problema NP-Completo. Sin embargo, estudios experimentales del problema aplicando diferentes algoritmos han reportado tiempos de solución relativamente cortos[Goldberg79]. Este fenómeno no significa que SAT sea un problema de fácil solución en el promedio de los casos si no qué, como se demuestra en otros trabajos experimentales[Cook97], la distribución de instancias utilizadas afectan notablemente los resultados. Se han realizado mucho trabajo alrededor de las condicione que hacen dura a una

instancia[Chabrier96][Horic97][Watanabe93], y la investigación continúa. Para evaluar efectivamente el desempeño de un algoritmo de solución, se debe asegurar que las instancias con las que es probado no sean triviales.

2.4.1 Generador de Selman

Uno de los generadores de casos de prueba más utilizados es el llamado generador de instancias SAT aleatorias[Selman95].

Una instancia SAT aleatoria es una fórmula proposicional en FNC obtenida mediante la generación de M cláusulas aleatorias de K literales. Las K literales son generadas seleccionando aleatoriamente K variables de un total de N . Cada variable en la cláusula tiene una probabilidad de 0.5 de aparecer negada. Siguiendo estas reglas para generar las instancias, se espera que las instancias duras aparezcan cuando el cociente M/N se encuentre alrededor de 4.25.

2.5 Resumen del Capítulo

SAT fue el primer problema identificado como NP-Completo, y desde entonces es una referencia obligada en trabajos relacionados con complejidad computacional. Su estrecha relación con múltiples áreas de las Ciencias Computacionales hacen de SAT un problema particularmente interesante. Se han propuesto varios métodos de solución de SAT que pueden ser clasificados desde un punto de vista funcional en completos e incompletos. A la par del desarrollo de métodos de solución, se han propuesto generadores de casos de prueba que produzcan instancias sin soluciones triviales.

Capítulo 3

Representaciones Gráficas del Problema SAT

En este capítulo se presentan algunas consideraciones relacionadas con la representación de instancias SAT utilizando algún medio gráfico. El utilizar un esquema gráfico para representar instancias del problema SAT tiene al menos dos aplicaciones:

- Permite buscar soluciones a SAT, resolviendo el problema gráfico equivalente[Gallo92].
- Permite evidenciar información propia de la instancia (como las relaciones entre variables).

Es la segunda vertiente de aplicaciones la que se busca explotar en este trabajo. En las secciones siguientes se describirá una propuesta de representación de instancias SAT, así como los antecedentes que le dieron origen.

3.1 Instancias SAT como Grafos

Los grafos son uno de los esquemas de representación más sencillo que se pueden utilizar para este problema, de manera que fueron los primeros en ser probados.

3.1.1 Grafos de instancias 2-SAT

Considérese el caso especial de instancias K -SAT con $K = 2$.

Una instancia 2-SAT $I_2 = (X, C)$ puede definirse por un conjunto X de N variables proposicionales, que pueden ser verdaderas o falsas, y un conjunto C de M cláusulas. Las cláusulas tienen la forma:

$$x_a \vee x_b \text{ en donde } x_a \text{ y } x_b \in X.$$

Un Grafo G es un par $G = (V, E)$ en donde V es un conjunto de vértices y E es un conjunto de arcos. Un arco e es un par $e = (v_a, v_b)$, en donde v_a y $v_b \in V$.

Aplicando estas dos definiciones se puede decir que, para encontrar un grafo equivalente $G_I(V, E)$ de una instancia 2-SAT $I_2(X, C)$, se debe:

- Crear un conjunto V , que tiene un vértice v por cada variable contenida en X .
- Crear un conjunto E , que tiene un arco e por cada cláusula en C que conecta los vértices correspondientes a las variables en la cláusula.

Aplicando lo anterior a la instancia:

$$(A \vee B) \wedge (A \vee C) \wedge (B \vee D) \wedge (C \vee D)$$

Se obtiene el grafo equivalente que se muestra en la figura 3-1.

Evidentemente, con el esquema anterior no es posible hacer distinción entre las variables que aparecen negadas de las que aparecen no negadas en la instancia. Esto se puede resolver utilizando grafos dirigidos.

Un grafo dirigido DG es un par $DG = (V, E)$ en donde V es un conjunto de vértices y E es un conjunto de arcos dirigidos. Un arco dirigido de es un par $de = (Cola(de), Cabeza(de))$, en donde $Cola(de)$ y $Cabeza(de) \in V$.

También debemos modificar la definición de instancia 2-SAT, agregando al conjunto de variables, una constante F que siempre se evalúa como falsa y una constante T que siempre se evalúa como verdadera.

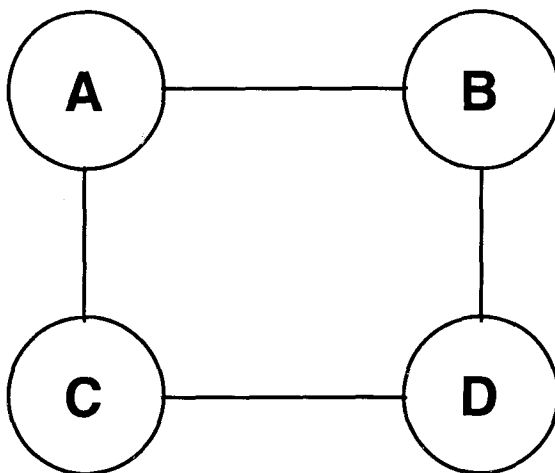


Figura 3-1: Grafo equivalente para la instancia $(A \vee B) \wedge (A \vee C) \wedge (B \vee D) \wedge (C \vee D)$

Utilizando lo que se ha definido se puede decir que, para encontrar un grafo dirigido equivalente $DG_I(V, E)$ de una instancia 2-SAT $I_2(X, C)$, se debe:

- Crear un conjunto V , que tiene un vértice v por cada variable contenida en X , además de un vértice T y un vértice F .
- Crear un conjunto E , que tiene un arco dirigido de por cada cláusula en C que tiene una sola variable negada. $Cola(de)$ será igual al vértice correspondiente a la variable negada de la cláusula y $Cabeza(de)$ será igual al vértice correspondiente a la variable no negada en la cláusula.
- Si existen cláusulas con solo variables no negadas, se crean dos arcos dirigidos de en E cuyos valores de $Cola(de)$ son iguales al vértice T , y sus valores de $Cabeza(de)$ son iguales a los vértices correspondientes a las variables en la cláusula.
- Si existen cláusulas con solo variables negadas, se crean dos arcos dirigidos de en E cuyos valores de $Cola(de)$ son iguales a los vértices correspondientes a las variables en la cláusula y sus valores de $Cabeza(de)$ son iguales al vértice F .

Aplicando lo anterior a la instancia:

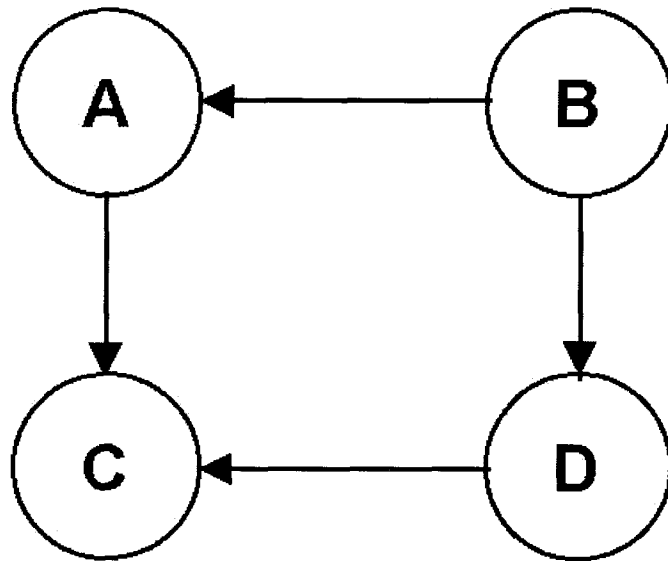


Figura 3-2: Grafo equivalente para la instancia $(A \vee -B) \wedge (-A \vee C) \wedge (-B \vee D) \wedge (C \vee -D)$

$$(A \vee -B) \wedge (-A \vee C) \wedge (-B \vee D) \wedge (C \vee -D)$$

Se obtiene el grafo dirigido equivalente que se muestra en la figura 3-2.

3.1.2 Ambigüedad en Instancias 3-SAT

Las definiciones presentadas en la sección anterior pueden generalizarse para cláusulas de cualquier tamaño (número de literales). Por ejemplo, si se quiere representar una instancia 3-SAT se necesitaría utilizar al menos tres vértices y dos arcos por cláusula. En general podemos decir que se necesitan al menos K vértices y $K - 1$ arcos por cada cláusula K-SAT que se quiera representar con un grafo.

Sin embargo, el problema surge cuando se trata de extraer información de un grafo dirigido y no se conoce el valor de K de su instancia K-SAT asociada. Para ejemplificar esto, obsérvese la figura 3-3.

La información que se muestra en la figura 3-3 no está libre de ambigüedades, ¿son dos cláusulas 2-SAT o una cláusula 3-SAT?, si no se conoce de antemano el tipo de instancia K-SAT con la que se creó el grafo es imposible decidirlo. Aún más, ¿cómo podemos utilizar este

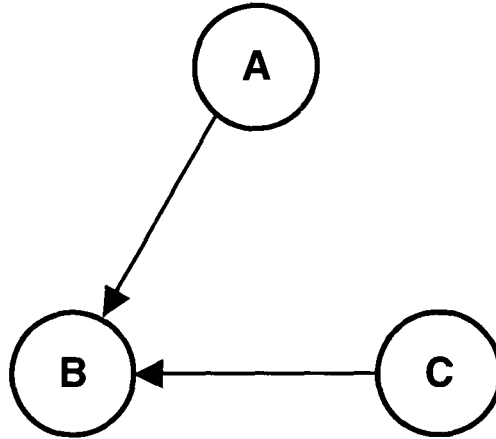


Figura 3-3: Ambigüedad en instancias K-SAT, cuando K es mayor que 2

esquema de grafos para representar instancias de problemas más generales de SAT (donde cada cláusula tiene diferente longitud, por ejemplo)?.

En la siguiente sección se discute una solución a este problema, la utilización de hipergrafos.

3.2 Instancias SAT como Hipergrafos

Si el problema para representar instancias K-SAT, cuando K es mayor que 2, es que la representación de la cláusula involucra más de un arco, ¿por que no buscar la solución precisamente en esos términos?, buscar que sea siempre el arco el que indique la extensión de la cláusula. Evidentemente esto no es posible utilizando grafos, lo que lleva a la exploración de la aplicación de hipergrafos.

Un hipergrafo dirigido, es una generalización al modelo de grafos dirigidos, y difiere de estos últimos en la manera en que se conectan sus vértices. Mientras que el arco de un grafo conecta como máximo dos vértices, el hiperarco de un hipergrafo puede conectar un vértice con todo el resto.

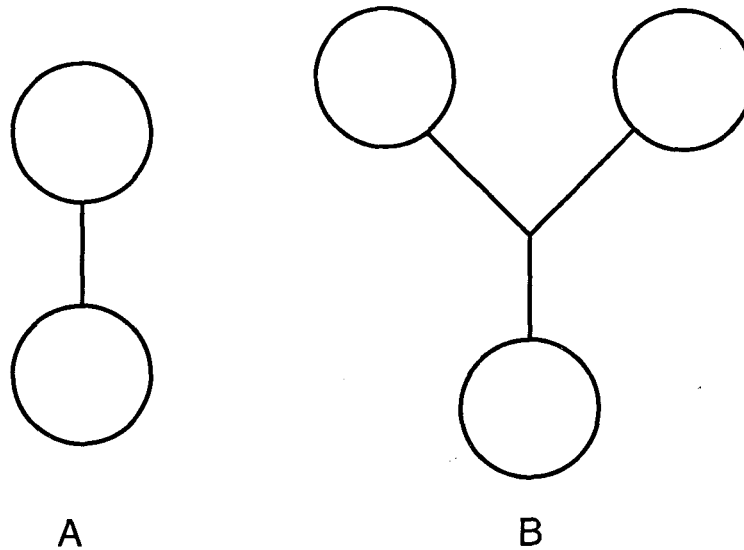


Figura 3-4: Conexión de vértices en un A) Grafo, y en un B) Hipergrafo

3.2.1 El modelo de Gallo

Gallo fue uno de los primeros en proponer el uso de hipergrafos dirigidos como representación gráfica del problema SAT[Gallo93].

Antes de continuar, conviene mostrar una definición formal de hipergrafo dirigido.

Un hipergrafo dirigido es una pareja $H = (V, E)$ donde V es un conjunto de vértices y E es un conjunto de hiperarcos. Un hiperarco e , es un par $(Cola(e), Cabeza(e))$, donde $Cola(e)$ y $Cabeza(e)$ son dos subconjuntos disjuntos y no vacíos de V .

Para generar el hipergrafo $H_I(V, E)$ asociado a una instancia SAT $I(X, C)$, se hace lo siguiente:

- Se crea un conjunto V que contiene un vértice v por cada variable en X , además de un vértice para la constante T y un vértice para la constante F .
- Se crea un conjunto E que contiene un hiperarco e por cada cláusula que contenga variables con diferentes signos y cuyo subconjunto $Cola(e)$ contiene los vértices asociados a las variables que aparecen negadas en la cláusula. El subconjunto $Cabeza(e)$ correspondiente contiene los vértices asociados a las variables que aparecen no negadas en la cláusula.

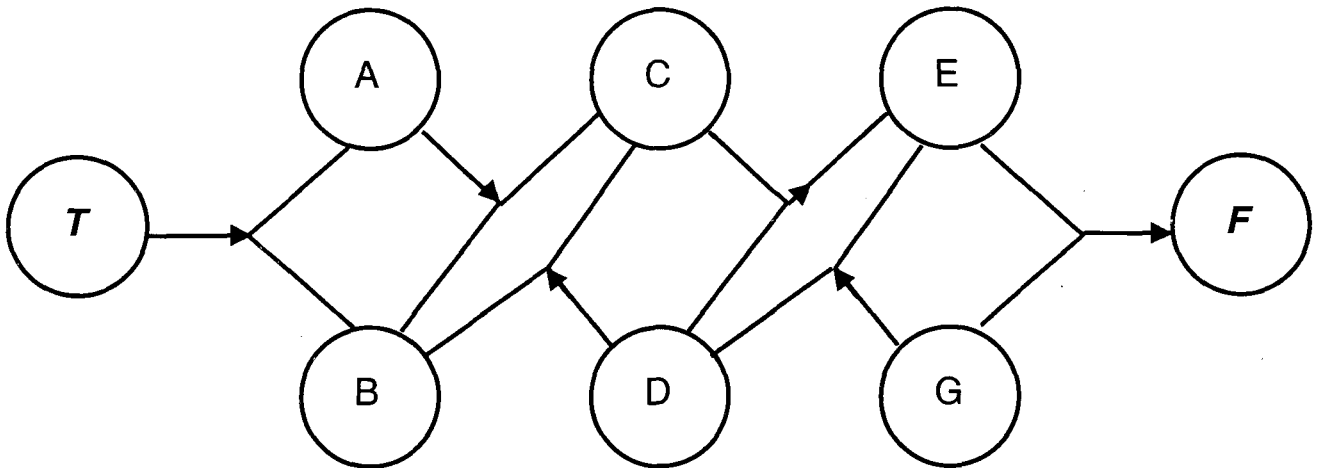


Figura 3-5: Hipergrafo equivalente a una instancia SAT

- Por cada cláusula con solo variables no negadas (si existen), se crea un hiperarco e cuyo subconjunto $Cola(e)$ contiene el vértice T y su subconjunto $Cabeza(e)$ contiene a los vértices correspondientes a las variables en la cláusula.
- Por cada cláusula con solo variables negadas (si existen) se crea un hiperarco e cuyo subconjunto $Cola(e)$ contiene a los vértices correspondientes a las variables en la cláusula y su subconjunto $Cabeza(e)$ contiene al vértice F .

Aplicando lo anterior sobre la instancia:

$$(-A \vee B \vee C) \vee (-E \vee -G) \vee (-C \vee -D \vee E) \vee (C \vee B \vee -D) \vee (A \vee B) \vee (-D \vee -E \vee G)$$

Se obtiene el hipergrafo equivalente que se muestra en la figura 3-5.

Utilizando esta representación basada en hipergrafos es posible representar cualquier instancia SAT expresada en FNC, independientemente de si se trata de una instancia K-SAT o no. Adicionalmente, no existe ambigüedad al interpretar los hipergrafos resultantes. Es por lo anterior que se seleccionó el modelo de hipergrafo para visualizar instancias SAT completas.

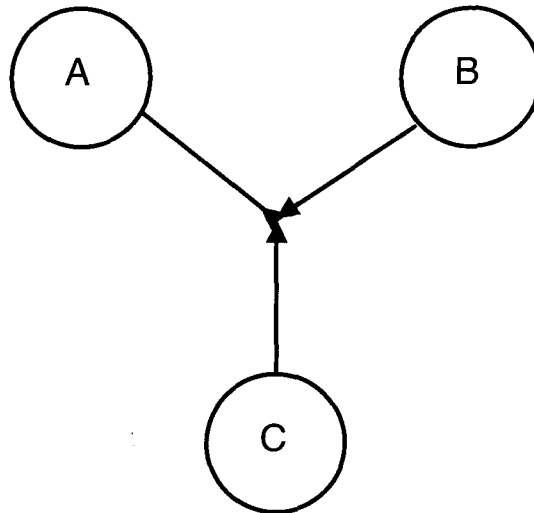


Figura 3-6: Problemas de interpretación en un hipergrafo

3.2.2 Cláusulas en las que se Repiten Combinaciones de Variables

Considérese la siguiente instancia SAT:

$$(-A \vee B \vee C) \wedge (-A \vee B \vee -C) \wedge (-A \vee -B \vee C)$$

Su correspondiente hipergrafo sería el mostrado en la figura 3-6.

De lo anterior es fácil notar que, cuando existen varias cláusulas que comparten las mismas variables, la interpretación del grafo se dificulta. Solo imagine una instancia en la que se presentara la combinación de variables que se muestra a continuación:

$$(-A \vee B \vee C) \wedge (-A \vee B \vee -C) \wedge (-A \vee -B \vee C) \wedge (A \vee B \vee -C) \wedge (A \vee -B \vee C) \wedge (A \vee -B \vee -C)$$

El modelo de hipergrafos es suficiente para representarlo, pero el resultado puede ser ilegible. Es por esto que al momento de implementar el modelo de visualización, se debe buscar una codificación que facilite la interpretación del grafo. En la sección 5.7 de este documento se abordará el tema.

3.3 Resumen del Capítulo

Es útil representar instancias del problema SAT en forma gráfica, ya que permite la búsqueda de su solución resolviendo un problema gráfico equivalente, además de poder identificar visualmente características de la instancia, como las relaciones entre las variables. De los modelos explorados, el de hipergrafos mostró ser el mejor para representar instancias SAT.

Capítulo 4

Visualización de Información

Solemos utilizar expresiones como "ya veo", cuando finalmente logramos entender una idea. Este tipo de expresión es tan frecuentemente utilizada, que da una pista de la estrecha relación entre lo que vemos y lo que pensamos. Los seres humanos, mostramos una tendencia a asociar representaciones gráficas de problemas con sus soluciones, y no lo hacemos por capricho o accidente.

Norman menciona que las capacidades de la mente humana por si sola, han sido sobrestimada. En efecto, sin ayudas externas, nuestras capacidades de razonamiento y memoria son limitadas. Es por esto que se han diseñado ayudas externas que le permiten a la mente rebasar estos límites. Una importante categoría de estas ayudas a la cognición son la herramientas visuales.

4.1 Cognición externa

Antes de entrar en los detalles propios de la visualización de información, conviene entender la importancia que tiene la percepción del mundo externo en el razonamiento. Esta noción, algunas veces referida como "cognición externa", expresa la manera en la cual las representaciones externas e internas producen, mediante el apropiado procesamiento, conceptos o ideas en la mente, es decir, pensamientos. El humano utiliza, natural y cotidianamente, esta combinación de factores, aún más, los humanos diseñan medios (que no residen en la mente) que potencian la capacidad cognitiva.

Existen múltiples ejemplos de herramientas de cognición externa, baste hacer referencia a los métodos que se utilizan para realizar operaciones aritméticas. Realizar una multiplicación de cifras con más de tres dígitos, sin ayuda externa es mucho más complicado que hacerlo con el adecuado suministro de papel y lápiz. La dificultad de hacerlo en la mente no se debe a que multiplicar sea una tarea compleja, simplemente es un problema de memoria. Si se hace uso del papel para almacenar los resultados parciales, realizar una operación como la multiplicación es sumamente sencillo. El ejemplo se puede extender a operaciones realizadas con cualquier otro dispositivo (como las regla cálculo y aún las calculadoras).

Como se mencionó, algunas de las principales herramientas de cognición externa, tienen carácter visual. En la siguiente sección se discutirá la forma en la que se puede utilizar una representación visual de información en la solución de problemas.

4.2 Visualización de Información

Existen muchas actividades relacionadas con la creación de representaciones visuales, es necesario aclarar como se relacionan estas actividades para poder colocar a la visualización de información en el contexto adecuado.

Comenzaremos con el concepto mismo de visualización, que se define, en términos de su propósito, como [Card99]:

Visualización: Es el uso de representaciones computacionales, visuales e interactivas sobre datos para ampliar la cognición.

Para ampliar este concepto, a continuación se muestran los objetivos principales de la visualización. Aplicando representaciones visuales a los datos se puede:

- *Descubrir*. Visualizando datos se puede encontrar explicación a fenómenos.
- *Tomar decisiones*. Se pueden hacer juicios y tomar decisiones con la información que se extrae de la representación visual.
- *Explicar*. La visualización, puede ser un mecanismo muy poderoso para la transmisión de ideas, recuérdese que: Una imagen dice más que mil palabras.

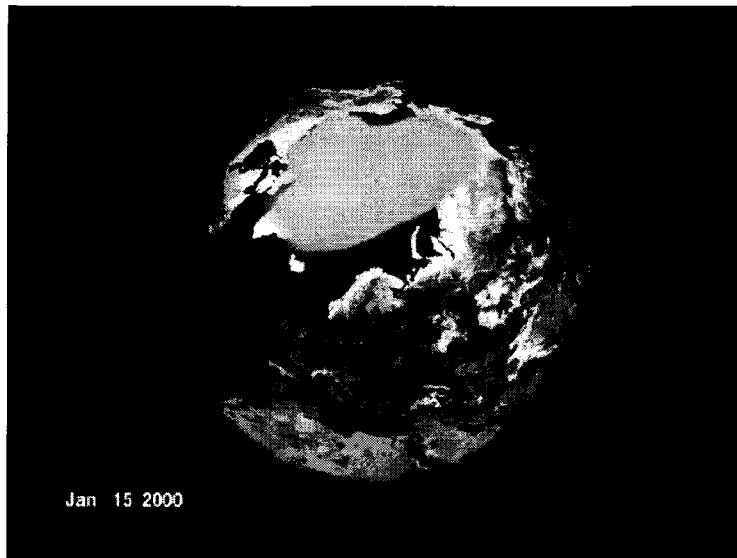


Figura 4-1: Debilitamiento de la capa de ozono en el polo Norte (NASA).

4.2.1 Visualización Científica y Visualización de Información

Se puede decir que la visualización científica es el origen de la visualización como un campo organizado. La visualización científica se concibe como una herramienta que permite manejar grandes volúmenes de datos científicos. A pesar de que no es necesariamente el origen de la concepción, la visualización científica tiende a estar basada en datos físicos (campos de velocidades, temperaturas, presión, etc.). En las figuras 4-1, 4-2 y 4-3 se incluyen algunos ejemplos.

Algunos datos usados en visualización científica puede ser resultado de realizar abstracciones en el espacio físico, sin embargo, siguen teniendo una naturaleza inherentemente geométrica.

Existe información de carácter no físico (datos financieros⁴⁻⁴, información de negocios, colecciones de documentos y concepciones abstractas en general), cuya interpretación puede facilitarse al presentarla en forma visual. El problema es que esta información no tiene un mapeo espacial obvio. El buscar esquemas de visualización que faciliten la interpretación de estos tipos de información es un tema de actualidad. Es en estos escenarios que nace la visualización de la información (ejemplos en las figuras 4-6 y 4-5).

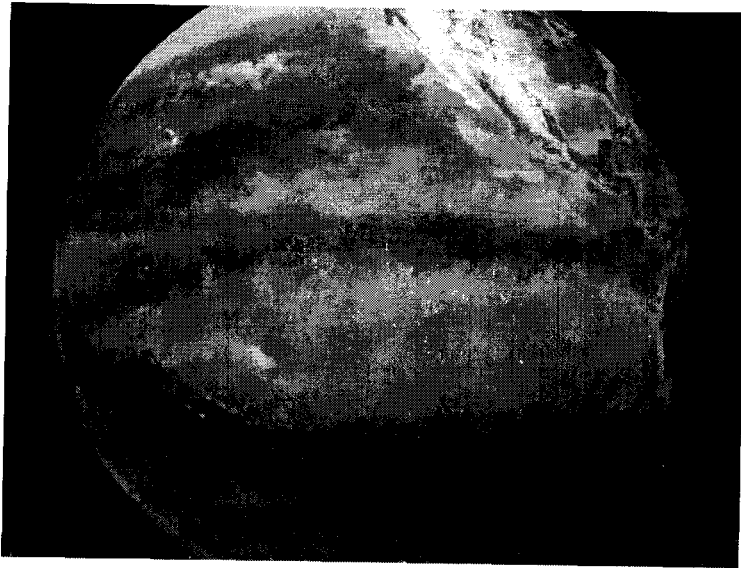


Figura 4-2: El fenómeno de "El Niño" (NASA).

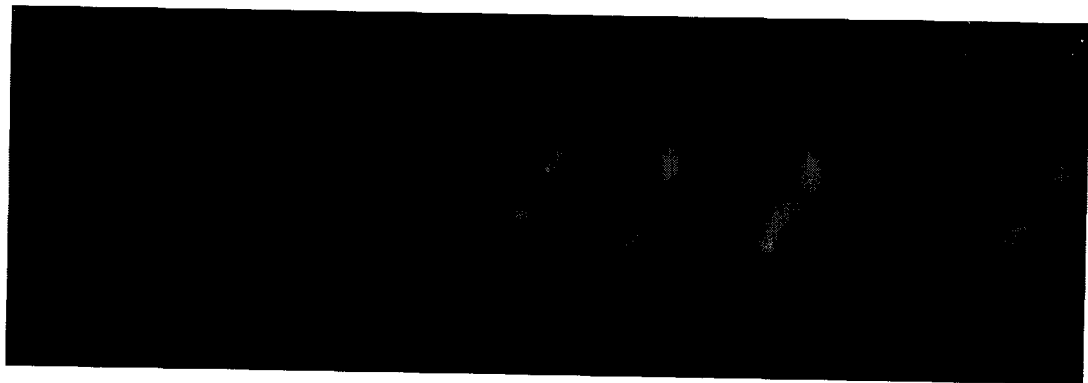


Figura 4-3: Flujo turbulento (SDSC).

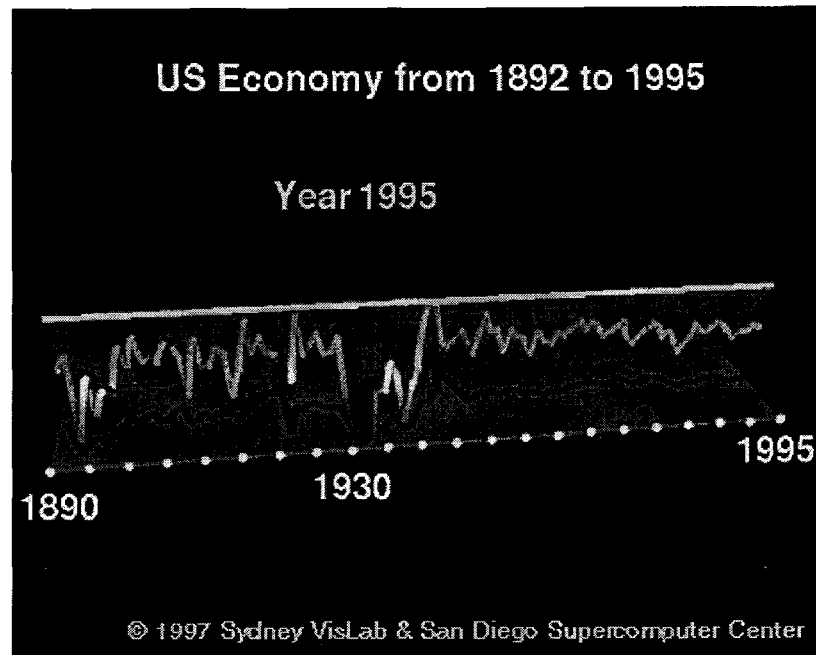


Figura 4-4: Evolución de la economía de los Estados Unidos (SDSC).

Nuevamente utilizaremos una definición propuesta por Card:

Visualización de la Información. Es el uso de representaciones computacionales, visuales e interactivas sobre datos abstractos para ampliar la cognición.

4.2.2 Orígenes y evolución de la Visualización de Información

La utilización de representaciones gráficas de datos se puede rastrear hasta finales del siglo XVIII, con los trabajos de Playfair. En 1967 Bertin publicó su teoría de gráficos[Bertin67]. Esta teoría identificó los elementos básicos de los diagramas y describió un marco de trabajo para su diseño. Tufte publicó una teoría de datos gráficos que enfatizaba la maximización de la densidad información útil[Tufte98]. Junto con Bertin, Tufte es reconocido como el padre de la visualización de información. A mediados de los 70s también se llevaban a cabo esfuerzos en el ámbito estadístico, tal es el caso de Turkey y su trabajo *Exploratory Data Analysis*, que se centraba en la extracción de información estadística cualitativa de representaciones gráficas. En la siguiente década, trabajos como los de Cleveland y McGill impulsaron la utilización de herramientas de visualización en el área estadística.

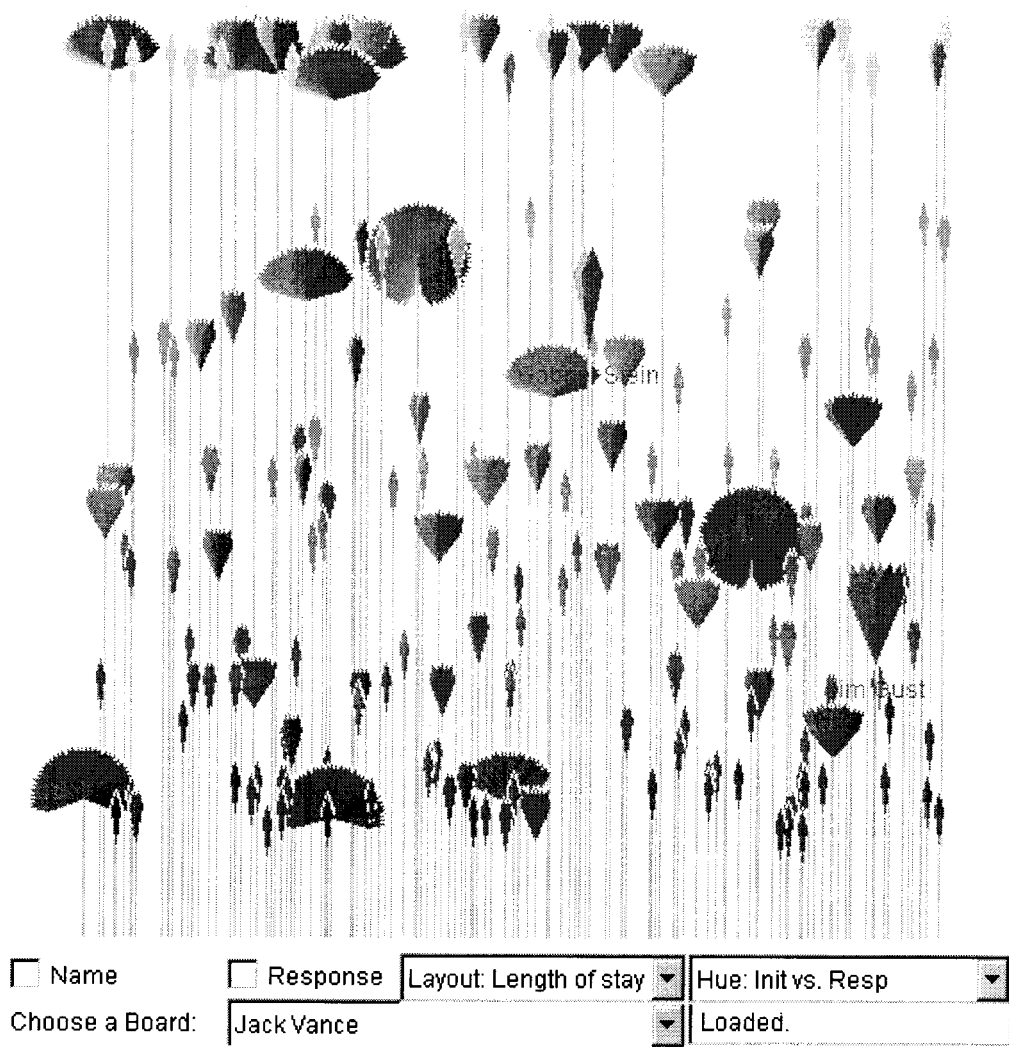


Figura 4-5: Visualización de conversaciones basadas en WWW (MIT Media Lab)

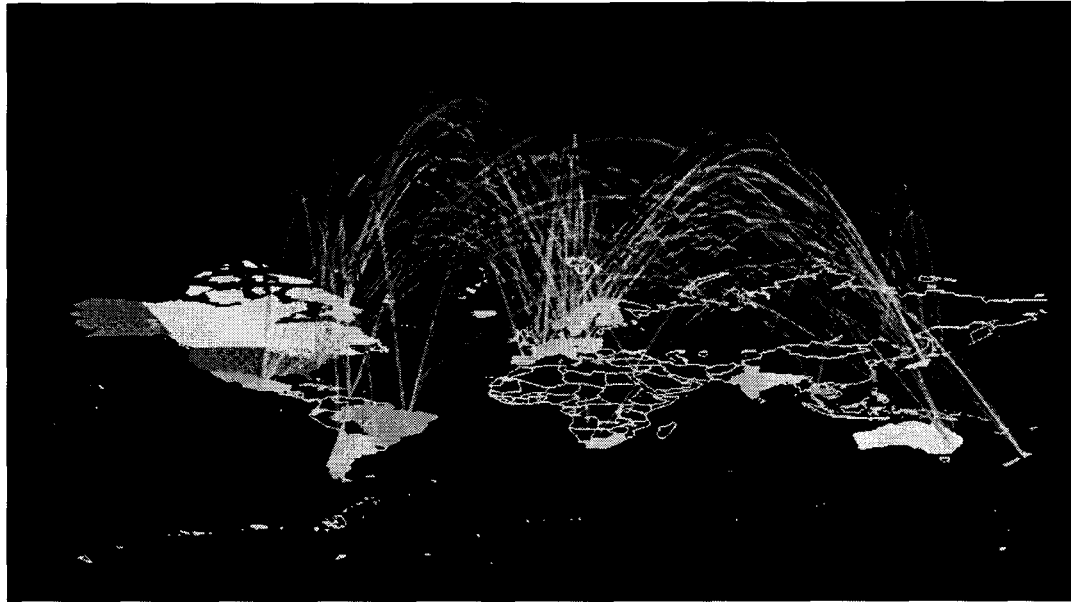


Figura 4-6: Flujo de tráfico en Internet (Laboratorios Bell)

Se puede decir, que la visualización científica nace como campo organizado en 1985 con la publicación de los trabajos de McCormik y DeFanti[McCormick87]. La primera conferencia de visualización de la IEEE fue llevada a cabo en 1990. En estos años una corriente impulsora al desarrollo de la visualización científica fue la necesidad de interpretar los enormes volúmenes de datos que enviaban los satélites de regreso a la Tierra. También ha existido un interés en la visualización por parte de la comunidad de inteligencia artificial, por ejemplo para el desarrollo de interpretaciones visuales automáticas[Mackinlay86].

Por otro lado, la comunidad de interfases de usuario vio con interés los avances en el Hardware gráfico para el desarrollo de nuevas interfases que permitiera a los usuarios interactuar de una manera sencilla e intuitiva con grandes volúmenes de datos. La primera referencia al término visualización de información se atribuye a Robertson[Robertson89].

4.3 Amplificación Cognitiva

La utilización de representaciones gráficas (tales como diagramas) de problemas reducen el tiempo y el esfuerzo para llegar a una solución. Esto es debido a que la utilización de diagramas

permite la agrupación de elementos, lo que reduce los espacios de búsqueda. También permiten asignar directamente a los objetos sus propiedades, reduciendo la necesidad de apareamientos simbólicos. Finalmente, los diagramas permiten identificar características del problema, que de otra forma sería difícil percibir.

Entonces, ¿en que formas mejora nuestra capacidad cognitiva el uso de representaciones gráficas?

- Incrementan los recursos disponibles de la persona (la memoria es el ejemplo más directo).
- Reducen la búsqueda de información.
- Permiten a la persona identificar patrones.
- Habilitan operaciones de inferencia a través de la percepción.
- Permiten el monitoreo de diversos eventos.
- Permiten la codificación de la información de una manera manejable.

4.4 Mapeo de Datos a una Representación Visual

4.4.1 Características Deseables de una Representación Visual

No todas las representaciones visuales tendrán la misma utilidad. La calidad de la representación gráfica, no se determina solamente por la selección de la información a visualizarse (no todos los datos son buena fuente para una visualización), también se ve afectada por la selección del paradigma que se vaya a utilizar. Si se toman las decisiones adecuadas, la utilidad de la herramienta de visualización se potenciará, y se dirá que la representación visual es buena.

Con una buena representación gráfica es más factible producir resultados creíbles, confiables y precisos[Tufte98a]. El objetivo principal es transmitir la mayor cantidad de información en la forma más directa posible, en otras palabras, el usuario debe de poder extraer la información desde la representación visual de la manera más natural posible. Una buena representación gráfica[Tufte98] debe de:

- Mostrar los datos.
- Inducir al observador a pensar en la naturaleza de los datos y no en los aspectos técnicos de la producción de la representación.
- Evitar la distorsión de los datos.
- Hacer coherentes conjuntos grandes de datos.
- Inducir a que el ojo humano compare diferentes piezas de datos.
- Revelar los datos a diferentes niveles de detalle.

Otros aspectos que se deben considerar para generar una buena representación gráfica son el adecuado manejo de los atributos (dimensiones) que se van a representar, esto puede ser alcanzado con una adecuada de codificación de colores y posiciones[Tufte99].

Ahora que ya se conocen los fundamentos y aplicaciones de la visualización, así como los aspectos deseables de una representación gráfica, surge la pregunta, ¿cómo se llevan los datos originales a una representación visual?. La figura 4-7 muestra un esquema de referencia de visualización. En las secciones siguientes se profundizará en cada uno de los módulos de este esquema de generación de representaciones visuales.

4.4.2 Tablas de Datos

Los datos originales pueden estar almacenados en distintos tipos de archivos y estructurados de distintas formas. La estrategia usual es transformar estos datos en una relación o conjunto de relaciones y almacenarlas en tablas. La estructura de las tablas creadas puede ser similar a la mostrada en la figura 4-8.

Tipos de Datos

Existen tres tipos básicos de variables:

- *Nominales*. Solo pueden ser iguales o diferentes a otros valores.

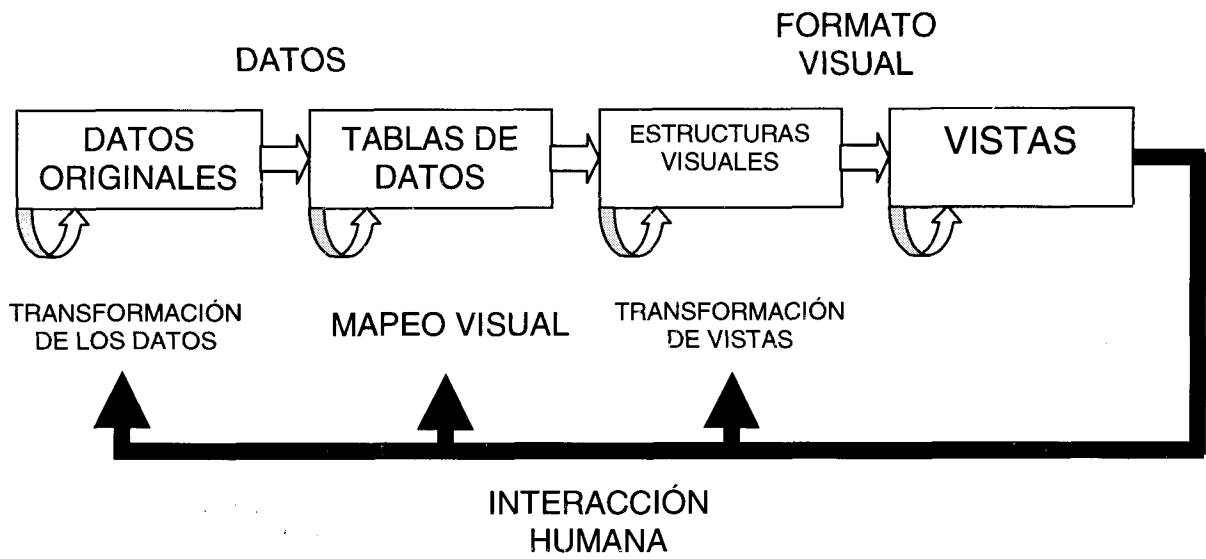


Figura 4-7: Esquema general de un sistema de visualización

	Caso 1	Caso 2	...	Caso n
Variable 1				
Variable 2				
...				
Variable n				

Figura 4-8: Tabla de datos

- *Ordinales*. Obedecen a relaciones de orden del tipo $<$, $>$, etc.
- *Cuantitativas*. Se pueden realizar operaciones aritméticas con ellas.

Las variables cuantitativas pueden ser puestas en correspondencia con variables ordinales mediante la división de sus posibles valores en rangos. Por ejemplo, se puede considerar que una persona es baja, media o alta de acuerdo a su estatura en centímetros: de 120-160 es bajo, de 160-170 medio, etc. Las variables nominales pueden ser ordenadas de acuerdo a algún criterio y establecer una relación de orden.

Metadatos

Los metadatos son información descriptiva acerca de los datos. Utilizando esta información se pueden crear representaciones con significado real. Los metadatos son necesarios, porque no toda la información referente a los datos (como valores permitidos) se puede inferir del tipo de variable.

Transformación de los Datos

En ocasiones, los datos originales no son apropiados para ser representados gráficamente. Para corregir estos problemas se pueden realizar transformaciones sobre los datos. Al hacerlo, usualmente se modifica la información contenida en el conjunto de datos (ganancias o pérdidas), por lo que se deben tener en cuenta las características de los datos que deseamos que queden sin alterar. Los tipos de transformaciones de datos son:

- Valores→Valores Derivados
- Estructura→Estructura Derivada
- Valores→Estructura Derivada
- Estructura→Valores Derivados

4.4.3 Estructuras Visuales

Una vez se han creado las tablas, y se han modificado los datos, es momento de crear la representación visual. Una premisa en esta etapa es la no alteración de la información contenida

en las tablas, en otras palabras, la representación visual debe transmitir toda y solamente la información contenida en las tablas.

Percepción

Una representación visual se considerará adecuada, solamente en los caso en los que promueva la percepción humana. Es por esto que es muy importante el estudiar la manera en la cual el ser humano percibe la información visual. Por ejemplo, es importante el manejo de parámetros tales como:

- Distancias entre elementos visuales.
- Tamaños de elementos visuales.
- Colores de cada elemento visual.
- Posiciones relativas de los elementos visuales.
- El grado de agrupamiento entre elementos visuales.
- Desvanecimiento de elementos visuales.

Sustrato Espacial

Así como existen límites a la percepción del ser humano (en términos de número de colores, por ejemplo), también hay limitaciones en el número de posibles representaciones que se pueden lograr a partir de los datos de una tabla. Los tipos de ejes que se pueden manejar en una representación son los que ya se presentaron en la sección de tipos de datos: Nominales, Ordinales y Cuantitativos. Así, una parte importante de la creación de una estructura visual es la asignación adecuada de los tipos de ejes. Según la colocación espacial de los ejes, se tienen las siguientes categorías.

- Composición. Colocación ortogonal de los ejes creando una métrica 2D.
- Alineamiento. Es la repetición de un eje en diferentes posiciones del espacio.
- Doblado. Es la continuación de un eje en una dimensión ortogonal.

	Espacial	Objeto
Extensión	Posición , tamaño	Escala de Gris
Diferenciación	Orientación	Color, textura, forma

Figura 4-9: Propiedades Retinales

- Recursión. Es la repetida subdivisión del espacio.
- Sobrecarga. Es la utilización del mismo espacio e más de una ocasión.

Marcas

Las marcas son simplemente indicadores visibles en el espació de la estructura visual. Existen cuatro tipos de marcas elementales: Puntos, Líneas, Áreas, Volúmenes.

Conexión y Cerradura

Las características de conectividad y cerradura transmiten mensajes importantes, por ejemplo en el caso de representaciones usando árboles y grafos, transmiten mensajes relativos a la estructura interna de los datos. En el caso de la cerradura, se transmite la idea de agrupamiento o contenedor.

Propiedades Retinales

Se deben considerar las propiedades de la retina humana para la creación de estructuras visuales. En general tenemos propiedades relacionadas con la extensión y la diferenciación de los objetos en el espacio. La figura 4-9 muestra esto.

Codificación Temporal

Se puede aprovechar la sensibilidad del ojo humano a los cambios en la posición de las marcas para codificar información temporal, es decir, se pueden utilizar animaciones para proveer la idea de cambio en el tiempo.

4.4.4 Transformaciones de Vistas

Las transformaciones visuales permiten la transformación de representaciones gráficas estáticas en lo que propiamente se entiende por visualización.

Localización de Sonda

Permite al usuario "navegar" en la estructura interna de una representación gráfica. Por ejemplo, se puede entrar a un sólido y verlo por dentro.

Control de Puntos de Vista

Es la aplicación sobre la representación gráfica, de transformaciones como la rotación traslación y escalamiento.

Distorsiones

Son transformaciones visuales que permiten magnificar o empequeñecer algunos elementos visuales.

4.4.5 Interacción Humana

Quizás la parte fundamental de una representación gráfica es la interacción con el usuario. Es la interacción la que le permite al usuario modificar la forma en la que se presenta la información y facilitar la extracción de información. Por ejemplo el usuario puede cambiar de posición algunos elementos de visualización, como en el caso de nodos y arcos de una representación basada en grafos.

4.5 Resumen del Capítulo

La mente de los seres humanos es receptiva a recibir ayudas externas que aumentan sus capacidades. Muchas de estas ayudas externas, tienen carácter visual. La visualización es una disciplina que busca explotar la habilidad natural del humano para interpretar información que le es presentada en forma visual. Es la nueva implementación de una vieja idea: utilizar medios

visuales para hacer evidente la información que contiene un conjunto de datos, utilizando las modernas herramientas computacionales. Si bien el origen de la visualización esta claramente identificado en la rama de la visualización científica, en los últimos años la necesidad de trabajar con información de carácter no física (abstracta) ha impulsado el desarrollo de la visualización de la información.

Capítulo 5

Técnicas de Visualización Aplicadas al Problema SAT

En este capítulo se muestran las técnicas de visualización de datos aplicables a instancias SAT que se han implementado como resultado de este proyecto. Las técnicas que a continuación se presentan pueden ser divididas en dos grupos:

1. Representaciones gráficas que ponen en evidencia relaciones o distribuciones de variables de instancias.
2. Representaciones gráficas de las propia instancias.

La primera categoría esta representada por las técnicas que se discuten en los apartados 5.2 a 5.6, en tanto que la sección 5.7 presenta una propuesta de representación visual de instancias SAT.

5.1 Preliminares

Para todas las herramientas desarrolladas en este proyecto se utilizaron como entrada instancias 3-SAT en su FNC. Antes de la aplicación de las cualquiera de las técnicas, se ordenaron las variables en las cláusulas en orden ascendente con respecto al número de variable. Buscando que todo el software resultante de esta tesis estuviera en concordancia con la mayoría de los esfuerzos en el área, se utilizó como formato de archivo de entrada el propuesto como estándar por el

reto DIMACS[DIMACS93]. La estructura del formato de archivo CNF permite la inclusión de comentarios en los que usualmente aparece el método de generación de la instancia, el nombre del autor y se indica si la instancia es satisfactible. Las cláusulas son representadas por líneas de enteros (número de variable). A continuación se presenta un ejemplo de archivo CNF.

```
c COMMENTS
c COMMENTS
c COMMENTS
p cnf 100 200
40 54 75 0
-40 54 58 0
.
.
.
```

Estructura de un archivo CNF

5.2 Distribución de Signos y Variables

Se puede extraer información útil de la distribución de las variables en la instancia. Por ejemplo, intuitivamente se esperaría que las instancias duras tengan una distribución uniforme de variables. Se puede decir lo mismo de la distribución de variables considerando el signo con el que aparecen en las cláusulas. Intuitivamente se espera que, en una instancia dura, el número de ocurrencias en las cláusulas de la literal x sea aproximadamente igual al número de ocurrencias de la literal $\neg x$.

Para visualizar la distribución de las variables se utilizó una gráfica de barras cuya altura está asociada con el número de ocurrencias de la variable. En la gráfica se utilizan dos colores para expresar la proporción de veces que cada variable aparece negada. Debido a que muchos de los generadores de casos de prueba siguen las indicaciones propuestas por Selman, cada variable tiende a aparecer aproximadamente el mismo número de veces que cualquier otra. La figura 5-1 es un ejemplo de lo anterior. Existen generadores que incluso restringen las instancias

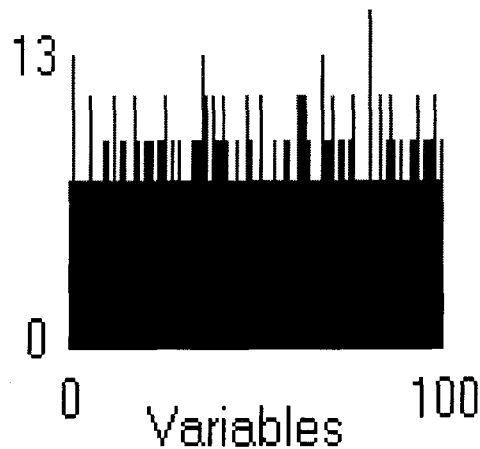


Figura 5-1: Distribución de variables

generadas a un conjunto en el que todas las variables aparecen con la misma frecuencia, lo que puede ser observado en la figura 5-2.

Los mismos patrones son observados con la distribución de signos. Las instancias que los generadores de casos de prueba duros arrojan, suelen tener aproximadamente el mismo número de ocurrencias de variables negadas que no negadas, lo que es evidente en las figuras 5-1 y 5-2.

5.3 Relaciones entre dos Variables en una Cláusula

Se dice que dos variables están relacionadas en una instancia, si existe al menos una cláusula en la que ambas variables aparecen juntas. El grado en que dos variables están relacionadas (número de cláusulas en las que aparecen juntas) es un factor que puede considerarse para mejorar el desempeño de un algoritmo de solución. Considérese la cláusula (3 5 6), en ella existen tres relaciones cuya ocurrencia puede ser almacenada en una matriz (figura 5-3) Para poder identificar visualmente el grado de relación entre variables, se diseñaron programas que graficaran de manera clara dichas relaciones.

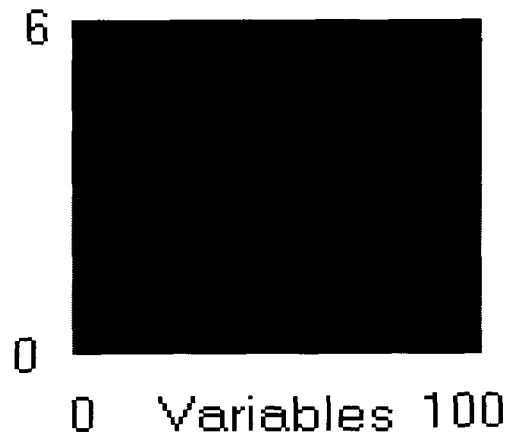


Figura 5-2: Distribución de variables

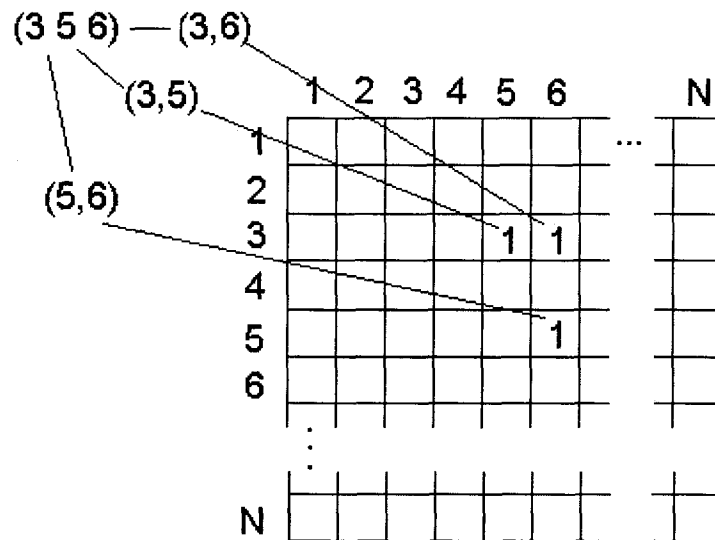


Figura 5-3: Matriz de relaciones

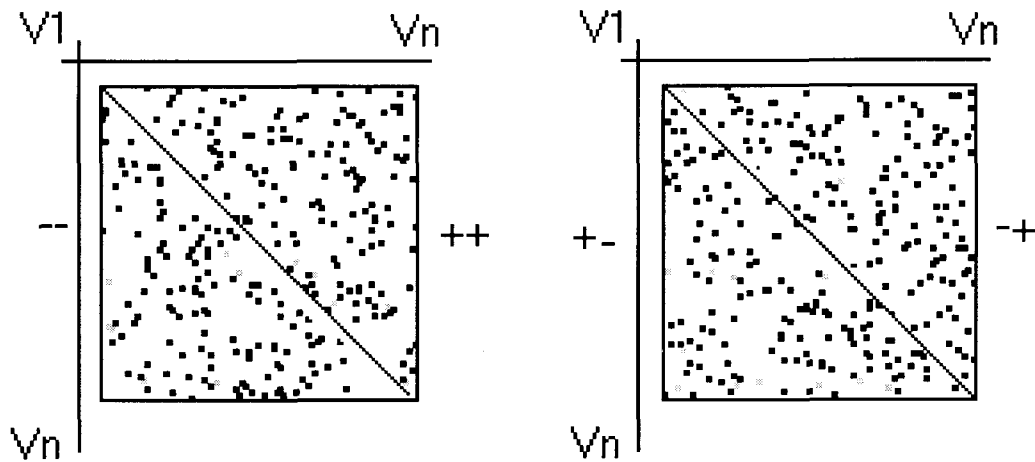


Figura 5-4: Representación de relaciones entre dos variables

5.3.1 Versión 2D

Se diseñó una representación gráfica que indicara la ocurrencia de relaciones entre dos variables y las combinaciones de signos con las que dichas relaciones ocurran. Originalmente se probó con una representación que dividía las relaciones en dos grupos en función de la combinación de signos de las literales en la cláusula (figura 5-4).

La diagonal superior de la primera matriz indica la ocurrencia de relaciones entre variables cuando ambas aparecen no negadas, la diagonal inferior indica las ocurrencias de relaciones cuando ambas variables aparecen negadas. Análogamente, la segunda matriz indica las relaciones referentes a las combinaciones negada/no negada y no negada/negada.

Para indicar el número de ocurrencias de una relación, se utilizó un código de colores .

Buscando tener toda la información de las relaciones en una sola representación, se diseñó la configuración de rombo que se muestra en la figura 5-5.

El código de colores se maneja de la misma manera que en la versión de dos matrices. El único cambio notable en esta representación es la modificación de la dirección de los ejes horizontales (figura 5-6), esto fue realizado para mantener todos los puntos dentro de la gráfica.

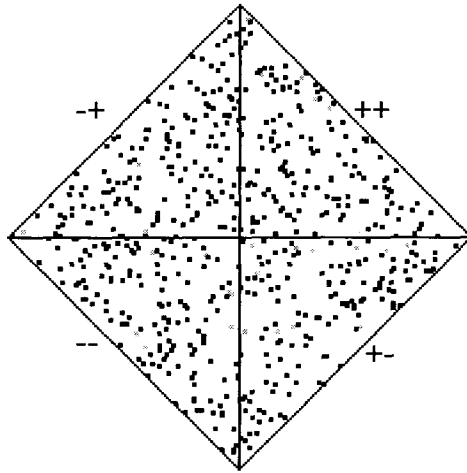


Figura 5-5: Relación entre dos variables utilizando la configuración de rombo

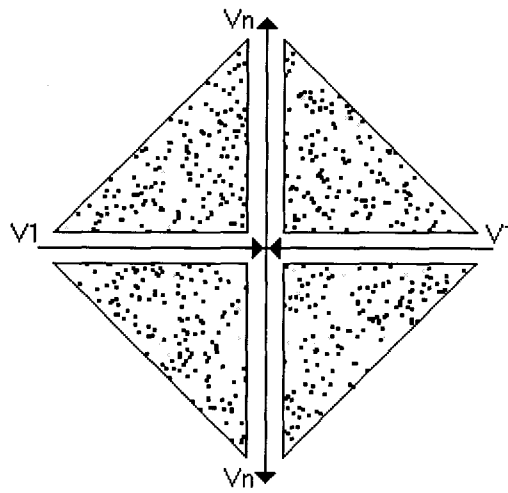


Figura 5-6: Direcciones de los ejes en la representación de rombo

5.3.2 Versión 3D

En algunos casos (por ejemplo cuando el número de cláusulas es mucho mayor al número de variables) la aplicación de la gráfica de rombo puede ser impráctica.

Para añadir la posibilidad de interactuar con la forma en que los datos son visualizados se diseñó una representación interactiva en 3D. Esta técnica es básicamente una versión en 3D de la representación de rombo discutida en la sección anterior. La dirección de los ejes y el código de colores son manejados de la misma manera que la representación de rombo, pero en lugar de puntos para identificar la frecuencia de las relaciones entre variables se utilizan líneas. La altura de estas líneas está relacionada con el número de ocurrencias de la relación. A diferencia de lo que ocurre con la versión 2D de la gráfica, la perspectiva de la versión en 3D puede ser reorientada por el usuario. El usuario puede escalar, rotar y trasladar la gráfica que el programa presenta. Con esto se busca facilitar la exploración de zonas de interés.

Las figuras 5-7 y 5-8 muestran una comparación de las relaciones de una misma instancia, visualizada utilizando las versiones 2D y 3D.

5.4 Relaciones entre tres Variables

Los métodos de visualización que a continuación se presentan, complementan la información que se obtiene aplicando las técnicas descritas en la sección 5.3. En este caso, se utiliza una representación en tres dimensiones para indicar las relaciones entre tres variables en una instancia 3-SAT.

En la figura 5-9, se muestran los ejes correspondientes a la primera, segunda y tercera variable en la cláusula. De igual forma que en los casos anteriores, se utiliza una codificación de colores para indicar la ocurrencia de relaciones. Al utilizar las tres variables de una instancia 3-SAT, los puntos que se grafican indican cuantas veces una combinación de variables (sin considerar los signos) se repite.

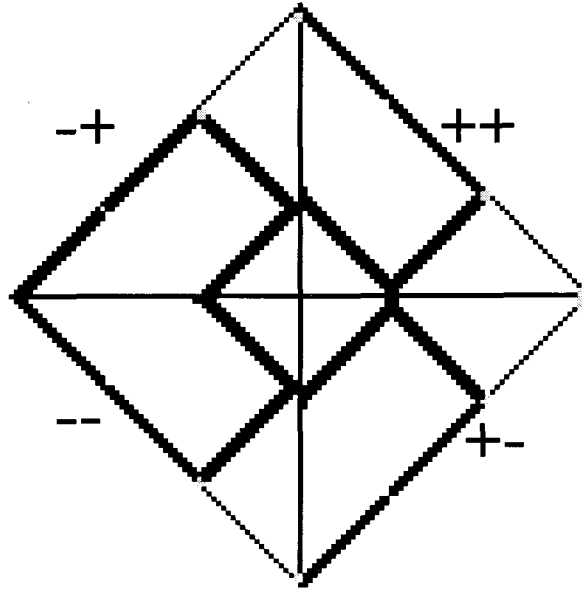


Figura 5-7: Relaciones entre dos variables utilizando la representación 2D

Al igual que con los casos anteriores, la perspectiva de la representación visual puede modificarse utilizando el ratón. El usuario puede rotar, escalar y trasladar la gráfica, lo que facilita la exploración de regiones de interés.

La figura 5-10 muestra un ejemplo de relaciones de variables en una instancia 3-SAT visualizadas utilizando esta técnica y se observa la imagen desde diferentes puntos de vista.

5.5 Relaciones entre tres Variables Considerando Signos

La herramienta presentada en la sección anterior contabiliza la ocurrencia de las variables sin tomar en consideración su signo. Para poder incluir la información del signo en la visualización, se deben delimitar regiones en el espacio y asignarlas a cierta combinación de signos. Dado que existen ocho combinaciones de signos posibles en una cláusula 3-SAT, se necesita de ocho regiones (octantes) para acomodar los puntos correspondientes en la representación visual. La figura 5-11 muestra los ejes coordenados utilizados y se indican las regiones ocupadas por dos

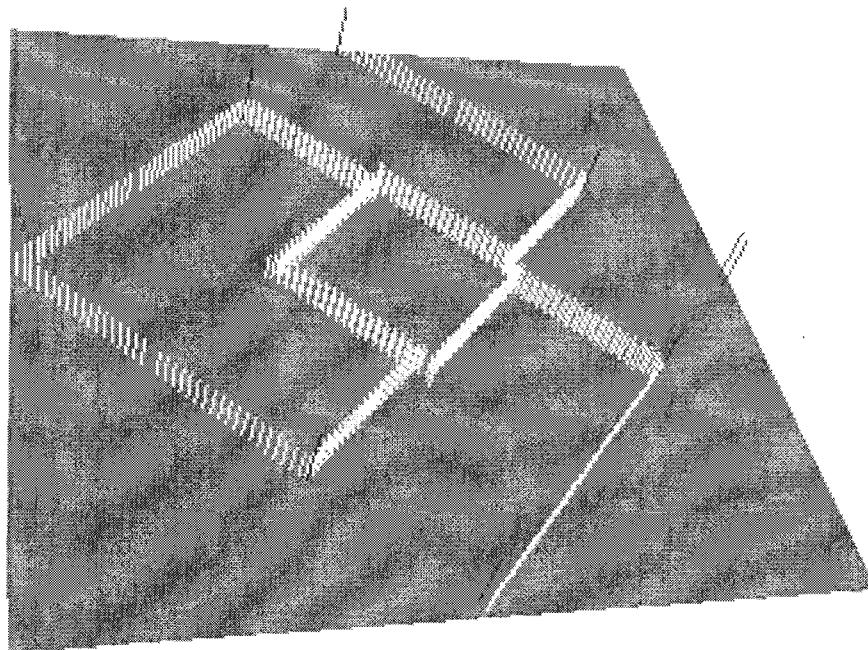
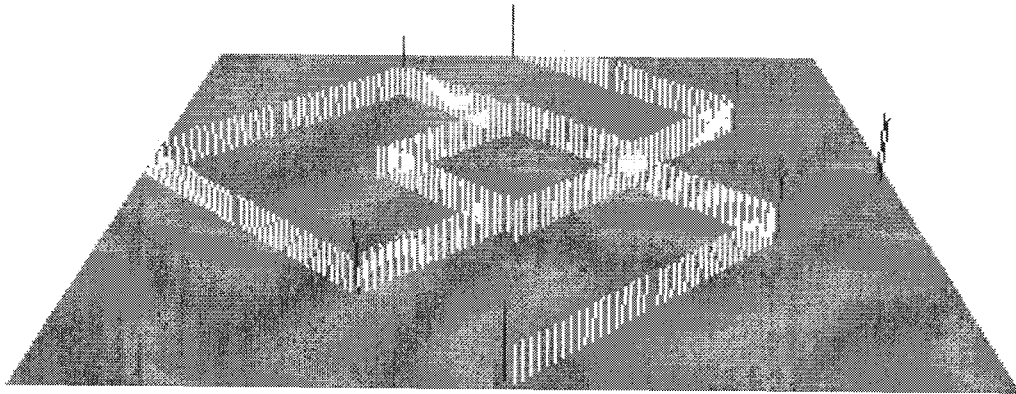


Figura 5-8: Relaciones entre dos variables utilizando la versión 3D

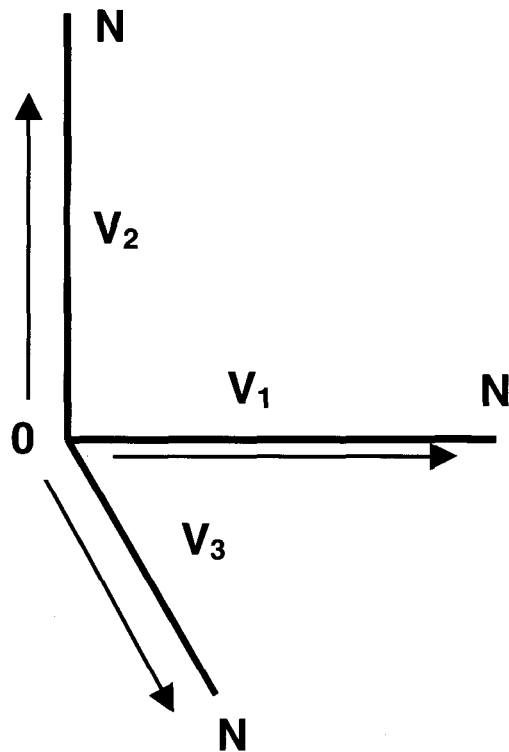


Figura 5-9: Ejes correspondientes a la primera, segunda y tercera variable de una instancia 3-SAT

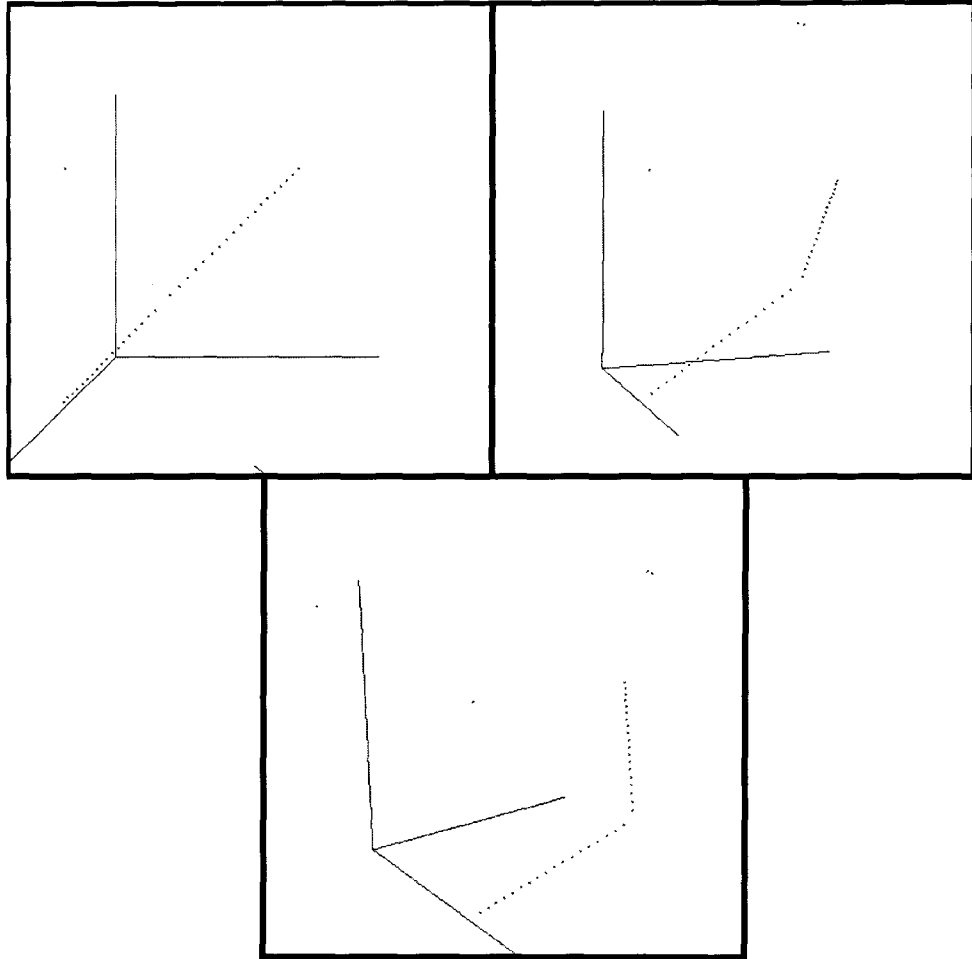


Figura 5-10: Relaciones de tres variables en una instancia, sin considerar los signos

de las posibles combinaciones.

Al igual que en los casos anteriores, la representación gráfica puede ser manipulada por el usuario para buscar un punto de vista que facilite la localización de patrones.

5.6 Distribución de Combinaciones de Signos

El último método de visualización desarrollado se utiliza para evidenciar la distribución de las combinaciones de signos en una cláusula, sin tomar en cuenta las variables a las que afectan. A diferencia de las muchas de las técnicas anteriores, esta última solo puede ser aplicada en problemas K-SAT.

En la figura 5-13 se muestra un ejemplo correspondiente a una instancia 3-SAT. Para visualizar todas las combinaciones de signos, se utiliza un octágono dividido en triángulos. El color de cada fracción del octágono es una estimación del número de ocurrencias de las ocho posibles combinaciones de signos. Los triángulos se colorean de acuerdo al número de ocurrencias de una combinación de signos asociada. Si alguna combinación no aparece en la instancia, el triángulo correspondiente se colorea en negro.

Con esta técnica se puede obtener información como el número de cláusulas de Horn en la instancia. Es importante notar esto, ya que el problema Horn SAT puede ser resuelto en tiempo polinomial. La gráfica también muestra si una instancia 3-SAT tiene una de dos soluciones triviales, cuando todas las cláusulas tiene al menos una literal no negada y cuando todas las cláusulas tienen al menos una literal negada.

5.7 Representación Gráfica de Instancias 3-SAT

Las herramientas de visualización presentadas en las secciones anteriores ponen en evidencia las distribuciones de signos y variables, así como las relaciones entre variables. Sin embargo, con excepción de la herramienta discutida en la sección 5.5, no es posible reconstruir a partir de la información visual la instancia que le dio origen. Para obtener una representación visual,

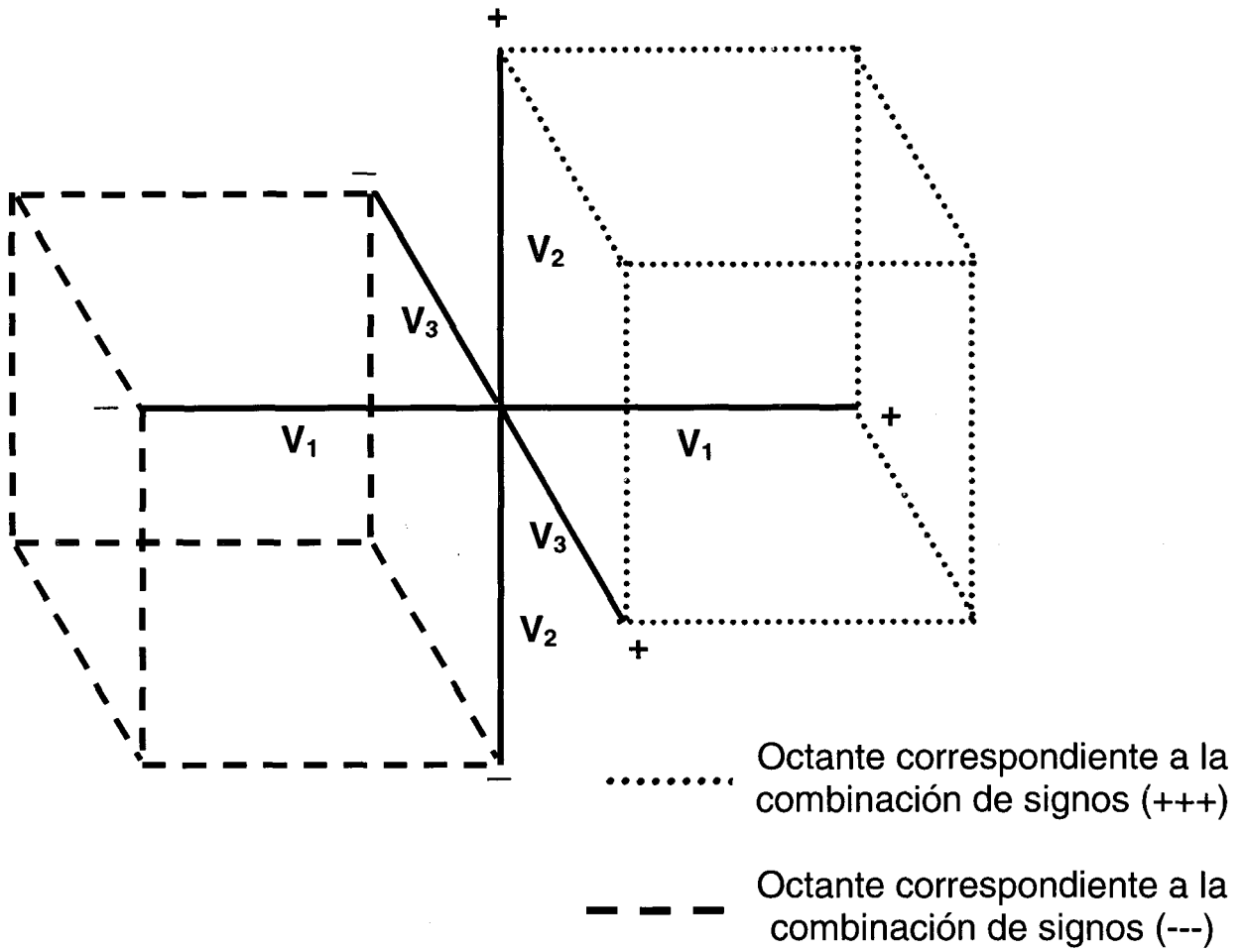


Figura 5-11: Regiones (octantes) correspondientes dos combinaciones de signos

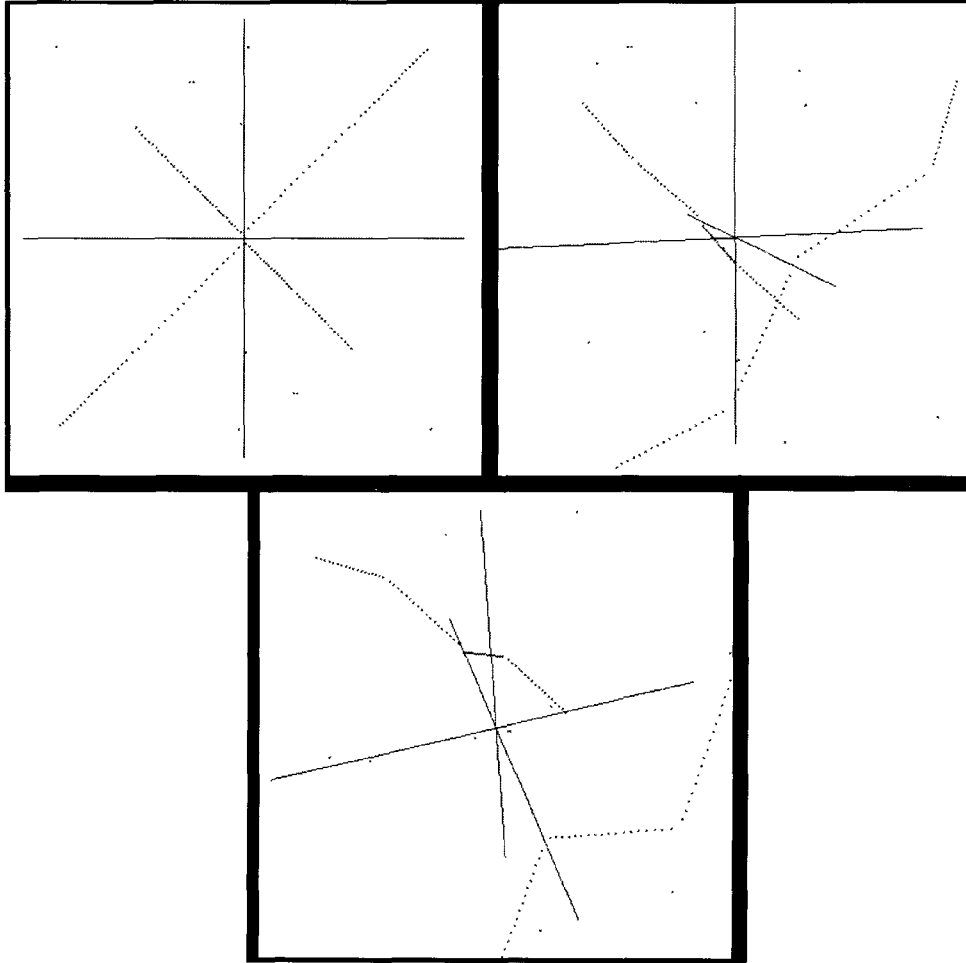


Figura 5-12: Relaciones entre tres variables considerando el siglo

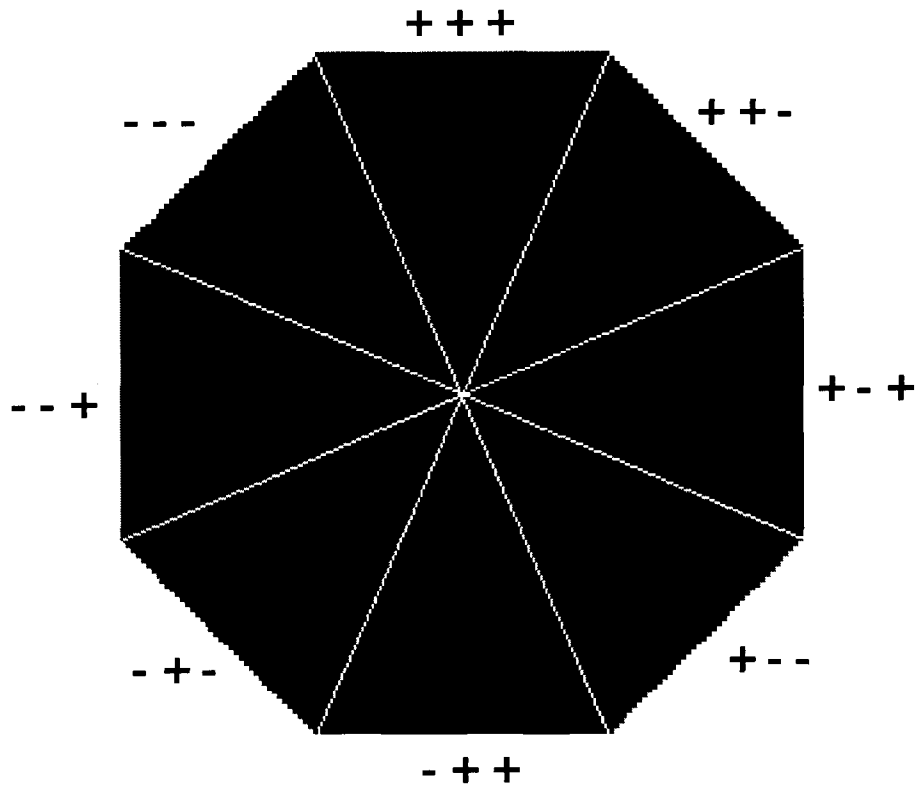


Figura 5-13: Distribución de signos en una instancia 3-SAT

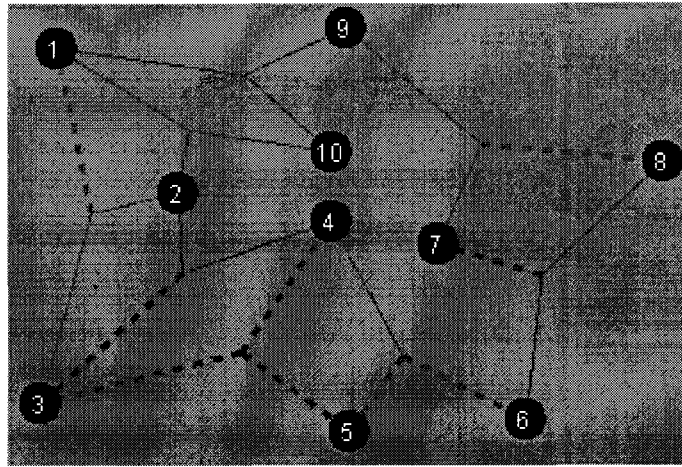


Figura 5-14: Representación gráfica de una instancia SAT

y no ambigua, de una instancia SAT, se exploraron modelos, como el de hipergrafos propuesto por Gallo[Gallo93].

Se partió de la propuesta original de Gallo, que se presentó en la sección 3.2.1 y se modificó para buscar facilitar la interpretación. El modelo de representación de instancias resultantes se muestra en la figura 5-14. La figura representa la instancia

$$(\neg 1 \vee 2 \vee 3) \wedge (4 \vee \neg 5 \vee \neg 6) \wedge (7 \vee \neg 8 \vee 9) \wedge (10 \vee 1 \vee 2) \wedge (\neg 3 \vee \neg 4 \vee \neg 5) \wedge (6 \vee \neg 7 \vee 8) \wedge (9 \vee 10 \vee 1) \wedge (2 \vee \neg 3 \vee 4)$$

Cada vértice del hipergrafo representa una variable de la instancia. Cada hiperarco representa una cláusula. Para indicar si la literal aparece en la cláusula negada o no negada, se utilizan diferentes tipos de líneas. Si la línea que conecta la variable con el resto del hiperarco es punteada, se entenderá que la variable aparece negada en dicha cláusula. Cada hiperarco tiene un color distintivo, lo que facilita identificar las cláusulas conforme aumenta el tamaño de las instancias que se representan. Además, el grosor de las líneas que representan segmentos de hiperarco asociados con una variable negada es mayor. Como última medida para poder identificar diferentes cláusulas sobre el mismo conjunto de variables (sección 3.2.2), los hiperarcos son semitransparentes.

En lo que se refiere a la posibilidad de interacción con el usuario, la herramienta permite

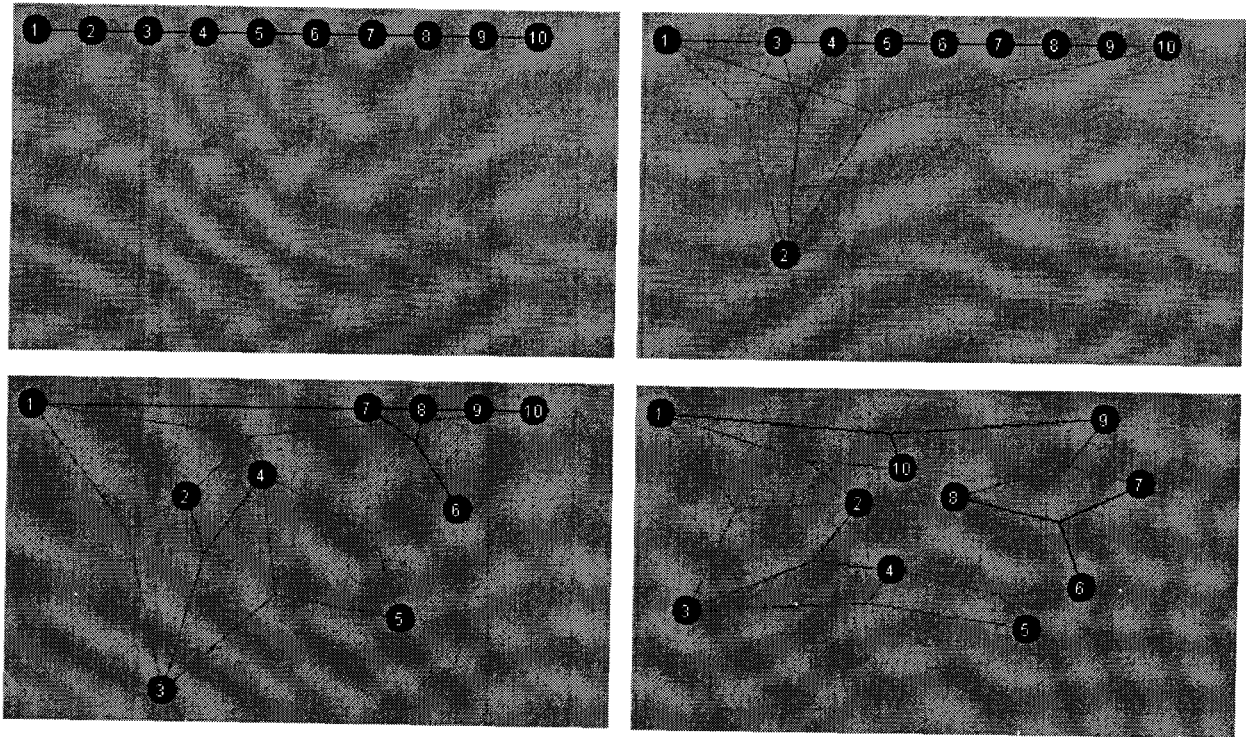


Figura 5-15: Relocalización de los vértices del modelo de hipergrafo modificado.

la relocalización de los vértices del grafo utilizando el ratón. El usuario solo debe colocar el cursor sobre uno de los vértices y "arrastrarlo" a su nueva posición, al hacerlo puede mejorar la calidad de la información que se le presenta (por ejemplo, puede intentar colocar los vértices de manera que se minimice el número de cruces entre hiperarcos). Los hiperarcos se reorganizan automáticamente. En la figura 5-15 se muestran diferentes momentos de la manipulación de una visualización de instancia.

5.8 Resumen del Capítulo

Las herramientas de visualización que se desarrollaron en este proyecto pueden dividirse en dos grupos: a) Técnicas que hacen evidentes las distribuciones y relaciones de variables y signos, y b) Representaciones gráficas de instancias SAT. Se desarrollaron cinco técnicas que pertenecen al primer grupo y una que pertenece al segundo. Se implementaron técnicas que permitieran visualizar la distribución de signos y variables en la instancia, así como la relaciones entre dos y tres de variables. Se utilizó una variación del modelo de hipergrafos propuesto por Gallo para

representar las instancias SAT.

Capítulo 6

Ejemplos de Aplicación y Resultados

El objetivo de este trabajo de investigación no es el descubrimiento de patrones asociados con la dureza de instancias, sino el diseño e implementación de herramientas que permitan hacerlos evidentes. Sin embargo, durante el desarrollo del proyecto se han identificado aplicaciones específicas para las técnicas de visualización, mismas que se comentan en las secciones siguientes.

En primer lugar se tratará la aplicación de estas herramientas visuales para identificar algunas características de las instancias resultado de generadores de casos de prueba. En segundo lugar se discutirá la posibilidad de extraer de las representaciones visuales, información que ayude en la selección de algoritmos de solución.

6.1 Identificación de Peculiaridades en los Generadores de Casos de Prueba

El desarrollador de algoritmos de solución puede notar que el desempeño de su algoritmo de solución varía cuando se utilizan instancias procedentes de diferentes generadores de casos de prueba. Es por esto que le reportaría utilidad el identificar características de las instancias que cada generador produce.

En el otro extremo del problema, durante el desarrollo de un nuevo generador de casos de prueba, es útil el tener una herramienta que permita analizar el tipo de instancias que se crean y, de esta manera, verificar el correcto funcionamiento del algoritmo de generación. Por

ejemplo, se puede desear tener una estimación de la distribución de signos y variables, para asegurarse que correspondan a lo esperado por el diseñador. Un conjunto de herramientas de visualización facilitan la identificación de estas peculiaridades.

6.2 Herramientas de Visualización como Auxiliar en la Selección de Algoritmos de Solución

El uso de las herramientas desarrolladas puede facilitar el proceso de selección de los algoritmos de solución. Lo anterior es posible ya que existen características de las instancias que afectan el desempeño de algoritmos específicos, en otras palabras, algunos métodos funcionarán mejor que otros en instancias específicas.

Para ejemplificar esto, se comentará como es posible utilizar las herramientas de visualización para determinar si una instancia es adecuada para ser atacada mediante Algoritmos Genéticos (AG).

6.2.1 Algoritmos Genéticos y SAT

Los Algoritmos Genéticos (AG), son unos de los métodos incompletos más utilizados en la solución de problemas NP-Complejos. Esto se debe al éxito que los AG tienen cuando se les aplica a problemas duros y con un espacio de solución grande.

La codificación es un elemento extra que hace a los AG especialmente adecuados para resolver SAT. Debido a que el problema SAT consiste de una búsqueda sobre n variables con solo dos posibles valores, una representación muy natural sería tener una cadena binaria de longitud n , donde el valor de Verdadero se representa con 1 y el de Falso con 0 y el i -ésimo bit de la cadena es el valor de verdad de la i -ésima variable de la fórmula proposicional en cuestión (figura 6-1).

6.2.2 Instancias SAT adecuadas para AG

El Teorema de Esquemas [Holland92] enuncia los prerequisites para la aplicación exitosa de AG. De acuerdo con Rodríguez [Rodríguez99], en una instancia SAT, estas condiciones se presentan

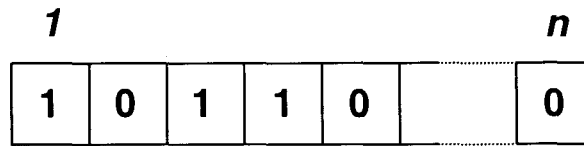


Figura 6-1: Cadena binaria asociada a valores de verdad de variables proposicionales

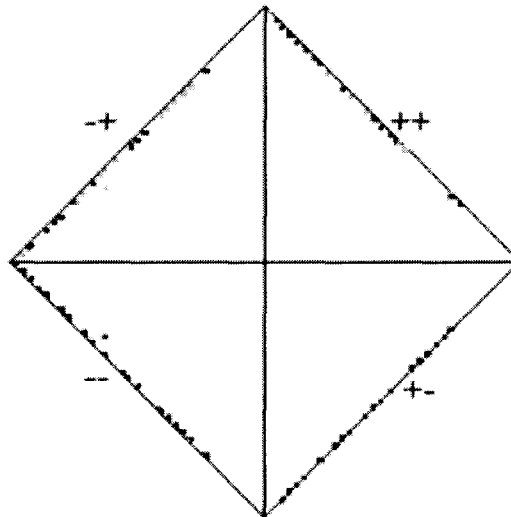


Figura 6-2: Variables numéricamente cercanas

cuando las variables relacionadas están numéricamente cerca y por ello, los bits relacionados en la cadena binaria se encontrarán cerca.

Identificación visual de instancias SAT adecuadas para la aplicación de AG

¿Cómo se "ve" una instancia SAT adecuada para ser resuelta con AG?. Como se ha mencionado, si las variables relacionadas se encuentran numéricamente, cerca el problema puede resolverse eficientemente aplicando AGs. Esta condición se puede detectar aplicando las técnicas discutidas en las secciones 5.3, 5.4 y 5.5.

Si los puntos se agrupan en las diagonales exteriores cuando se aplican las técnicas descritas en la sección 5.3, estamos frente a una instancia susceptible de ser resuelta utilizando AGs. Un ejemplo de lo anterior puede ser observado en la figura 6-2 en la figura 6-3.

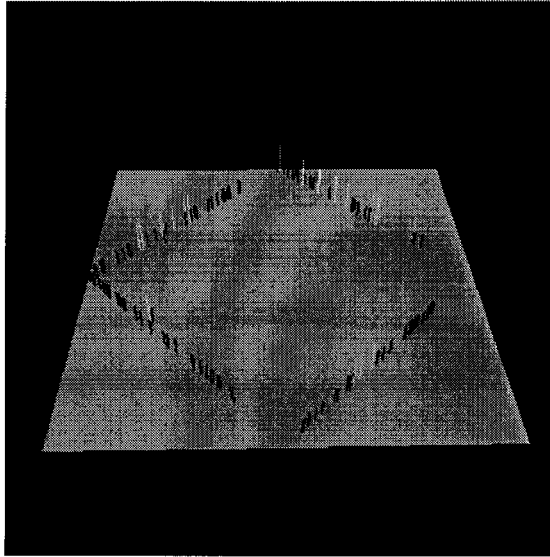


Figura 6-3: Variables numéricamente cercanas

Aplicando las herramienta descrita en la sección 5.4 y 5.5, se identifica un alto grado de relación entre variable numéricamente cercanas, si se observa el patrón presentado en las figuras 6-4 y 6-5.

6.3 Resumen del Capítulo

La identificación de patrones asociados con la dureza de instancias está más allá de los alcances de esta tesis. Sin embargo, ha sido posible identificar aplicaciones específicas para las herramientas de visualización propuestas. Como ejemplos de aplicación se presentaron la utilización de la técnicas desarrolladas, en la etapa de caracterización de las instancias resultado de generadores de casos de prueba, y la identificación de instancias susceptibles a ser resueltas eficientemente por un método específico de solución (Algoritmos Genéticos).

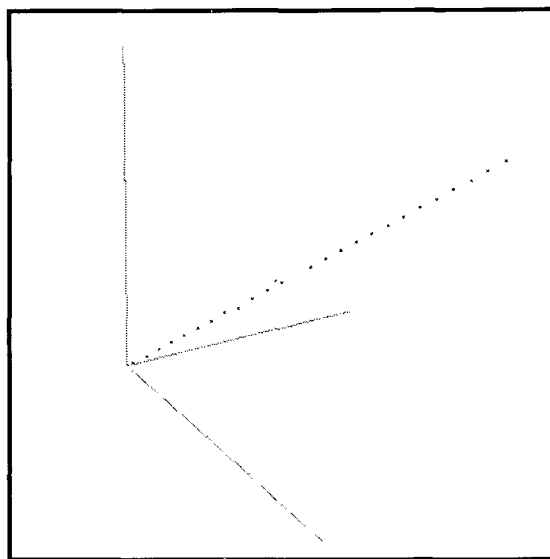


Figura 6-4: Variables numéricamente cercanas (versión 3D)

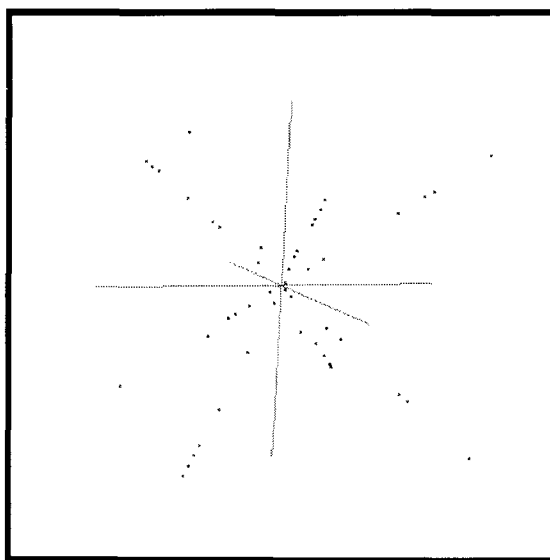


Figura 6-5: Variables numéricamente cercanas (versión 3D con signos)

Capítulo 7

Trabajos Futuros y Conclusiones

7.1 Ampliaciones y posibles modificaciones

La colección de herramientas de visualización que se han presentado, son apenas un discreto subconjunto del total de posibles representaciones visuales aplicables a este problema. Es por esto que un área importante de posibles ampliaciones, es el desarrollo de nuevas herramientas de visualización apropiadas. Lo anterior no solo es aplicable a técnicas de visualización que hagan evidente información nueva, sino que se pueden diseñar mejores representaciones para transmitir la información que ya manejan las técnicas resultado de este proyecto.

7.2 Conclusiones

La utilidad de este trabajo se centra en dos aspectos:

- El uso de la representación visual de instancias y sus relaciones entre variables como una herramienta en la identificación de instancias 3-SAT duras.
- Un medio para identificar si un método específico de solución es adecuado para un conjunto de instancias SAT específicas.

La aplicación de las técnicas de visualización de datos en la identificación de instancias duras parece ser una opción viable y un buen complemento a la relación M/N .

La utilización de las herramientas de visualización también fueron utiles como auxiliar en la selección de un método de solución adecuado a las instancias específicas.

Mucho trabajo debe ser llevado a cabo para identificar patrones visuales asociados con la dureza de instancias SAT.

Bibliografía

- [Bertín67] Bertín, J. *Semiology of Graphics: Diagrams, Networks, Maps*. *University of Wisconsin Press* (1967).
- [Card99] Card, S., Mackinlay, J. y Shneiderman, B. *Readings in Information Visualization*. *Morgan Kaufmann* (1999)
- [Chabrier96] Chabrier, J., Richer, J. M. and Chabrier, J. J. Resolution of structured SAT problems with Score(FD/B). *Université of Bourgogne* (1996).
- [Chang87] Chang, C. L. *Syboolic Logic and Mechanical Theorem Proving*. *McGraw Hill* (1987)
- [Cook71] Cook, S.A The complexity of theorem-proving procedures. *Proceedings of the 3rd Annual ACM Symposium on the Theory of Computing* (1971).
- [Cook97] Cook, S.A. and Mitchell, D.G. Finding Hard Instances of the Satisfiability Problem: A Survey. *DIMACS Series in Discrete Mathematics and Theoretical Computer Sciences* (1997).
- [Davis60] Davis, M. y Putnam, H. A computing procedure for quantification teory. *Journal of ACM* (1960).
- [Devadas89] Devadas, S. Optimal layout via boolean satisfiability. *Proceedings of ICCAD'89* (1989).
- [DIMACS93] Satisfiability Suggested Format, DIMACS (1993).
- [DOWLING84] Dowling, W. and Gallier, J. Linear-time algorithms for testing the satisfiability of propositional Horn formulae. *Journal of Logic Programming* (1984).

- [Gallo93] Gallo, G., Longo, G., Nguyen, S. y Pallottino, S. Directed hypergraphs and applications. *Discrete Appl. Math* (1993).
- [Gallo92] Gallo, G. y Pretolani, D. A new algorithm for the propositional satisfiability problem. *Discrete Applied Mathematics* (1992).
- [Garey79] M.R. and Johnson, S. Computers and Intractability: A Guide to the Theory of NP-Completeness. *W. H. Freeman, San Francisco*, Garey (1979).
- [Goldberg79] Goldberg, A. On the complexity of the satisfiability problem. *Courant Computer Science Report, No.16, New York University* (1979).
- [Horic97] Horic, S. and Watanabe, O. Hard Instance Generation for SAT. *Technical Report Dept. of Computer Science, Tokyo Institute of Technology* (1997).
- [Holland92] Holland, J. Adaptation in Natural and Artificial Systems. *MIT Press* (1992).
- [McCormick87] McCormick, B. y DeFanti, T. Visualization is Scientific Computing. *Computer Graphics, 21* (1987).
- [Mackinlay86] Mackinlay, J. Automatic Design of Graphical Presentations. *Ph.D. Thesis, Stanford University* (1986).
- [Robertson89] Robertson, G., Card, S. y Mackinlay, J. The Cognitive Co-processor for Interactive User Interfaces. *Proceedings of UIST'93, ACM Symposium on User Interface Software and Technology* (1989).
- [Rodriguez99] Rodriguez, E.A Optimización Genética de Problemas SAT. *I.T.E.S.M. Campus Morelos* (1999).
- [Selman92] Selman, B., Levesque, H. y Mitchell, D. A New Method for Solving Hard Satisfiability Problems. *Proceedings of the Tenth National Conference on Artificial Intelligence AAAI-92* (1992).
- [Selman95] Selman, B., Mitchell, D. y Levesque, H. Generating Hard Satisfiability Problems. *Elsevier Science* (1995).

- [Tuft98] Tufte, E.R. The Visual Display of Quantitative Information. Sixteenth printing. *Graphics Press* (1998).
- [Tuft98a] Tufte, E.R. Visual Explanations. Third printing. *Graphics Press* (1998).
- [Tuft99] Tufte, E.R. Envisioning Information. Seventh printing. *Graphics Press* (1999).
- [Watanabe93] Watanabe, O Test Instance Generation for Promised NP Search Problems. *Technical Report Dept. of Computer Science, Tokyo Institute of Technology* (1993).

Centro de Información-Biblioteca



30002006011027