

DISEÑO MEDIANTE SATISFACCIÓN DE RESTRICCIONES

Dr. Daniel Gálvez Lio
Lic. María M. García Lorenzo
Lic. Guillermo Sánchez Hernández
Dr. Rafael Bello Pérez*

RESUMEN

En este artículo se analiza el método de solución de problemas de la Inteligencia Artificial conocido por satisfacción de restricciones, su empleo en el diseño y cómo éste fue realizado en el Sistema Inteligente de Ayuda al Diseño (SIAD).

1. La satisfacción de restricciones como un método de diseño.

Al definir la problemática del diseño aparece claramente que la satisfacción del conjunto de restricciones impuestas juega un papel principal a la hora de encontrar la solución deseada. Bajo fuertes condiciones, varias clases particulares de problemas de diseño se pueden formular como problemas de optimización, satisfacción de restricciones, o problemas algebraicos. Lo que es común a todas estas formulaciones es que la solución yace en un espacio determinado por restricciones simultáneas y existen clases específicas de algoritmos para operar sobre este espacio.

En particular, cuando la estructura del diseño está especificada pero los parámetros están determinados por las especificidades de un problema de diseño

resultan útiles las técnicas de optimización simbólicas o numéricas. Algunas versiones del problema de satisfacción de restricciones se pueden resolver por propagación de restricciones. Las restricciones pueden ser propagadas de tal manera que los parámetros convergen incrementalmente hacia un conjunto que satisface todas las restricciones.

Formalmente todo diseño puede verse como un problema de satisfacción de restricciones, en el cual el diseño se crea dentro de límites establecidos de tiempo, costo, materiales, etc., y por lo tanto este método podría considerarse como una solución universal para el diseño. Sin embargo, a menos que se utilice cierto conocimiento para reducir el tamaño del espacio (por ejemplo descomposición del problema) la aplicación del método puede resultar computacionalmente intratable.

2. Algoritmos para la satisfacción de restricciones.

Los problemas de satisfacción de restricciones (PSR) se describen mediante tres componentes: las variables, su conjunto de valores y las restricciones. Resolver un problema de satisfacción de restricciones es

* Departamento de Ciencia de la Computación Facultad de Matemática, Física y Computación Universidad Central de Las Villas, CUBA

determinar una asignación de valores a las variables tal que se satisfacen todas las restricciones.

Un PSR de tamaño n relaciona n variables V_1, \dots, V_n a los conjuntos de valores D_1, \dots, D_n . Las restricciones $P_k, k=1, \dots, L$ se definen sobre el producto cartesiano de esos conjuntos valores. Seguidamente se analizarán métodos de satisfacción de restricciones basados en el enfoque de la Inteligencia Artificial simbólica. Posteriormente se analizará uno basado en el enfoque conexionista.

El primero de los métodos es una versión de la satisfacción de restricciones como una variación del algoritmo de búsqueda conocido como Genera-y-Prueba. Adaptando el método a la problemática del diseño la idea es simplemente generar varios diseños y probarlos para ver si se obtienen los objetivos deseados y no se violan las restricciones:

Satisfacción de Restricciones mediante GENERA-Y-PRUEBA:

P1. Repetir hasta que el generador sea vacío.

- a) Generar el siguiente conjunto de valores posibles para las variables.
- b) Probar el conjunto de variables para ver si se satisfacen todas las restricciones.
- c) Terminar con ÉXITO si todas las restricciones se satisfacen.

P2. Terminar con FALLA si no se encuentra solución.

Otra formulación de un algoritmo de satisfacción de restricciones es la siguiente:

P1. Colocar en OPEN el conjunto de todos los objetos que tienen que tener valores asignados en una solución. Luego iterar hasta que se encuentre una inconsistencia u OPEN sea vacío.

- a) Seleccionar un objeto OB de OPEN. Fortalecer tanto como sea posible el conjunto de restricciones válidas sobre OB.
- b) Si este conjunto es diferente del conjunto que fue asignado la última vez que OB fue examinado o si es la primera vez que OB es examinado, entonces añadir a OPEN todos los objetos que comparten alguna restricción con OB.
- c) Remover OB de OPEN.

P2. Si la unión de las restricciones define una solución, entonces terminar con Éxito.

P3. Si la unión de las restricciones define una contradicción, entonces terminar con Falla.

P4. Iterar hasta hallar una solución o hasta que todas las soluciones posibles hayan sido eliminadas:

- a) Seleccionar un objeto cuyo valor no haya sido calculado y seleccionar una vía de fortalecer las restricciones sobre ese objeto.
- b) Recursivamente invocar la satisfacción de restricciones con el conjunto actual de restricciones aumentado con las restricciones seleccionadas en el paso anterior.

3. Sistema Inteligente de Ayuda al Diseño (SIAD)

Seguidamente se describe un sistema que apoya el proceso de diseño y que emplea algunas de estas técnicas.

3.1 Descripción del sistema

El SIAD es un paquete integrado que utiliza un ambiente interactivo basado en menús jerárquicos enmarcados en ventanas que brindan facilidades para:

- a) Demandar información sobre sistemas creados previamente.

Se utiliza un procesador de lenguaje natural simple.

- b) Interactuar con el Sistema Operativo.

- c) Crear, salvar y cargar bases de conocimientos.

El sistema posee facilidades de edición de textos, edición dirigida por sintaxis, etc.

- d) Editar y salvar los datos iniciales.

El sistema ofrece un editor de pantalla en el cual aparecen todos los nombres de los parámetros, desde él se puede activar el dominio de un parámetro y seleccionar el valor deseado, o escribirlo directamente.

- e) Calcular una variante de diseño.

Mediante un proceso de inferencia a partir de la base de conocimientos se busca inteligentemente una variante de diseño, utilizando una máquina de inferencia dirigida por datos.

Durante el proceso de búsqueda de una solución se puede:

- Detener la búsqueda y luego continuar la misma.
- Mostrar paso a paso el proceso de búsqueda.
- Ejecutar acciones tales como escribir en un fichero texto, ejecutar un programa, automodificar la base de conocimientos, cargar otras bases de conocimientos, generar elementos geométricos hacia el editor gráfico del sistema, etc.

- f) Calcular soluciones de diseño alternativas.

- g) Imprimir la solución, cosa que es posible a través del formato del reporte de salida que se desee o de uno establecido por el sistema.

- h) Evaluar la solución.

Las facilidades gráficas del paquete integrado se expresan en tres direcciones:

- Editor gráfico simple que permite crear, salvar y recuperar gráficos.
- Generar dibujos hacia el editor gráfico a partir de la semántica geométrica incorporada en los nombres de los parámetros, o a través de las acciones que se ejecutan durante la inferencia.
- Generar ficheros textos con la información necesaria para comunicarse con otros paquetes gráficos profesionales.

3.2 Forma de representación del conocimiento del SIAD

La forma de representación del conocimiento está definida en términos de la combinación de frames y reglas de producción.

En la forma de representación definida, el modelo físico del producto se describe usando dos tipos de frames. El primer tipo permite almacenar el conocimiento sobre el producto como un todo y se denomina Frame-Objeto, mientras el otro tipo se utiliza para representar el conocimiento sobre cada parámetro del producto por lo que se llama Frame-Parámetro. Por lo tanto, una base de conocimientos que almacene el modelo físico de un producto está constituida de un Frame-Objeto y tantos Frame-Parámetro como parámetros de diseño tenga el producto.

3.3 Método de búsqueda de soluciones.

El cálculo de una solución de diseño se realiza mediante un proceso de inferencia dirigido por datos (forward-chaining). El diseñador

tiene la posibilidad de fijar valores para algunos de los parámetros de diseño y luego activar el proceso de inferencia para encontrar una solución. Se asume que se ha obtenido una solución de diseño cuando todos los parámetros del modelo tienen valor.

Para el cálculo del valor de un parámetro se utiliza un procedimiento de cálculo (PC). Un procedimiento de cálculo puede ser utilizado cuando cumple las condiciones siguientes:

- todos los parámetros de diseño que aparecen como operandos en el mismo tienen valor.
- si el PC está expresado mediante una regla de producción, para ser utilizado tiene que ser verdadera la condición.

Cada vez que se evalúa un parámetro de diseño se comprueba si las restricciones donde aparece como operando se cumplen.

Una restricción se cumple si:

- todos sus operandos tienen valor y al evaluar la expresión lógica que la define, ésta da como resultado el valor verdadero.
- alguno de sus operandos no tiene valor.

Para realizar el proceso de inferencia se crea una lista con todos los Frame-Parámetro, en el mismo orden en que aparecen en la base de conocimientos. La esencia del proceso de inferencia es: buscar el primer parámetro sin valor, que al tratar de evaluarlo se pueden obtener tres resultados: se calculó satisfactoriamente, no se pudo evaluar pero existen PC que no se pueden utilizar por tener operandos sin valor o no se pudo evaluar aunque se utilizaron todos sus PC; en el primer caso el proceso de inferencia continúa buscando otro parámetro a evaluar desde el inicio de la lista, en el segundo se continúa el proceso de inferencia desde el parámetro

siguiente en la lista y en el tercero es necesario realizar un retroceso para recalcular alguno de los parámetros antecesores (backtracking).

El proceso de inferencia puede terminar por tres razones: fueron evaluados todos los parámetros; existen parámetros sin calcular pero no pueden calcularse pues sus PC tienen operandos sin valor o en el proceso de recálculo se llegó hasta tratar de recalcular parámetros evaluados directamente por el diseñador (datos iniciales). La segunda razón indica que faltan datos iniciales y la tercera que con los datos iniciales dados es imposible obtener una solución de diseño.

En resumen, el método de búsqueda de soluciones utilizado es una inferencia dirigida por datos (forward-chaining) con posibilidades de retroceso (backtracking).

Este método puede verse como una variante del método de satisfacción de restricciones derivado de Genera-y-Prueba en la cual los pasos 1a y 1b se superponen pues cada vez que se genera un valor para una variable se verifican todas las restricciones donde esta variable aparece, por lo que esencialmente es una combinación de la búsqueda dirigida por dato con satisfacción de restricciones y retrocesos (backtracking).

El retroceso es otro aspecto de la satisfacción de restricciones mediante el cual cada vez que se ejecuta P1a se desvalorizan las variables y se generan nuevos valores; en el caso del SIAD no se desvalorizan todas sino las más recientemente calculadas tratando de no perder la versión parcial del diseño del artefacto alcanzada hasta ese momento. El retroceso, o desvalorización de variables, se realiza desde la última variable calculada y con la cual se viola una restricción y

sucesivamente se desvalorizan las demás variables que aparecen en dicha restricción siguiendo el orden inverso al que fueron evaluadas.

Por ejemplo, sean las variables A, B, C y D sujetas a la restricción $A+B > C+D$; si en la etapa actual (numero 51) se calcula un valor para A que viola la restricción y las variables B, C y D habían sido calculadas en las etapas 10, 27 y 38 respectivamente, entonces se buscará otro valor para A; si este valor no existe se desvaloriza D y se intenta encontrar valores para D y A; si no hay solución se desvaloriza y recalcula C, y así sucesivamente.

Esta es una variación del algoritmo que permite no perder todo el trabajo hecho anteriormente, lo cual ocurre si en el paso P1a se recalculan todas las variables.

El SIAD ofrece otra variante de control del retroceso en la cual las variables pueden tener asociadas sus antecesores explícitamente, es decir, se indica cuál variable se debe recalculan cuando resulta imposible encontrar un valor que no viole las restricciones para una variable dada.

Otro aspecto de interés en el proceso de diseño es la búsqueda de soluciones alternativas.

Comúnmente se utilizan dos métodos para la búsqueda de soluciones alternativas, uno de ellos es eliminar todos los nodos del espacio de estado (espacio de búsqueda) que aparecen en el camino que llevó a la solución encontrada y el otro es eliminar el último nodo de este camino; seguidamente

se repite el proceso de inferencia como si la solución a calcular fuera la primera. En el caso del dominio de conocimiento presentado anteriormente, la primera variante significa eliminar del conjunto de PC de todos los parámetros calculados el PC utilizado en el cálculo del mismo, mientras la segunda variante significa eliminar del último parámetro calculado el PC usado para evaluarlo.

El método de búsqueda de soluciones alternativas utilizado es una variante del segundo. Se ordena la lista de parámetros en orden decreciente de la etapa del proceso de inferencia en que fueron evaluados, o sea, el último calculado es el primero de la lista, se elimina el PC utilizado para calcularlo y se activa el proceso de inferencia.

El proceso se puede repetir mientras existan parámetros a los cuales se le puedan eliminar PC y se puede obtener una nueva solución.

CONCLUSIONES

La esencia de la problemática de diseño es encontrar un producto que satisface un objetivo y existirá en un entorno el cual se fija límites a las dimensiones y funciones del mismo. Por eso el diseño puede verse como un problema de satisfacción de restricciones.

En este artículo se describe este método de solución de problemas de la Inteligencia Artificial y se analiza una versión del mismo desarrollada para construir el Sistema Inteligente de Ayuda al Diseño (SIAD).

BIBLIOGRAFIA

- BELLO P., Rafael et al. *An Intelligent System to Help Designer*. Applications of Computer to Engineering design, manufacturing and management, editado por G.L. Lastra, J.L. Encarnasao y A.A.G. Requich, Elsevier Science Publishers, B.V. (North Holland), 1989.
- BELLO P., Rafael, D. GÁLVEZ L., M.M. GARCÍA, G. SÁNCHEZ. *Integrated package oriented to intelligent CAD system implementation*. Proceeding of Conference on CAD/CAM Technology Transfer tp Latin America, IFIP, 1990.
- BELLO P., Rafael, D. GÁLVEZ L., M.M. GARCÍA, G. SÁNCHEZ. *A knowledge representation form for mechanical engineering*. Knowledge-based system, Vol 7, No 3, sept 1994.
- DYM C., L. y E. LEVITT . *Knowledge-Based System in Engineering*. McGraw Hill, 1991.
- D'AMBROSIO, J.G. y W.P. BIRMINGHAM. Preference-directed design. *Artificial Intelligence for engineering design, analysis and manufacturing*, 1995, No 9, pp. 219-230.
- KIMURA, F. y H. SUZUKI. *Varational product design by constraint propagation and satisfaction in product modelling*. Annals of de CIRP, Vol 35, No 1, 1986.
- RUSSELL, S. y P. NORRIG. *Artificial Intelligent a modern approach*. Prentice Hall, 1995.
- SHAPIRO, S.C. *Encyclopedia of Artificial Intelligence*. A Wiley-Interscience Publications, John Wiley & Sons, 1990.