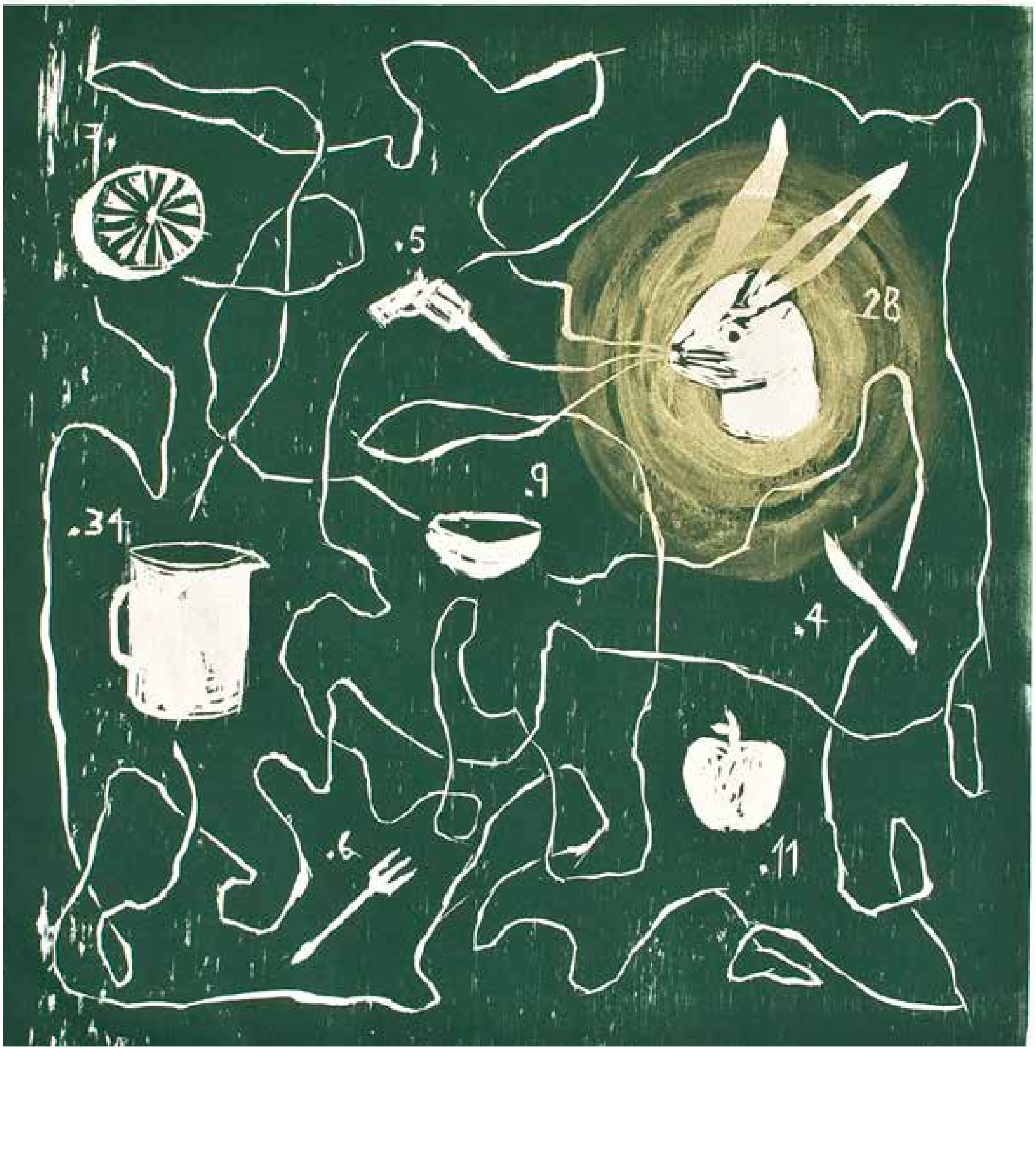



*Comida para conejos 5*. Xilografía, 2010

# Optimización combinatoria

◆ **Marco A. Cruz Chávez**
**Pedro Moreno Bernal**
**Martín Martínez Rangel**

En las sociedades modernas se presentan cada vez con mayor frecuencia problemas de distintos tipos, y para resolverlos hacen falta pruebas experimentales. Elaborar un modelo matemático que se asemeje a la realidad para representar un problema es muy complicado, ya que en muchos de ellos el número de variables puede ser muy grande. Por otro lado, dichos problemas muestran comportamientos subjetivos difíciles de controlar, por lo que hacen aún más complicado que dichos modelos se asemejen a la realidad.

Desde los años treinta, en diferentes países se ha ensayado la solución de problemas por medio de la investigación de operaciones (IO). Los orígenes de la IO son muy remotos, ya que muchas sociedades organizadas siempre se han planteado cuestiones de optimización en el uso de recursos y en la planificación de actividades por realizar.

A mediados de los años cuarenta, George Dantzig publica sus trabajos de planificación de tareas dentro del Pentágono (Estados Unidos), donde introduce explícitamente la función-objetivo en el modelo matemático.[1] Dantzig propuso el método Simplex para tratar problemas modelados en sistemas de ecuaciones lineales. Nació así la programación lineal, y con ella la programación matemática, también llamada optimización combinatoria. De hecho, el concepto tan usado de "programa" tiene sus orígenes en que así es como se denominan los proyectos en el argot militar. Con la aparición de las computadoras electrónicas, las cuales tienen un papel clave en este proceso, la programación matemática se fortalece y comienza a producir resultados exitosos mediante la implementación de algoritmos para computadora.

Como hemos mencionado, la mayor parte de los problemas precisan relajaciones de dimensiones moderadas, sobre todo si compartimos la idea de que el mundo es no lineal y estocástico (elecciones probabilísticas). Algunos problemas o fenómenos de la naturaleza pueden ser descritos como sistemas dinámicos. La teoría del caos hace precisos los caminos en que muchos

[1] Juan José Salazar González, *Programación matemática*, Díaz de Santos, Madrid, 2001, pp. 8-11.

◆ Profesor e investigador, Centro de Investigaciones en Ingeniería y Ciencias Aplicadas (Ciicap), UAEM
Profesor e investigador, Facultad de Contaduría, Administración e Informática (FCAeI), UAEM
Posgrado en Ingeniería y Ciencias Aplicadas, Centro de Investigaciones en Ingeniería y Ciencias Aplicadas (Ciicap), UAEM


de estos sistemas muestran un comportamiento determinista e impredecible.

Las matemáticas aplicadas, junto con la estadística, proveen metodologías que son utilizadas en las ciencias, ingeniería, economía y en la industria, basadas en el conocimiento para estudiar problemas de decisión, tales como reducir el riesgo o una función objetivo, estimar parámetros o pruebas de hipótesis, y seleccionar la mejor solución. En este sentido, se hace uso de la optimización para resolver problemas de IO, teoría del control y matemáticas económicas, entre otras.[2] Por lo tanto, las ciencias computacionales proponen y analizan métodos para resolver problemas matemáticos que requieren procesar o analizar una gran cantidad de datos, difíciles para los humanos. También permiten simular procesos y obtener pruebas experimentales suficientes que logran explorar el mayor número de soluciones posibles, las cuales permiten ahorrar costos de experimentación física y, a su vez, obtener datos estadísticos para futuras tomas de decisiones sobre las consecuencias de la implementación de procesos.

**Métodos de solución**

Los métodos de resolución de sistemas de ecuaciones lineales se pueden dividir en dos grupos: los *métodos exactos* o algoritmos finitos, que permiten obtener la solución del sistema de manera directa, y los *métodos aproximados*, que utilizan algoritmos iterativos e infinitos que calculan la solución del sistema por aproximaciones sucesivas.

En muchas ocasiones, los métodos aproximados permiten obtener un grado de exactitud superior del que se puede obtener de los métodos exactos, debido fundamentalmente a los errores de truncamiento. Es decir, al dividir en distintas partes el problema, el método de aproximación de forma iterativa permite explorar distintas partes del problema dentro de un mayor espacio de soluciones, lo cual en muchas ocasiones abre la posibilidad de encontrar una mejor solución mediante el uso de algoritmos.

Un algoritmo es un conjunto ordenado y finito de operaciones que permite hallar la solución de un problema. La importancia de encontrar algoritmos eficientes ya era conocida mucho antes de la época de las computadoras electrónicas. Un caso muy conocido es el algoritmo de Euclides para calcular el máximo común divisor. A lo largo de la historia, varios autores han tratado de definir formalmente los algoritmos utilizando modelos matemáticos, como máquinas de Turing, entre otros.[3]

Los algoritmos de interés para las ciencias computacionales son aquellos que se van a utilizar en una computadora. Sin embargo, hay otros métodos sistemáticos para calcular un resultado, por ejemplo, los métodos que aprendemos en la escuela para sumar, multiplicar y dividir. Una excepción de esta regla permite admitir como algoritmo los procedimientos que efectúan elecciones aleatorias acerca de lo que se debe hacer en una situación dada. Estos algoritmos se denominan "algoritmos estocásticos" o "probabilistas".

---

[2] C. Radhakrishna Rao (ed.), *Statistics and truth: putting chance to work*, World Scientific, 2a ed., Singapur, 1997, p. 192.

[3] Michael Sipser, *Introduction to the theory of computation*, Thomson Course Technology, 2ª ed., Boston, 2006, p. 23.

Pero el término "aleatorio" no quiere decir "arbitrario"; por el contrario, utilizamos valores seleccionados de tal manera que la probabilidad de seleccionar cada uno de los valores es conocida y está controlada.[4]

De forma general, podemos clasificar los algoritmos en dos grupos: los *algoritmos deterministas*, en los cuales en cada paso se determina de forma única el siguiente paso, y los *algoritmos no deterministas*, que deben decidir en cada paso de la ejecución entre varias alternativas y agotarlas todas antes de encontrar la solución. En este contexto, los algoritmos no deterministas finalizan su ejecución con *admitir* o *rechazar* dos instrucciones especiales, o bien, pueden quedar en un bucle infinito. Por lo tanto, los algoritmos no deterministas no deben confundirse con los algoritmos probabilistas.

Todos los algoritmos tienen una serie de características entre las cuales hay algunas que requieren de una serie de recursos al momento de implementarlos en una máquina. Estos recursos son principalmente el tiempo y la memoria. A través de la teoría de la complejidad algorítmica se intenta responder a preguntas del tipo: ¿qué tan bueno es un algoritmo para resolver un problema? o ¿qué tan difícil es intrínsecamente un problema dado? A pesar de esta intrínseca naturaleza compleja, algunos de dichos problemas necesitan algoritmos para su solución. Un enfoque teórico (*a priori*) consiste en determinar matemáticamente la cantidad de recursos necesarios para cada uno de los algoritmos como función del tamaño de los casos considerados. El enfoque empírico (*a posteriori*) consiste en programar las técnicas competidoras e ir probándolas en distintos casos con ayuda de una computadora.[5]

Aunque un problema sea muy difícil de resolver, puede evitarse perder tiempo buscando un algoritmo, si bien esto no hace que el problema desaparezca. Se tiene que hallar algún tipo de solución para el problema, sea difícil o no. Estos son los dominios de los algoritmos aproximados, como las heurísticas y metaheurísticas.

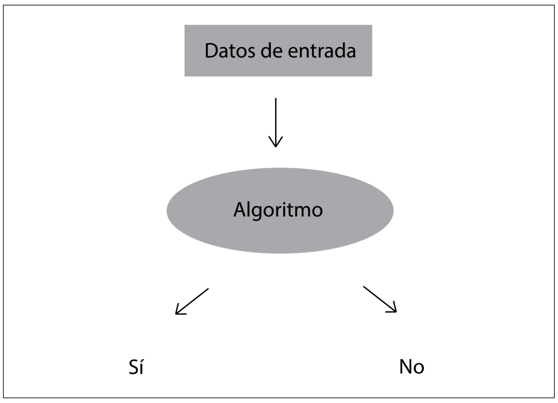
Una heurística se puede definir como un método con reglas empíricas para encontrar soluciones de problemas, el cual está basado en la experiencia y no tiene pruebas de optimalidad. Una metaheurística es una estrategia de alto nivel aplicada a problemas combinatorios que mejora óptimos locales, guiando el proceso de búsqueda que permite encontrar buenas soluciones.

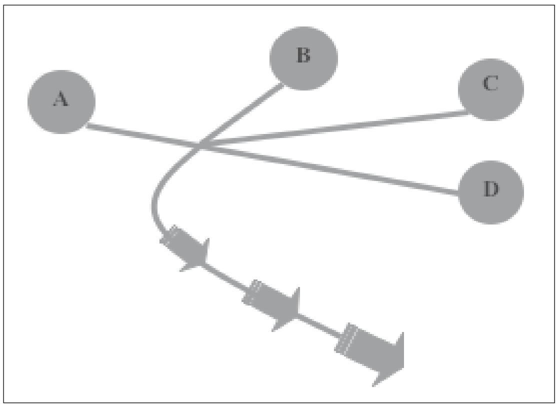
**Tipos de problemas**

Existen problemas para los cuales hasta el mejor algoritmo posible requiere una cantidad de tiempo exorbitante, incluso para casos pequeños. Postularemos que un algoritmo es eficiente si su complejidad, es decir, el conjunto de operaciones, puede ser reducido y representado a través de un polinomio, de tal forma que el algoritmo pueda resolver cualquier caso de tamaño *n* en un tiempo determinado. Los algoritmos de tiempo exponencial se vuelven rápidamente inútiles en la práctica,

4 A. S. Para-Vasquez y R. V. Oakford, "Simulations as a technique for comparing decision procedures", *The engineering economist*, vol. 21, núm. 4, 1976, pp. 221-236.

5 Bratley Brassard, *Fundamentos de algoritmia*, Pretince Hall, Madrid, 1997.




Figuras 1 y 2. Esquemas de un problema de decisión.

mientras que un algoritmo de tiempo polinómico nos permite resolver casos mucho más grandes.[6]

En los problemas de decisión existe una función que relaciona un conjunto de datos de entrada con el conjunto *X = {si, no}*, equivalente a verdadero o falso, como se muestra en la figura 1. Por ejemplo, buscar un ciclo hamiltoniano en un grafo no es un problema de decisión, pero la respuesta a la pregunta: ¿es hamiltoniano el grafo?, sí que lo es. Si el problema de decisión define un conjunto *X* de casos en que la respuesta es "sí", todos los demás casos son "no". Por lo tanto, un algoritmo correcto que resuelva el problema de decisión *acepta* el caso "sí" y *rechaza* los casos "no".

Tomando el ejemplo anterior, un problema de optimización es encontrar el mejor camino para realizar el recorrido hamiltoniano en el grafo. Como se observa en la figura 2, cada arista del grafo representa la conexión entre dos nodos, con un costo asociado. Supongamos que los nodos del grafo representan ciudades y las aristas el tiempo en llegar de una ciudad a otra. La pregunta: ¿cuál es el camino más corto para ir del nodo de inicio al nodo final?, es un problema de optimización combinatoria que requiere minimizar los costos. Por lo tanto, se deben evaluar todas las combinaciones posibles de los diferentes caminos y, según sea el caso, se elegirá como la mejor solución la respuesta que satisfaga el objetivo de reducir costos.

Este problema es conocido en la literatura especializada como el "problema del agente viajero" (TSP, por sus siglas en inglés), el cual, de manera formal, consiste en determinar un ciclo hamiltoniano en un grafo no orientado, con un costo asociado a cada arista, cuyas aristas tengan un costo

[6] Michael R. Garey y David S. Johnson, *Computers and intractability: a guide to the theory of NP-completeness*, Bell Telephone Laboratories Inc., Murray Hill, 1979, pp. 5-15.

total mínimo. Se suele asumir que el grafo es completo y, en el contexto del TSP, a cada ciclo hamiltoniano se le llama también "*tour*" o "recorrido".

Este problema aparece en numerosas aplicaciones en las cuales hay que secuenciar un conjunto de tareas. Cada una de ellas tiene asociado un costo de procesamiento, una inmediatamente después de la otra. El TSP ha sido un problema de especial relevancia en la optimización combinatoria desde sus inicios, por su definición simple y su dificultad implícita de resolución.

Varios de los problemas de optimización pueden modelarse mediante el uso de herramientas de abstracción alternativas a los modelos matemáticos basados en variables e inecuaciones. Estas herramientas son los grafos. La teoría de grafos permite describir algoritmos que aprovechan la estructura combinatoria de sus problemas.

Existen diferentes problemas de flujos sobre redes comprobados como problemas polinomiales en la literatura especializada. Los algoritmos para resolverlos son especializaciones de algoritmos de programación lineal, como el problema del árbol generador de costo mínimo, por ejemplo. Los requerimientos de tiempo de un algoritmo son convenientemente expresados en términos de una sola variable: el "tamaño" de una instancia del problema es la cantidad de datos de entrada necesarios para describir dicha instancia.

En la Universidad Autónoma del Estado de Morelos (UAEM), el Cuerpo Académico Consolidado en Optimización y Software realiza investigaciones para encontrar algoritmos más eficientes que permitan resolver diferentes tipos de problemas, tanto de la literatura especializada como aquellos reales que se presentan en la sociedad o en la industria. Se busca siempre que dichos algoritmos sean eficientes, lo cual involucra todos los recursos de cómputo necesarios para ejecutarlos. Esto significa que el algoritmo más eficiente es el más rápido, lo cual constituye un factor determinante.

En el Centro de Investigaciones en Ingeniería y Ciencias Aplicadas (Ciicap) se encuentra el Laboratorio de Cómputo Intensivo Grid Morelos,[7] iniciativa del doctor Marco A. Cruz Chávez. Actualmente, en dicho laboratorio están en producción las siguientes aplicaciones: "Algoritmo genético híbrido cooperativo en ambiente grid para talleres con flujo flexible", "Algoritmo evolutivo en ambiente grid para el problema de redes de distribución de agua", entre otras. En dichos proyectos participan investigadores de las facultades de Ciencias (FC), Ciencias Químicas e Ingenierías (FCQeI), y Contaduría, Administración e Informática (FCAeI) de la UAEM. Con este nodo de alto rendimiento, la UAEM se coloca a la vanguardia en infraestructura de cómputo de alto rendimiento.

---

[7] Grid Morelos, http://www.gridmorelos.uaem.mx/