

CAPÍTULO II

2. OPTIMIZACIÓN COMBINATORIA

2.1 RESUMEN

La optimización combinatoria es un campo vivo de la matemática aplicada, que estudia el modelado combinando técnicas de combinatoria, programación lineal, lineal-entera, la teoría general de problemas extrémales y la teoría de algoritmos, con el propósito de encontrar una solución algorítmica de problemas donde se busca optimizar (maximizar o minimizar) una función de varias variables definidas sobre un conjunto discreto.

Esta disciplina tiene numerosas aplicaciones a problemas que se presentan en la industria, logística, ciencias, ingenierías y en la administración de organizaciones.

La versatilidad del modelo de la optimización combinatoria resulta del hecho en que muchos problemas prácticos, actividades y recursos, tales como máquinas, aviones y personas son indivisibles. También, muchos problemas tienen únicamente un número finito de alternativas y consecuentemente pueden apropiadamente ser formuladas como problemas de optimización combinatoria (la palabra combinatoria se refiere al hecho que únicamente existen un número finito de soluciones factibles).

Como ejemplos podemos mencionar, entre otros, el ruteo y carga de vehículos en redes de distribución, el diseño de redes de telecomunicación, la planificación de la producción, la selección de carteras financieras, la asignación de tareas a procesadores, el análisis de estructuras moleculares, las subastas de frecuencias para radiotransmisión, la asignación de tripulaciones en líneas aéreas, la planificación de la generación de electricidad y la distribución de ambulancias en una región para asegurar un cierto nivel de servicio a su población. La gran variedad de posibles

aplicaciones es una fuente constante de problemas nuevos para la investigación en este área.

El término combinatorio se entiende en el sentido de que el conjunto factible es de gran talla

Un problema de programación lineal consiste en hallar el valor óptimo de una función objetivo lineal cuyas variables están sujetas a restricciones lineales. Si además se exige que las variables tomen valores enteros, entonces se tiene un problema de programación lineal-entera.

Varios problemas de Optimización Combinatoria pueden ser resueltos en tiempo polinomial utilizando los métodos y teoría de la programación lineal.

En el caso de los problemas NP-Complejos, los métodos más eficaces en la actualidad para encontrar una solución exacta a muchos de estos problemas utilizan, típicamente, la programación lineal-entera o técnicas de ramificación y acotamiento y generación de columnas.

2.2 COMPLEJIDAD COMPUTACIONAL

Las Ciencias Computacionales tratan en mayor o en menor grado problemas de "complejidad computacional", que informalmente se puede describir como el costo requerido para encontrar la solución a un problema en términos de recursos computacionales (fundamentalmente memoria o tiempo de cómputo).

Los procedimientos desarrollados para la solución de problemas son llamados algoritmos, y la complejidad computacional, trata de encontrar los algoritmos más eficientes para resolver cada problema. La eficiencia hace referencia a todos los recursos computacionales, sin embargo la eficiencia puede ser tomada como sinónimo de rapidez.

La teoría de complejidad plantea los problemas como casos de decisión para los cuales su solución corresponde a una respuesta SI/NO

2.3 PROBLEMAS P, NP Y NP-COMPLETOS

Para los problemas de carácter combinatorio (donde el dominio de soluciones es finito y se elige la mejor solución posible de éste), existen distintas formas de resolverlos, una de ellas es la búsqueda exhaustiva del conjunto de soluciones y así poder encontrar la óptima, es decir generar todas las soluciones factibles (que cumplan con todas las restricciones), calcular su costo respectivo asociado y de éstas elegir la mejor. Pero el tiempo de cálculo crece de manera exponencial de acuerdo al número de items del problema.

Podemos encontrar problemas en que se produce una explosión combinatoria (donde el tiempo de ejecución es no polinomial), de acuerdo al tamaño del problema, de los que solo se conocen algoritmos que encuentran una solución exacta en tiempos excesivamente largos.

Cuando nos enfrentamos a un problema concreto, habrá una serie de algoritmos aplicables. Se suele decir que el orden de complejidad de un problema es el del mejor algoritmo que se conozca para resolverlo. Así se clasifican los problemas, y los

estudios sobre algoritmos que se aplican a la realidad. Estos estudios han llevado a la constatación de que existen problemas muy difíciles, problemas que desafían la utilización de los ordenadores para resolverlos. En lo que sigue esbozaremos las clases de problemas que hoy por hoy se escapan a un tratamiento informático.

Clase P. Los algoritmos de complejidad polinómica se dice que son tratables en el sentido que suelen ser abordables en la práctica. Los problemas para los que se conocen algoritmos con esta complejidad se dice que forman la clase P. Aquellos problemas para los que la mejor solución que se conoce es de complejidad superior a la polinómica, se dice que son problemas intratables. Sería muy interesante encontrar alguna solución polinómica (o mejor) que permitiera abordarlos.

Clase NP. Algunos de estos problemas intratables pueden caracterizarse por el curioso hecho de que puede aplicarse un algoritmo polinómico para comprobar si una posible solución es válida o no. Esta característica lleva a un método de resolución no determinista consistente en aplicar heurísticos para obtener soluciones hipotéticas que se van desestimando (o aceptando) a

ritmo polinómico. Los problemas de esta clase se denominan NP (la N de no-determinísticos y la P de polinómicos). Es evidente que $P \subseteq NP$.

Clase NP-completos. En 1971 Cook demostró que hay problemas en NP que son especialmente difíciles, son los denominados NP-completos. Se conoce una amplia variedad de problemas de tipo NP, de los cuales destacan algunos de ellos de extrema complejidad. Gráficamente podemos decir que algunos problemas se hallan en la "frontera externa" de la clase NP. Son problemas NP, y son los peores problemas posibles de clase NP. Estos problemas se caracterizan por ser todos "iguales" en el sentido de que si se descubriera una solución P para alguno de ellos, esta solución sería fácilmente aplicable a todos ellos.

Clase NP-duros. Cualquier problema de decisión, pertenezca o no a los problemas NP, el cual pueda ser transformado a un problema NPC (NP-completo) tendrá la propiedad que no podrá ser resuelto en tiempo polinomial a menos que $P=NP$. Podríamos entonces decir que dicho problema es al menos tan difícil como uno NP Completo.

2.4 ALGORITMO DE CAMINOS MAS CORTOS (DIJKSTRA)

Dijkstra es un algoritmo que encuentra caminos más cortos desde un nodo origen s hacia todos los nodos en una red, con pesos no negativos en los arcos, es decir encuentra una arborescencia de caminos más cortos.

El pseudocódigo del Algoritmo de Dijkstra se muestra a continuación en el Algoritmo 2.1, donde se comienza etiquetando todos los nodos como infinito y llamando a esta etiqueta $d(i)$ para cada nodo i , la cual representa la distancia más corta desde el nodo origen hasta el nodo i , a excepción del nodo s que se etiqueta con 0. Se tiene además que identificar dos conjuntos S y \bar{S} que representan los nodos etiquetados definitivamente y los nodos etiquetados temporalmente respectivamente, estos conjuntos van cambiando de acuerdo a cada iteración del Algoritmo según el siguiente criterio: Sea $i \in \bar{S}$ tal que $d(i) = \min \{d(j) : j \in \bar{S}\}$, actualizándose así los dos conjuntos S y \bar{S} . El criterio de parada es cuando todos los nodos están etiquetados definitivamente.

Algoritmo 2.1: ALGORITMO DIJKSTRA

BEGIN

$S := f; \bar{S} := N;$

$d(i) := \infty$ para cada nodo $i \in N;$

$d(s) := 0$ y $\text{predecesor}(s) := 0;$

WHILE $|S| < n$ *DO*

BEGIN

Sea $i \in \bar{S}$ tal que $d(i) = \min \{d(j) : j \in \bar{S}\};$

$S := S \cup \{i\};$

$\bar{S} := \bar{S} - \{i\};$

FOR EACH $(i, j) \in A(i)$ *DO*

IF $d(j) > d(i) + c_{ij}$ *THEN* $d(j) := d(i) + c_{ij}$ y $\text{predecesor}(j) := i;$

END;

END;

END;

END;

END;

2.5 HEURISTICAS

En los últimos años ha habido un crecimiento espectacular en el desarrollo de procedimientos heurísticos para resolver problemas combinatorios. Este hecho puede ser constatado examinando el gran número de artículos en revistas de Investigación Operativa en los que se proponen y estudian métodos heurísticos. Además, han aparecido publicaciones específicas para el estudio y divulgación de dichos procedimientos tales como Journal of Heuristics.

El auge que experimentan los procedimientos heurísticos se debe sin duda a la necesidad de disponer de herramientas que permitan ofrecer soluciones rápidas a problemas reales. Es importante destacar el hecho de que los algoritmos heurísticos (por sí solos) no garantizan la optimalidad de la solución encontrada, aunque su propósito es encontrar una solución cercana al óptimo en un tiempo razonable. Sin embargo, la gran cantidad de publicaciones en donde problemas de gran dificultad son resueltos con gran rapidez (en muchos casos óptimamente), avalan estos métodos.

Son procedimientos simples, a menudo basados en el sentido común, que se supone ofrecerán una buena solución (aunque no

necesariamente la óptima) a problemas difíciles, de un modo fácil y rápido.

Los factores que pueden hacer interesante la utilización de heurísticas para la resolución de un problema pueden ser los siguientes:

1. Cuando no existe un método exacto de resolución o éste requiere mucho tiempo de cálculo o memoria.
2. Cuando no se necesita la solución óptima
3. Cuando los datos son poco fiables
4. Cuando hay limitaciones de tiempo
5. Cuando es paso intermedio en la aplicación de otro algoritmo

Por otro lado la implementación de heurísticas presenta ciertos inconvenientes una de ellas es que por lo general no es posible conocer la calidad de la solución, es decir cuan cerca está del óptimo.

2.5.1 Clasificación de Heurísticas

Según el modo en que buscan y construyen sus soluciones, podemos clasificarlas de la siguiente manera:

a) Métodos constructivos

Consisten en ir poco a poco añadiendo componentes individuales a la solución hasta que se obtiene una solución factible. El más popular de éstos métodos lo constituyen los algoritmos “Glotonos”, los cuales construyen paso a paso la solución buscando el máximo beneficio en cada paso.

b) Métodos de descomposición

Este tipo de algoritmo trata dividir el problema en subproblemas más pequeños, siendo la salida del uno la entrada del siguiente, de forma que al resolverlos se obtenga una solución para el problema global.

c) Métodos de reducción

Tratan de identificar alguna característica que pueda poseer la solución óptima y así hacer más simple el problema, por

ejemplo si algunas variables están altamente correlacionadas o si tomarán el valor de 0 siempre, etc.

d) Métodos de búsqueda por entornos

Aquí se encuentran la mayoría de las heurísticas, parten de una solución factible inicial (obtenida quizás mediante otra heurística) y mediante modificaciones de esa solución, van pasando de forma iterativa, y mientras no se cumpla algún criterio de parada determinado, a otras factibles de su entorno o vecindad, almacenando como óptima la mejor de las soluciones visitadas.

Definición 2.4.- Entorno o Vecindad de la solución S

Es el conjunto $N(s)$ de soluciones parecidas a s . El significado de “parecidas” debe entenderse como la posibilidad de obtener una solución $s' \in N(s)$ a partir de la s realizando solo una operación elemental, llamada movimiento, sobre s (eliminar o añadir un elemento en el subconjunto solución, intercambiar elementos en la permutación s , etc.).

2.6 HEURISTICAS DE ÉXITO (META-HEURISTICAS)

Dado que estos métodos generales sirven para construir o guiar el diseño de métodos que resuelvan problemas específicos se les ha dado el nombre de Meta heurísticos. Los profesores Osman y Kelly (1995) introducen la siguiente definición:

"Los procedimientos Meta heurísticos son una clase de métodos aproximados que están diseñados para resolver problemas difíciles de optimización combinatoria, en los que los heurísticos clásicos no son ni efectivos ni eficientes. Los Meta heurísticos proporcionan un marco general para crear nuevos algoritmos híbridos combinando diferentes conceptos derivados de: inteligencia artificial, evolución biológica y mecanismos estadísticos"

- Algoritmos Genéticos
- Búsqueda Tabú
- Redes Neuronales
- Templado Simulado

a) Algoritmos Genéticos

Son técnicas de búsqueda basadas en la mecánica de la selección natural y la genética [Goldberg, 1989]. Su estructura

se ha diseñado en un intento de abstracción artificial del algoritmo de selección propio de la naturaleza, con la esperanza de que así se consigan éxitos similares en relación con la capacidad de adaptación a un amplio número de ambientes diferentes.

La información hereditaria en los organismos biológicos es pasada a través de los cromosomas que contienen la información de todos esos factores, es decir, los genes, los cuales a su vez están compuestos por un determinado número de "valores". Varios organismos se agrupan formando una población, y aquellos que mejor se adaptan son aquellos que tienen mayor probabilidades de sobrevivir y reproducirse. Algunos de los supervivientes son seleccionados para crear nuevos organismos. Además los genes de un cromosoma pueden sufrir cambios produciendo mutaciones en los organismos.

En los algoritmos genéticos los "valores" por lo general son binarios (0 o 1), representando valores de las variables de decisión, que se correspondan a los genes. Los cromosomas

representan a las soluciones, en otras palabras los cromosomas no son más que una cadena de bits.

Un algoritmo genético aplica operaciones de reproducción, sobrecruzamiento y mutación a una población y generar nuevos organismos de forma iterativa. La reproducción se guía a través de una función que mide el grado de adaptación del cromosoma y el proceso de selección es dependiente de los valores que esa función le asigne: a mayor valor, mayor probabilidad de selección y supervivencia y los miembros de la nueva selección de cromosomas que son emparejados de nuevo aleatoriamente. Para sobre cruzar se intercambian genes entre las parejas, dividiendo cada una de las cadenas de bits en dos subcadenas y luego intercambiar valores de las últimas subcadenas, obteniendo así dos nuevos cromosomas de una nueva generación. Las mutaciones consisten en alterar un bit de un cromosoma (de 0 a 1 o viceversa), con una probabilidad mas o menos pequeña. Todo esto se repite hasta que algún criterio de parada se cumpla.

b) Búsqueda Tabú

Es un tipo de búsqueda por entornos. Guía un procedimiento de búsqueda local para explorar el espacio de soluciones más allá del óptimo local [Glover, 1994]. La búsqueda Tabú selecciona de una manera agresiva el mejor de los movimientos posibles a cada paso. Al contrario que en la búsqueda local que siempre se mueve al mejor de su entorno y finaliza con la llegada a un óptimo local, la búsqueda Tabú permite moverse a una solución de su entorno o vecindad que no sea tan buena como la actual, de tal manera que pueda tener oportunidad de salir de óptimos locales y continuar estratégicamente la búsqueda de soluciones aún mejores.

Para evitar ciclos (volver a un óptimo local) clasifica un determinado número de los más recientes movimientos como “movimientos Tabú”. Por lo tanto el escape de los óptimos locales se produce de manera sistemática y no aleatoria. Es decir la búsqueda Tabú mantiene una memoria de eventos. Actualmente, el método consiste en tratar de explotar la memoria adaptativa para así controlar el proceso de búsqueda.

La filosofía de la búsqueda Tabú es la creencia de que la elección de una mala estrategia sistemática de búsqueda es mejor que una buena elegida al azar.

c) Redes Neuronales

Surgieron dentro del proceso de tratar de entender el comportamiento del cerebro humano, viéndolo como un sistema de procesamiento de la información de tipo altamente complejo, no lineal y en paralelo. Los modelos de redes neuronales intentan conseguir unos buenos resultados basándose en una interconexión de unos nodos computacionales llamados neuronas.

“Es un procesador distribuido paralelo que posee una propensión natural para el almacenamiento de conocimiento experimental haciéndolo posible para su uso. Recuerda al cerebro humano en dos aspectos: el conocimiento se adquiere mediante un proceso de aprendizaje, y la conexión interneuronal se utiliza para el almacenamiento del conocimiento” [Aleksander y Morton, 1990].

Otra de las heurísticas muy conocidas y que tiene ganado un lugar importante en la resolución de problemas difíciles es el Recocido Simulado, el que se implementará para el desarrollo de esta tesis. Es por esta razón que hemos profundizado la teoría de éste algoritmo en el capítulo 3, desde sus orígenes en la Mecánica Estadística hasta su forma Actual.