

# Capítulo 1

## Búsqueda

La solución de problemas está asociado a la inteligencia.

El proceso “normal” de solución de problemas involucra:

- identificación y definición del problema
- identificación del criterio de evaluación
- generación de alternativas
- búsqueda de una solución y evaluación
- selección de opción y recomendación
- implementación

En IA la solución es principalmente búsqueda y evaluación.

La búsqueda es necesaria en la solución de problemas y normalmente involucra introducir *heurísticas*.

Las heurísticas son criterios, métodos, o principios para decidir cuál de varias alternativas de acción promete ser la más efectiva para cumplir con una meta.

Representan un compromiso entre: (i) simplicidad y (ii) poder discriminatorio entre opciones buenas y malas.

Las heurísticas no garantizan la acción más efectiva, pero muchas veces lo hacen.

En problemas complejos, las heurísticas juegan un papel fundamental para reducir el número de evaluaciones y para obtener soluciones dentro de restricciones de tiempo razonables.

## 1.1 Ejemplos e Ideas

**El problema de las 8 reinas** : El problema consiste en colocar 8 reinas en un tablero de ajedrez sin que se ataquen.

Una posibilidad es intentar lograr una solución de manera *incremental*, acercándose a la meta poco a poco siguiendo una serie de *decisiones locales* basadas en la información obtenida durante el proceso.

Hay que asegurarse que la secuencia de transformaciones sea *sistemática*, para (i) no generar configuraciones repetidas y (ii) no excluir configuraciones deseables.

Una forma de sistematizar la búsqueda es *construyendo* más que transformando configuraciones (aunque ver GSAT).

Por ejemplo el hecho de que deba de haber sólo una reina por columna nos reduce el número de alternativas a menos de 8 en cada renglón.

*El hecho de que podamos sistematizar la búsqueda de esta forma es que no nos podemos recuperar de violaciones a restricciones mediante operaciones futuras.* Osea que una vez que violamos una restricción no podemos seguir añadiendo más reinas para rectificarla.

Posibles candidatos de heurísticas:

1. Preferir colocar reinas que dejen el mayor número de celdas sin atacar. En el ejemplo:  $heu1(A)=8$ ,  $heu1(B)=9$ ,  $heu1(C)=10$ .
2. Ver cuál es el menor número de celdas no atacadas en cada renglón y escoger la que su número menor sea mayor. En el ejemplo:  $heu2(A)=1$ ,  $heu2(B)=1$ ,  $heu2(C)=2$

La heurística 2 permite detectar *camino sin salida*.

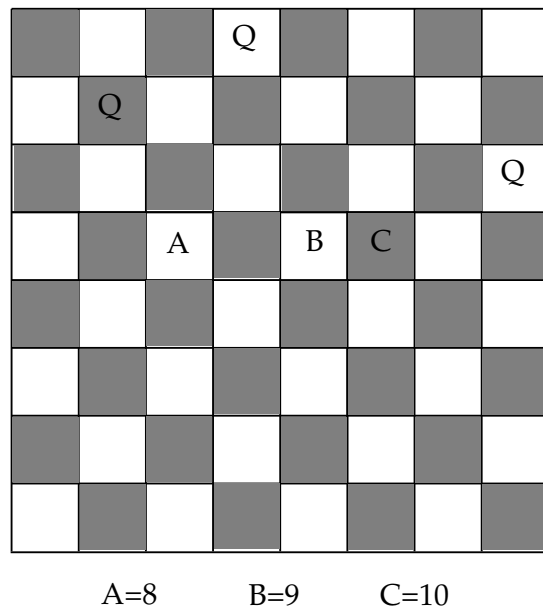


Figura 1.1: El problema de las 8 reinas

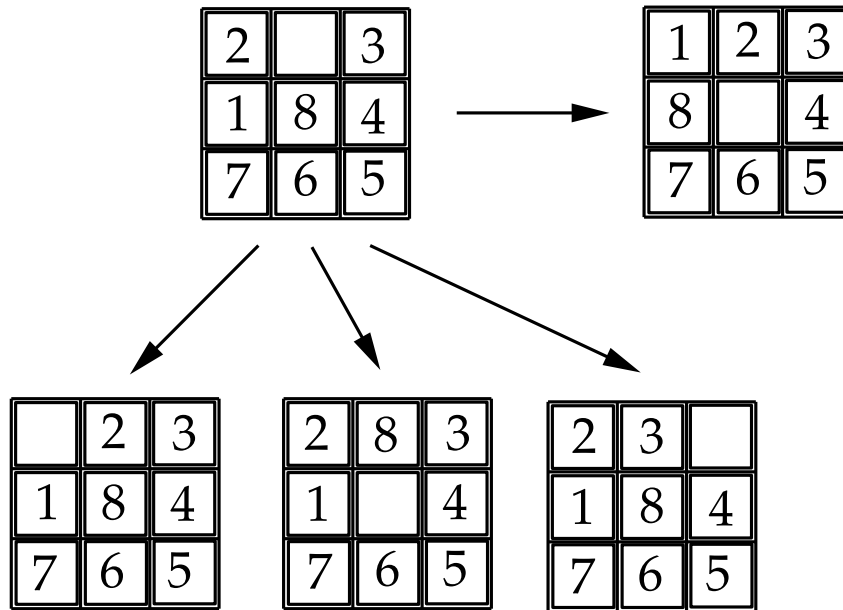


Figura 1.2: El problema del 8 puzzle

**El problema del 8-puzzle** : consiste en pasar de una configuración inicial a una final deslizando cuadros aprovechando el espacio vacío.

El tratar de hacer *búsqueda exhaustiva* es impráctico cuando el número de posibles estados es muy grande, por lo que la decisión de qué hacer depende de una regla o juicio anterior a la búsqueda.

Una posible regla es estimar *qué tan cerca se está de la solución*.

Algunas heurísticas que se han usado para ésto son:

1. Contar el número de cuadros que no corresponden a la meta (sin contar el blanco). En el ejemplo:  $heu1(A)=2$ ,  $heu1(B)=3$ ,  $heu1(C)=4$ .
2. La suma de la distancia Manhattan de los cuadros que no corresponden a su lugar. En el ejemplo:  $heu2(A)=2$ ,  $heu2(B)=4$ ,  $heu2(C)=4$ .
3. Distancia del cuadro blanco al primer cuadro fuera de su lugar. En el ejemplo:  $heu3(A)=1$ ,  $heu3(B)=1$ ,  $heu3(C)=3$ .

4. Distancia entre la posición final del blanco y su posición actual.  
En el ejemplo:  $heu4(A)=2$ ,  $heu4(B)=0$ ,  $heu4(C)=2$ .

**Encontrar la ruta más corta entre dos ciudades** : dado un mapa con varias ciudades encontrar el camino más corto entre un par de ciudades. Con el mapa podemos descartar automáticamente rutas poco prometedoras.

Pero si en lugar del mapa se nos dá información de las distancias entre ciudades no es obvia la preferencia.

Esto es porque en el mapa es posible estimar las distancias Euclidianas. Sin embargo, se pueden calcular las distancias a partir de las coordenadas de las ciudades, por lo que se puede usar esa *información extra* para determinar que acción tomar, en base a una estimación de lo que falta mas la distancia recorrida.

**El problema del agente viajero** : encontrar el circuito más corto entre ciudades.

Es un problema NP-duro, por lo que se requiere de *heurísticas adecuadas que acoten* el problema.

Si tenemos un pedazo de ruta, de las diferentes alternativas que existen en esa ruta la mejor es la que nos estime completar la menor ruta.

Esa estimación puede ser difícil de hacer, pero si es *optimísta* (subestima consistentemente el costo real) entonces el que nos de la mejor estimación es el mejor.

Cuáles pueden ser buenos estimadores sobre los nodos faltantes?

1. El grafo de grado dos más barato ( $O(n^3)$ ).
2. El árbol de expansión mínimo ( $O(n^2)$ ).

Aunque estas aproximaciones no representan ni se asemejan a la solución del problema, son buenos estimadores.

Algo a considerar, es qué tanto afecta la calidad de los estimadores a la complejidad de la búsqueda.

**Detectar la moneda falsa (12 monedas, pesando 3 veces)** : Supongamos que tenemos una moneda falsa (más/menos pesada) en un grupo de 12, una balanza y tres intentos para detectarla.

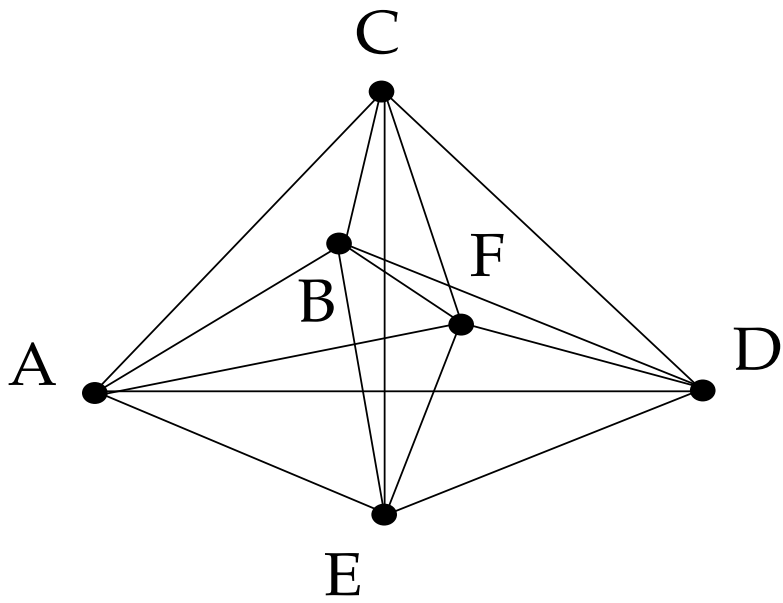


Figura 1.3: El problema del agente viajero

Se requiere de una *estrategia*, o sea una descripción de qué hacer primero y en base al resultado obtenido qué hacer después y así sucesivamente.

Se tienen 3 posibles resultados: a) queda balanceado b) se inclina a la derecha y c) se inclina a la izquierda.

Cada acción requiere hacer una estimación de qué hacer para las 3 posibles situaciones. Por lo que el mérito de una acción depende de la combinación de los méritos de las 3 posibles acciones (*regla de combinación*).

Una vez que se decide alguna alternativa, se necesita especificar qué subproblema tiene que ser considerado después.

Todos estos ejemplos ilustran ideas valiosas como búsqueda incremental, sistemática, el detectar a tiempo caminos sin salida, usar subestimaciones para evaluar posibles alternativas, utilizar información extra, definir estrategias ante posibles resultados de acciones y para cada subproblema, etc.

## 1.2 Optimización vs. Satisfacción de Restricciones

El problema del agente viajero y de la ruta mínima entre dos ciudades, requiere de un proceso de *optimización*. Los otros problemas sólo tratan de *satisfacer restricciones*.

La mayoría de los problemas se pueden plantear tanto como de optimización como de satisfacción de restricciones.

La diferencia entre los dos puede ser muy importante. Encontrar una ruta entre ciudades en el problema del agente viajero es trivial, encontrar la ruta más corta es NP.

Una idea fundamental dentro de los métodos heurísticos es que *lo más corto/barato es lo más rápido/mejor*.

Si existe un criterio de aceptación (tolerancia) se habla de un problema de

*semi-optimización*. Cuando se habla de alta probabilidad, se tiene un problema de *optimización aproximada*.

La mayoría de los problemas son de semi-optimización, estableciendo un balance razonable entre calidad de la solución y costo de solución.

Algo un poco más ambicioso es dotar a los algoritmos con parámetros ajustables para cambiar el compromiso entre calidad y costo.

## 1.3 Representación

La representación que se usa para caracterizar un problema es fundamental para su solución.

Los requerimientos necesarios para una representación de un problema de búsqueda son:

1. Una estructura simbólica llamada código o base de datos o agenda que contiene conjuntos de soluciones potenciales
2. Un conjunto de operaciones o reglas que nos modifican los símbolos en la agenda y nos producen conjuntos más refinados de soluciones potenciales
3. Una estrategia de búsqueda o de control que decide qué operación aplicar a la agenda.

La estrategia de control se dice *sistemática* si considera todos los posibles caminos (completa), y no explora un objeto más de una vez (eficiente).

La representación debe de ser capaz de representar los objetos de trabajo (caminos, tableros, estrategias).

Es deseable poder representar también *subconjuntos de objetos* o de soluciones potenciales, ya que nos permite eliminar espacios de búsqueda grandes.

Idealmente también debemos de tener transformaciones entre estos subespacios de búsqueda.



Una de las más usadas es la de dividir el espacio en *subconjunto de soluciones potenciales*.

Los candidatos que mostramos en el ejemplo de las 8 reinas, nos dividen el espacio en 3 espacios mutuamente exclusivos (las demás alternativas no se tienen que considerar porque no nos llevan a ninguna solución).

Un tablero parcialmente ocupado nos representa un subconjunto de soluciones potenciales. El añadir una nueva reina, nos refina el espacio.

Es esta habilidad de detectar y eliminar o cortar grandes subconjuntos de búsqueda que hacen que el problema sea fácil de resolver.

Si no, considere resolver el problema poniendo las reinas en una configuración aleatoria y perturbarla hasta dar con la solución (aunque si se ha hecho con buenos resultados para tableros muy grandes). En esta representación sí se tienen representados los objetos individuales, pero no sus subconjuntos. Por lo que podemos eliminar elementos, pero no subconjuntos.

En el 8-puzzle no se ve claro como eliminar subconjuntos, ya que en principio uno puede regresarse al camino anterior. Claro que si ponemos restricciones en la longitud máxima permitida ( $L$ ) del camino, podemos eliminar todos aquellos que sean mayores a  $L$ .

El estimar en forma temprana caminos sin salida a veces requiere de heurísticas que explotan información que no está explícita en la formulación del problema (v.g., 8-reinas).

El uso de heurísticas imponen requerimientos adicionales en el formato de representación. La representación debe de identificar adecuadamente los subconjuntos y facilitar el cálculo de la heurística.

Por ejemplo, podemos jugar con un contrincante a seleccionar dígitos en forma alternada, tratando de encontrar tres dígitos que sumen 15. Si se usa un cuadro mágico y se juega gato el juego es trivial.

En muchos casos es deseable incluir dentro de la representación información adicional que nos defina el subproblema que falta por resolver. El código que especifica esta información adicional se llama un *estado*.

4	9	2
3	5	7
8	1	6

Figura 1.4: El cuadro mágico

El conjunto de todos los subproblemas obtenidos de ejecutar alguna secuencia de operadores de refinamiento se llama un *espacio de estados*.

La importancia de representar los subproblemas que faltan es que muchas veces éstos: (i) se repiten y podemos aprovechar las experiencias obtenidas o (ii) podemos juzgar que opción es mejor de dos si las dos tienen el mismo espacio faltante y una de ellas es superior a la otra.

El problema de las monedas introduce un nuevo elemento, la *estrategia*.

La estrategia nos debe de decir qué acción tomar en respuesta a cualquier evento externo, que puede ser el resultado de una prueba, la movida de un oponente, el resultado de un cálculo complicado.

Ahora lo que buscamos, en lugar de caminos, son árboles.

En este caso se tiene una representación en donde los subproblemas candidatos se pueden resolver independientemente de los otros.

Esto sugiere que los subproblemas sean considerados como nodos individuales, *resueltos independientemente*, *guardarlos* y *usados* en el resto de la estrategia.

Un código adecuado es el de gráficos AND/OR.

Los gráficos tienen dos tipos de arcos:

1. Ligas que representan caminos alternativos (OR).
2. Ligas que conectan un problema con sus subproblemas asociados.

En general, las ligas AND y OR pueden apuntar simultaneamente al mismo nodo. Por otro lado, un nodo puede tener salidas AND y OR. En este caso es conveniente crear un nodo AND *artificial*.

Como (casi) siempre, el representar el árbol de búsqueda completo es imposible en memoria. Una buena estrategia trata de encontrar una solución explorando sólo una pequeña porción del grafo.

Algo común es etiquetar los nodos como resueltos (no resueltos). Las reglas de etiquetación son las siguientes:

1. Si es nodo terminal etiquétalo
2. Si es OR no terminal, etiquétalo si existe al menos una etiqueta
3. Si es AND no terminal etiquétalo si todos están etiquetados

## 1.4 Representación Basada en Estados vs. Reducción de Problemas

En una representación basada en estados el proceso de solución trata de encontrar una secuencia de operaciones que transformen al estado inicial en uno final.

Reducción de problemas implica que los subconjuntos candidatos se pueden ver como una conjunción de varios subproblemas a resolver.

En general, si la solución puede especificarse como un camino o un estado, la representación basada en estados es más adecuada.

Por otro lado si la solución es más conveniente representarla como un árbol, entonces una representación de reducción de problemas o grafo es más adecuada.

Las estrategias se representan mejor si los arcos AND representan cambios ocasionados por situaciones externas y las OR son formas alternativas de reaccionar ante esos cambios.

Otro tipo de problemas susceptibles de ser representados por gráfos AND/OR son los que su solución es una *secuencia parcialmente ordenada* de acciones (v.g., integración simbólica, prueba de teoremas, las torres de Hanoi).

En particular el problema de las torres de Hanoi, puede ser fácilmente representado como un espacio de estados. Sin embargo, las metas poseen una jerarquía lineal. Una vez resuelto el pasar varios de los discos más grandes, los demás se pueden resolver sin alterar las metas obtenidas.

La clave está en empezar con la meta más crítica (mover el disco más grande a su lugar). Lo cual implica moverlo y mover todos los discos arriba de él a otro poste.

Las metas serían:

1. Mover los  $n - 1$  discos al poste intermedio
2. Mover el disco  $n$  al poste meta
3. Mover los  $n - 1$  discos al poste meta

En principio la búsqueda de las soluciones puede hacerse de manera independiente. Este *plan abstracto* puede seguirse aplicando a los siguientes niveles hasta llegar a *acciones primitivas*.

Algo que tenemos que considerar es cómo encontrar una buena heurística? Cómo usarla para que sea más efectiva? Cómo evaluar sus méritos? Parte de los objetivos del curso es ayudar a contestar estas preguntas.