

Tema No. 2: Métodos de solución de problemas.

Actividad No. 23

Tipo de clase: Conferencia.

Título: Métodos locales de búsqueda.

Sumario:

- ✚ Hill Climbing.
- ✚ Simulated Annealing.
- ✚ Local Beam Search.

Objetivo:

- ✚ Caracterizar los métodos de búsqueda local.

Introducción:

Los métodos de búsqueda vistos hasta el momento son diseñados para explorar el espacio de búsqueda sistemáticamente. Esta sistematicidad es lograda almacenando en memoria uno o más caminos y teniendo en cuenta qué alternativas han sido exploradas en cada momento a lo largo del camino y cuáles no. Cuando se llega a un objetivo, el camino hasta ese objetivo también forma parte de la solución del problema.

En muchos problemas es irrelevante el camino para llegar al objetivo. Por ejemplo, en el problema de las 8 reinas, lo que interesa es llegar a la configuración final de las piezas no amenazadas entre sí, no el orden en que fueron colocadas cada una de ellas sobre el tablero. Esta clase de problemas contempla muchas aplicaciones importantes como el diseño de circuitos integrados, programación automática, problemas de horarios, optimización en redes de telecomunicaciones, entre otros.

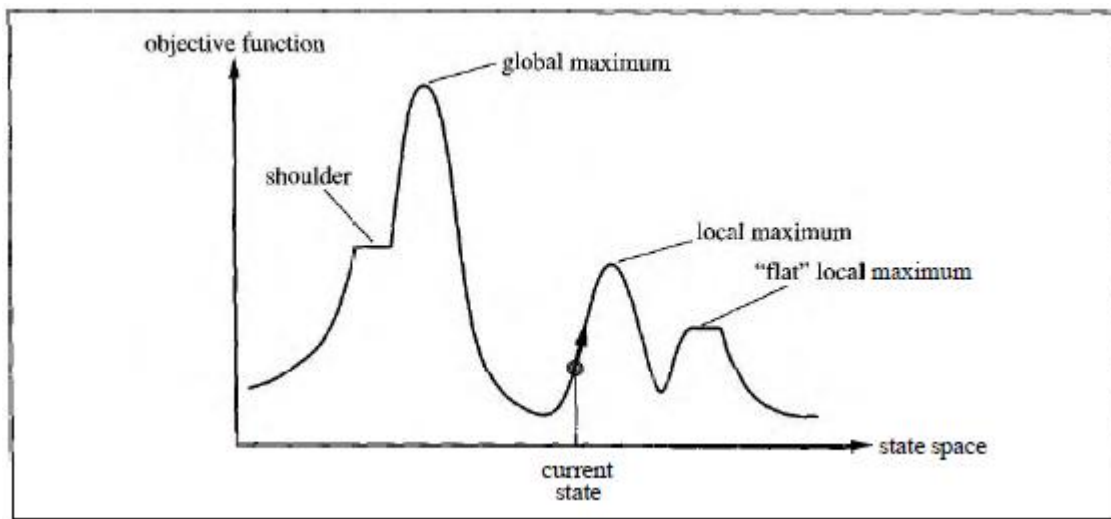
Si el camino hasta el objetivo no importa, estamos en presencia de una clase de algoritmos diferente. Los algoritmos de búsqueda locales operan usando un único estado actual (en vez de múltiples caminos), y generalmente se mueven solo a los vecinos de ese estado. Los caminos seguidos por el algoritmo no son almacenados. Aunque los algoritmos de búsqueda local no son sistemáticos, presentan dos ventajas fundamentales:

- Necesitan de muy poco almacenamiento para su ejecución
- Frecuentemente encuentran soluciones razonables en amplios o infinitos (continuos) espacios de estados para los cuales los algoritmos sistemáticos son inapropiados.

Los algoritmos de búsqueda local son muy útiles en problemas de optimización, en los cuales el objetivo es encontrar el mejor estado de acuerdo a una función objetivo. Muchos problemas de optimización no encajan con el modelo de búsqueda "estándar" definido anteriormente. Por ejemplo, la naturaleza brinda una función objetivo (ajuste reproductivo), que identifica la evolución darwiniana como un enfoque favorable de optimización, sin contar con las definiciones ya vistas de costo del camino y criterio objetivo.

Búsqueda Local.

Para entender la búsqueda local, consideremos el espacio de estados de la siguiente figura:



Como se puede apreciar, el gráfico es de espacio de estados contra función objetivo. Los algoritmos de búsqueda local exploran dicho espacio de estados. Un algoritmo de búsqueda local completo siempre encuentra un objetivo si existe; un algoritmo óptimo siempre encuentra un extremo global.

Hill Climbing

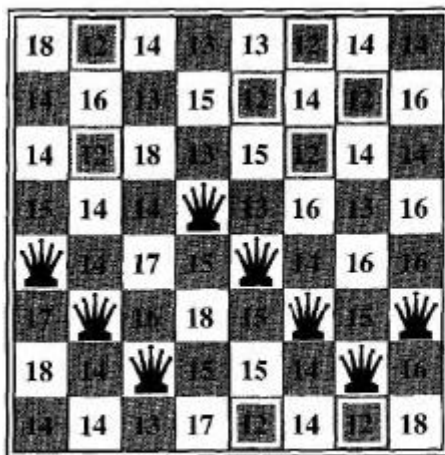
El algoritmo Hill Climbing (Escalador de Colinas) es simplemente un ciclo que se mueve en la dirección de máximo ascenso o descenso, en dependencia de si se esté maximizando o minimizando la función objetivo. El algoritmo no mantiene un árbol de búsqueda, de tal forma que el nodo actual solo necesita almacenar el estado y su evaluación de la función objetivo. A continuación veamos un pseudocódigo del Hill Climbing:

1. Evaluar el estado inicial. Si es también el estado objetivo, devolverlo y terminar. En caso contrario continuar con el estado inicial como el estado actual

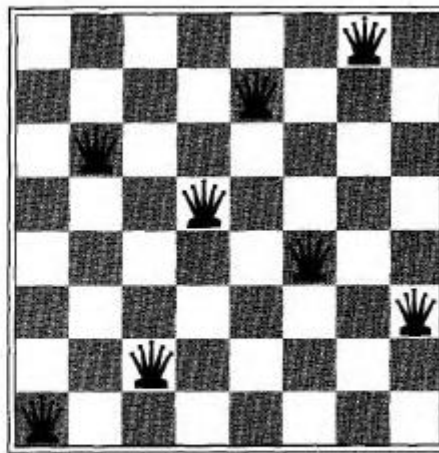
2. Repetir hasta que se encuentre una solución o hasta que una iteración completa no produzca un cambio en el estado actual.

- a) Sea Sucesor (SUCC) un estado tal que algún posible sucesor del estado actual sea mejor que este SUCC.
- b) Para cada operador aplicado al estado actual hacer lo siguiente:
 - i. Aplicar el operador y generar un nuevo estado.
 - ii. Evaluar el nuevo estado. Si es un estado objetivo, devolverlo y terminar.
 - Si no, compararlo con SUCC. Si es mejor, asignar a SUCC este nuevo estado. Si no es mejor, dejar SUCC como está.
- c) Si SUCC es mejor que el estado actual, hacer que el estado actual sea SUCC.

Ilustremos el Hill Climbing con el problema de las 8 reinas. Los algoritmos de búsqueda local usan una formulación completa de estado, en este caso pongamos que sea una ubicación de las 8 reinas sobre el tablero, una por cada columna. La función sucesor devuelve todos los posibles estados que se obtienen de desplazar una única dama a cualquier otra casilla de la misma columna (o sea, cada estado tiene $8 \times 7 = 56$ sucesores). Definamos como heurística sobre el estado la cantidad de pares de reinas que se atacan entre sí, directa o indirectamente. El mínimo de esta función es cero y se obtiene con la solución del problema. En la siguiente figura, el a) muestra un estado con $h=17$ y todos los posibles sucesores que se pueden obtener a partir de él con sus respectivas evaluaciones de h (véase que los mejores sucesores son de evaluación $h = 12$). En el b) se muestra un mínimo local de dicho problema, su evaluación es $h=1$ pero cada sucesor es mayor.



(a)



(b)

Si varios sucesores comparten el mismo mínimo de la función heurística, el Hill Climbing escoge aleatoriamente entre dichos sucesores el próximo estado. El Hill Climbing es frecuentemente llamado Greedy Local Search porque se queda con el mejor vecino sin ver más hacia adelante. Frecuentemente progresa muy rápido hacia una solución porque le es muy fácil mejorar un mal estado. Sin embargo, el algoritmo pudiera detenerse por cualquiera de los siguientes motivos:

1. Un máximo local: un estado que es mejor que todos sus vecinos, pero no es mejor que otros estados de otros lugares. En un máximo local, todos los movimientos producen estados peores. Los máximos locales son particularmente frustrantes porque frecuentemente aparecen en las cercanías de una solución. En este caso se denominan estribaciones.
2. Una meseta: un área plana del espacio de búsqueda en la que un conjunto de estados vecinos posee el mismo valor. En una meseta no es sucesor determinar la mejor dirección a la que moverse, haciendo comparaciones locales.
3. Una cresta: un tipo especial de máximo local. Es un área del espacio de búsqueda más alta que las áreas circundantes y que además posee en ella misma una inclinación (la cual se podría escalar). Pero la orientación de esta región alta, comparada con el conjunto de movimientos disponibles y direcciones en las que moverse, hace que sea imposible atravesar la cresta mediante movimientos simples.

Existen algunas formas de evitar estos problemas, si bien estos métodos no dan garantía.

- Volver atrás hacia algún nodo anterior e intentar seguir un camino diferente. Es especialmente razonable si el nodo posee otra dirección que de la impresión de ser tan prometedora, o casi tan prometedora, como la que se eligió. Para implementar esta estrategia, se debe mantener una lista de caminos que casi se han seguido y volver a uno de ellos, si el camino que se ha seguido da la impresión de ser un callejón sin salida. Este método es especialmente adecuado para superar máximos locales.

- Realizar un gran salto en alguna dirección para intentar buscar en una nueva parte del espacio de búsqueda. Este método está especialmente indicado para superar mesetas. Si la única regla aplicable describe pequeños pasos, aplicarla varias veces en la misma dirección.

Incluso con estas medidas de primeros auxilios, el escalador de colinas no es siempre muy eficaz. Especialmente es inadecuado para problemas en los que el valor de la función heurística cambia bruscamente al alejarse de una solución. Esto ocurre frecuentemente cuando aparece algún tipo de efecto umbral.

El escalador de colinas es un método local, lo que significa que decide cuál va a ser el siguiente movimiento atendiendo únicamente a las consecuencias inmediatas que va a tener esa elección, en lugar de explorar exhaustivamente todas las consecuencias.

Entre las disímiles variantes de escalador de colinas que existen se puede destacar el estocástico, el cual escoge aleatoriamente el siguiente estado de entre los sucesores que mejoran el estado actual, lo cual hace que converja más lentamente que el método principal pero en determinados espacios de estados encuentra mejores soluciones. Igual que el clásico hill-climbing, se detiene cuando ninguno de los sucesores mejora el estado actual.

Simulated Annealing.

El recocido simulado (simulated annealing) es una variación del método Hill Climbing el cual al inicio del proceso puede realizar algunos movimientos de descenso. Fue propuesto por S. Kirkpatrick, C. D. Gelatt y M. P. Vechi en 1982 y publicada en 1983. La idea es hacer una exploración que permita que la solución final sea relativamente insensible al estado inicial. Esto también disminuye la posibilidad de caer en un máximo local, una meseta o una cresta.

Este método está inspirado en el proceso físico de recocido, en el cual las sustancias físicas tales como los metales son fundidos (elevando su temperatura) y luego se enfrían gradualmente hasta alcanzar el estado sólido. El objetivo de este proceso es producir un estado final de mínima energía. En este proceso los cambios son graduales usualmente hacia abajo, pero pueden ocurrir movimientos de ascensos de la energía con probabilidad:

$$P=e^{-dE/kT}$$

donde dE es el cambio positivo en el nivel de energía, T es la temperatura y k la constante de Boltzmann. La probabilidad de que ocurra un movimiento de ascenso de la energía decrece a

medida que la temperatura decrece. Por eso tales movimientos son más posibles al inicio del proceso cuando la temperatura es alta y menos probable al final.

Una característica de este proceso es que en cualquier momento pueden ocurrir movimientos de ascenso, aunque con menos probabilidad y magnitud al final del proceso hasta que finalmente el proceso converge a un mínimo.

El proceso es gobernado por el parámetro T. El plan de cómo variar T se denomina esquema de recocido. Un enfriamiento demasiado rápido puede llevar a regiones estables de alta energía (mínimos locales en lugar de globales). Un enfriamiento lento propicia alcanzar con más seguridad una estructura uniforme (mínimo global), pero si es demasiado lento se pierde tiempo. El esquema de recorrido óptimo para cada problema particular tiene que ser descubierto empíricamente.

Se puede establecer una analogía de este proceso físico con un proceso de búsqueda Hill Climbing. En él dE representa el cambio en la función heurística, y el término kT se integra sólo en T. La diferencia principal de este algoritmo con respecto al hill- climbing es que se admite con cierta probabilidad movimiento a estados peores que el actual.

Algoritmo:

P1. Evaluar el estado inicial. Si es objetivo SALIR con ÉXITO, sino hacerlo estado actual.

P2. Asignar a BEST-SO-FAR el estado actual.

P3. Inicializar T según el esquema de recocido.

P4. Iterar hasta encontrar una solución o hasta que no haya nuevos operadores aplicables al estado actual.

a) Seleccionar un operador no usado y aplicarlo al estado actual para producir un nuevo estado.

b) Evaluar el nuevo estado. Calcular.

$$dE = (\text{valor del estado actual}) - (\text{valor del nuevo estado}).$$

i. Si el nuevo estado es objetivo, SALIR con ÉXITO.

ii. Si no es objetivo pero es mejor que el actual, hacerlo actual. Asignarlo a BEST-SO-FAR.

iii. Si no es mejor entonces calcular

$$P' = e^{-dE/T}$$

Generar un número aleatorio a en el rango [0,1].

Si $a < P'$ entonces hacer el movimiento.

c) Revisar T de acuerdo al esquema de recorrido.

P5. Retornar BEST-SO-FAR como la respuesta.

El esquema de recorrido tiene tres componentes. La primera es el valor inicial para T. El segundo el criterio para decidir cuándo reducir la T. El tercero es la cantidad para la cual T debe ser reducida cada vez que sea cambiada.

El recorrido simulado también ha servido para construir modelos de redes neuronales artificiales, como la Máquina de Boltzmann.

Local Beam Search

La búsqueda en haz local (Local Beam Search en inglés) mantiene en memoria k estados en lugar de uno solo. Comienza con k estados generados aleatoriamente. En cada paso, todos los sucesores de los k estados son generados. Si alguno es el objetivo, el algoritmo se detiene. Si no, selecciona los k mejores sucesores de toda la lista de estados obtenidos y se repite el proceso. En su forma más simple, la búsqueda en haz local puede sufrir de una falta de diversidad entre los k estados, o sea, rápidamente pueden concentrarse en una pequeña región del espacio de estados, convirtiéndose la búsqueda en una versión más costosa del escalador de colinas. Una variante denominada búsqueda en haz estocástica (stochastic beam search), análoga al escalador de colinas estocástico, ayuda a aliviar este problema. En lugar de escoger los k mejores estados de todos los candidatos, este método escoge k sucesores al azar, con la probabilidad de escoger entre dichos k sucesores un estado que no mejore su heurística.

Conclusiones

A partir de preguntas dirigidas a los estudiantes, concluir la gran importancia que revisten los métodos de búsqueda local en la resolución de problemas y resumir las características de los tres métodos vistos en clase.

Bibliografía

- [1] Patrick Henry Winston, Inteligencia Artificial, 3ra edición, 1992.
- [2] Elaine Rich & Kevin Knight, Inteligencia Artificial, 2da edición, 1994.
- [3] Russell, S, Norvig, P. AI, A Modern Approach, 3era. edición 2010.