



INSTITUTO TECNOLÓGICO de nuevo laredo

Ingeniería en Sistemas Computacionales

Inteligencia Artificial

Ing. Bruno López Takeyas

Algoritmo Hill Climbing

Alumnos

Ylliana Samantha Anderson Benavides	01100161
Pablo Saúl Hernández Ribota	01100230
Andrea Ramírez Saavedra	01100288
Karla Rocío Reyes Chavez	01100290
Efraín Velásquez Flores	01100320

Octubre de 2005

ALGORITMO HILL CLIMBING (ASCENSO DE COLINA)

Similares a los algoritmos genéticos, aunque más sistemáticos y menos aleatorios.

Hill climbing es una técnica que permite solo ir mejorando la solución por que aplica un mecanismo de restar o multistart tratando de mejorar la solución.

En cada iteración, un nuevo punto es seleccionado de la vecindad del punto actual.

Un algoritmo de ascenso a colina comienza con una solución al problema a mano, normalmente elegida al azar.

Es una técnica que permite solo ir mejorando una solución. Luego, la cadena se muta, y si la mutación proporciona una solución con mayor aptitud que la solución anterior, se conserva la nueva solución; en caso contrario, se conserva la solución actual.

Si el nuevo punto es mejor, se transforma en el punto actual, sino otro punto vecino es seleccionado y evaluado

El método termina cuando no hay mejoras, o cuando alcanza un número predefinido de iteraciones.

Luego el algoritmo se repite hasta que no se pueda encontrar una mutación, cuando no hay mejoras o cuando se alcanza un número predefinido de iteraciones y esta solución se devuelve como resultado.

El algoritmo de ascenso de colina es lo que se conoce como algoritmo voraz.

Lo que significa que siempre hace la mejor elección disponible en cada paso, con la esperanza de que de esta manera se puede obtener el mejor resultado global. En contraste, los métodos como los algoritmos genéticos y el recocido simulado, discutido abajo, no son voraces; a veces, estos métodos hacen elecciones menos óptimas al principio con la esperanza de que conducirán hacia una solución mejor más adelante.

Los algoritmos de ascenso a colina son típicamente locales

Ya que deciden que hacer, mirando únicamente a las consecuencias inmediatas

Este algoritmo toma muchas formas diferentes.

Es según el número de dimensiones del problema solución, el valor de incremento y en la dirección en la cual se tiene que dar.

Hill-climbing es una estrategia basada en optimización local.

Sigue la dirección de ascenso/descenso más empinada a partir de su posición y requiere muy poco costo computacional.

Se llama también una estrategia irrevocable porque no permite regresarnos a otra alternativa.

Es útil si se tiene una función heurística muy buena o cuando los operadores de transición entre estados tienen cierta independencia (conmutativa), que implica que la operación de un operador no altera la futura aplicación de otro.

Se puede realizar la búsqueda de dos maneras diferentes:

Escala Simple.-

- Se dirige a un estado mejor que el actual
- Función heurística de proximidad
- No se almacenan los estados anteriores
- Es un método local, sus movimientos están determinados por ser mejor al anterior.

Escala por máxima pendiente

- Busca no solamente un estado mejor que el actual, si no el mejor de todos los estados posible, es una máxima pendiente.

Algunas ocasiones no se encuentran la solución y se pueden presentar 3 tipos de problemas:

- **Un máximo local:** Es un estado mejor que sus vecinos, pero aun así no es el mejor que otros que están algo más alejados.

La solución cuando se presenta este tipo de problemas es regresar a un estado anterior y explorar en una dirección diferente.

- **Una meseta:** Es un espacio de búsqueda en el que todo un conjunto de estados vecinos tienen el mismo valor.

La solución es dar un salto grande en alguna dirección (al azar) y tratar de encontrar una nueva sección de estados.

- **Un risco:** Es parecido al máximo local, imposible de atravesar con movimientos simples.

La solución es aplicar dos o más reglas, antes de realizar una prueba del nuevo estado. Se tiene que mover en varias direcciones a la vez.

El Hill Climbing, tiene las siguientes características:

Informado.– Utiliza información del estado por elegir un nodo u otro

No exhaustivo.– No explora todo el espacio de estados. Solo encuentra una solución.

Encuentra buenas soluciones.– Encuentra muy buenas soluciones, aunque muchas veces no es la mejor, puesto que no explora todos los estados.

Es eficiente.– Debido a que evita la exploración de una parte del espacio de estados.

Ventaja:

Reduce el numero de nodos a analizar

Desventaja:

Puede ser que encuentre una solución, pero no es la más óptima.

CODIGO ALGORITMO

Escalada simple (hill climbing)

evaluar el estado INICIAL

si es un estado objetivo entonces devolverlo y parar

si no ACTUAL := INICIAL

mientras haya operadores aplicables a ACTUAL y no se haya encontrado solución

 seleccionar un operador no aplicado todavía a ACTUAL

 aplicar operador y generar NUEVOESTADO

 evaluar NUEVOESTADO

 si es un estado objetivo entonces devolverlo y parar

 si no

 si NUEVOESTADO es mejor que ACTUAL

 entonces ACTUAL := NUEVOESTADO

 fin si

 fin si

fin mientras

fin si

Escalada por Máxima Pendiente

EstadoActual= Estadoinicial

final = falso

Mientras ¬final

 Hijos= generarsucesores(EstadoActual)

 si ¬vacio (Hijos) entonces

 Hijos=Ordenarhijos(EstadoActual)

 si son iguales entonces

 si hijos (1) > EstadoActual entonces

 EstadoActual=Hijo(1)

 si no

 EstadoActual=Random()

 fin si

 si no

 Hijos=Eliminar_peoreshijos(Hijos,Estado Actual).

 EstadoActual=Seleccionar_mejorhijo(Hijos)

 fin si

si no

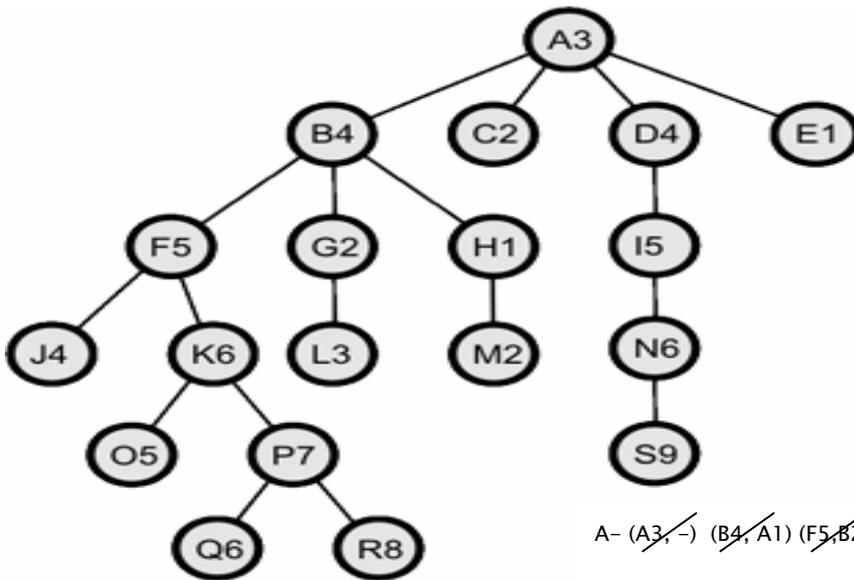
 Final=Verdadero

fin si

fin mientras

Solucion=EstadoActual

EJERCICIOS ARBOL

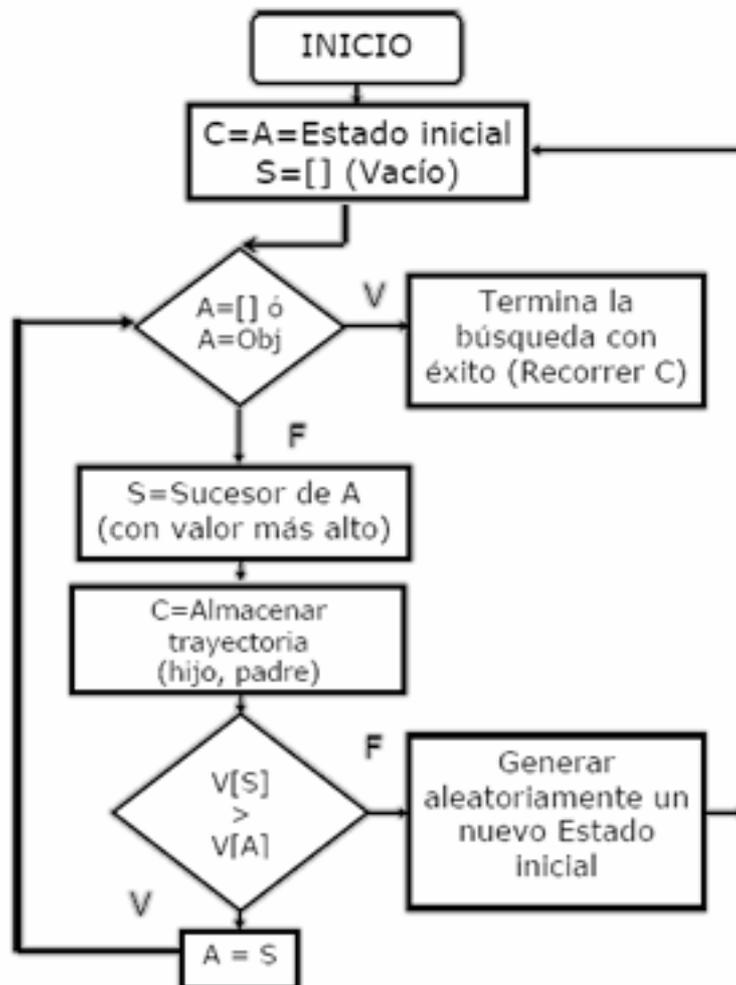


A- (~~A3, -~~) (~~B4, A1~~) (~~F5, B2~~) (~~K6, F3~~) (~~P7, K4~~) (R8, P5)

C- (A3, -), (B4, A1), (F5, B2) (K6, F3) (P7, K4) (R8, P5)

S- (~~B4, A1~~) (~~F5, B2~~) (~~K6, F3~~) (~~P7, K4~~) (R8, P5)

DIAGRAMA GENERAL DEL HILL CLIMBING



APLICACIONES

El Generador de Rutas

La distribución (o topología) de las áreas del edificio y sus interconexiones se pueden representar fácilmente con un grafo no dirigido. En este grafo cada nodo representa un área y cada arco representa una vía que conecta a dos áreas. Tomando esta representación, el problema de determinar una ruta segura para evacuar a la gente del edificio, se reduce al conocido problema de recorrer un grafo.

Existen varias técnicas de búsqueda en un grafo. Estas difieren entre sí por la manera en que recorren el grafo buscando una buena solución al problema. Las técnicas de búsqueda pueden ser de inteligencia artificial, de programación entera o métodos de investigación de operaciones. Algunos de ellos, los métodos de investigación de operaciones y de programación entera, pueden encontrar la mejor ruta, en cambio las técnicas de inteligencia artificial encuentran una buena ruta, sin ser probablemente la mejor de todas las posibles.

Sin embargo, la cantidad de información que se necesitaría procesar en el caso de un edificio es enorme. Es tan significativa la cantidad de información, que es más factible utilizar técnicas de inteligencia artificial, ya que es probablemente más rápido, comparado con una búsqueda exhaustiva entre todas las posibilidades para encontrar la mejor solución.

Las técnicas de búsqueda de inteligencia artificial se dividen en Depth-first y Breadth-first (profundidad primero y anchura primero respectivamente). El método conocido como Hill-Climbing, que es de tipo Depth-first, es el que se escogió para el sistema de toma de decisiones en caso de emergencia para encontrar las rutas de evacuación. No hay una razón especial por la que se optó por éste método. Simplemente, analizando el algoritmo presentado en [STER86] para el método Hill-Climbing, éste no presentó problema alguno para nuestro caso.

Demos

<http://www.dma.fi.upm.es/mabellanas/tfcs/visibilidad/ejecuta1.html>

<http://www.coolbuddy.com/games/8queens/default.htm> (juego)

<http://www.lacerta.be/hill.php> (dar click en launch application)

<http://www-poleia.lip6.fr/~drogoul/projects/eco/npuzzle.html>

<http://www.dc.fi.udc.es/lidia/mariano/demos/Tools%20for%20Learning/hill.html>

BIBLIOGRAFIA

<http://html.rincondelvago.com/algoritmos-geneticos.html>

[http://www.fdi.ucm.es/profesor/evah/IAIC/transparencias/Tema2\(a4\).pdf](http://www.fdi.ucm.es/profesor/evah/IAIC/transparencias/Tema2(a4).pdf)

www.itnuevolaredo.edu.mx\takeyas