

PRÁCTICA 1 - BÚSQUEDAS

Uxue Ayechu Abendaño
INTELIGENCIA COMPUTACIONAL

Enfriamiento simulado

Para la resolución de esta práctica he escogido la búsqueda basada en trayectorias simples conocida como enfriamiento simulado, cuyo término procede del ámbito de la Termodinámica.

Movimiento de escape

El algoritmo realiza una búsqueda escogiendo el mejor vecino y permitiendo en algunos casos escoger un vecino que no sea el mejor, para evitar óptimos locales. Lo que se denomina **movimientos de escape**. En la teoría aparecen permitidos al principio y restringidos al final, es decir, diversificar al principio e intensificar al final. Sin embargo, yo he realizado una implementación que hace lo contrario, pues he obtenido con ello mejores resultados. Además, ambas filosofías están permitidas.

Intensificar al principio y diversificar al final, aporta avanzar rápido al principio en caso de que empleemos inicializaciones lejanas. En verdad, si realizamos una segunda iteración sobre una solución encontrada, sería como diversificar al principio e intensificar al final, ya que seguimos investigando en profundidad sobre una solución ya encontrada.

Técnica de enfriamiento

En cuanto al mecanismo de **enfriamiento** de la temperatura, inicialmente realice pruebas reduciendo en cada iteración un grado. Pero con esto, realizaba en algunos casos muchas iteraciones innecesarias. ¿Y porque no reducir la temperatura inicial (número de iteraciones)? Porque como he dicho, esto solo sucedía en algunos casos, por ejemplo, si el umbral inicial se encuentra muy lejos de la solución y reducimos la temperatura, es probable que esto corte la búsqueda sin llegar al mejor umbral. De modo que pensé otra forma de hacerlo, siempre inicializo la temperatura a 40 grados y la voy reduciendo de uno en uno, salvo en los casos en que durante varias iteraciones no cambiemos la mejor solución encontrada, en este caso la temperatura se reduce más grados, para avanzar más rápido la búsqueda.

Este planteamiento tiene ventajas e inconvenientes, pero para las pruebas realizadas he notado que los tiempos de ejecución mejoran considerablemente y los casos en los que cortamos la búsqueda antes de encontrar mejores soluciones son escasos. Para escoger esta técnica dependeremos de muchos factores, como si para nuestro problema es más importante encontrar la mejor solución o hacerlo de forma rápida. En este caso para este proyecto, he insistido más en la velocidad, para poder realizar más pruebas, aunque no obtengamos el mejor umbral, ya que visualmente las diferencias entre umbrales cercanos apenas se notan.

Inicialización de la búsqueda

Normalmente el inicio de estas búsquedas es aleatorio, ya que desconocemos en absoluto el umbral solución. Aunque tal y como está implementado, también permite emplear como umbral inicial el mejor umbral obtenido en búsquedas anteriores.

Como es obvio, el tiempo de ejecución debería ser acorde a la cercanía o no a la solución del umbral inicial que escojamos. Como ya se ha explicado se han establecido medidas, para conseguirlo y que se reduzca el tiempo de ejecución considerablemente si la solución inicial es bastante buena, en lugar de realizar siempre el mismo número de iteraciones.

Generación de vecinos

En este caso, he pensado para la generación de vecinos me he basado en mi pensamiento lógico. Cuando implementé únicamente las varianzas y probaba a encontrar un umbral cuya varianza fuera mejor que la que tenía, modificaba el elemento del umbral correspondiente a un color y volvía a probar, si mejoraba probaba con ese color a cambiar los demás y así sucesivamente, ya que cuando cambiaba varios colores a la vez, el cálculo de la varianza se dispersaba considerablemente y era como volver a empezar. Pues bien, mi búsqueda realiza algo similar y a la hora de generar nuevos vecinos cambia únicamente el píxel correspondiente a un color.

En definitiva, lo que hacemos es generar tres vecinos y cada uno de ellos cambia únicamente un píxel. Dicho cambio puede ser sumar o restar una cantidad de píxeles, que dependerá de la temperatura. La mitad de las veces incrementaremos y la otra mitad decreceremos, de forma totalmente aleatoria. En cuanto al número de píxeles, se incrementarán en un máximo inicial de 10 unidades y disminuirán conforme avanza la búsqueda, entendiendo que estamos más cerca de encontrar el umbral solución para no 'pasárnoslo'. A pesar de esto, si nos lo saltamos no supondrá un grave problema ya que podemos volver a él, gracias a que en algunos vecinos sumamos píxeles y en otros restamos.

Escoger vecinos

Por último, para escoger cual será el próximo vecino a analizar, al inicio cogeremos el mejor vecino y conforme avance la búsqueda, permitiremos coger otros vecinos que no sean los mejores. La probabilidad de coger el mejor vecino se obtiene dividiendo la temperatura entre la temperatura inicial.

En verdad, si unimos la forma en que escogemos los vecinos con la forma en que los creamos es una mezcla de las dos filosofías mencionadas en el apartado 'movimientos de escape'. Pues al principio generamos vecinos más lejanos, pero escogemos los mejores y al final generamos vecinos más próximos, pero permitimos escoger peores.

Pruebas

Conclusiones