Pràctica 1 d'Intel·ligència Artificial

Josep Maria Olivé Fernández, Pol Monroig, Yaiza Cano

20 de Març del 2020

Introducción

La documentación deberá incluir:
La descripción/justificación de la implementación del estado
La descripción/justificación de los operadores que habéis elegido
La descripción/justificación de las estrategias para hallar la solución inicial
La descripción/justificación de las funciones heurísticas
Para cada experimento:
• Condiciones de cada experimento
• Resultados del experimento
• Qué esperabais y qué habéis obtenido
• Comparaciones
• Comentarios adicionales que os parezcan adecuados
Comparación entre los resultados obtenidos con Hill Climbing y Simulated Annealing (no olvidéis explicar cómo habéis ajustado los parámetros para este último algoritmo). Respuestas razonadas a las preguntas del enunciado.

INFORMACIÓN DEL TRABAJO DE INNOVACIÓN

Breve descripción del tema que habéis escogido (no deberíais necesitar más de 3 líneas).
Reparto del trabajo entre los miembros del grupo. Básicamente quién se ha encargado de la búsqueda de la información para desarrollar cada apartado del trabajo.
Lista de referencias que hayáis encontrado, indicando para que apartados del documento son relevantes y la fecha en la que accedisteis a la referencia.
Dificultades que hayáis encontrado a la hora de buscar la información necesaria para el trabajo.

Descripción del Problema

Muchas aplicaciones en internet necesitan tolerancia a fallos y alta disponibilidad. Una solución frecuente para este problema es el tener un sistema de ficheros distribuido a lo largo de varios servidores y tener otro grupo de servidores dedicados a redireccionar las peticiones de ficheros a los servidores que tienen la copia del fichero pedido.

El punto clave de esta solución es cómo el servidor que atiende a las peticiones decide qué servidores van a enviar su copia del fichero a su destinatario. El objetivo de esta práctica es experimentar con algoritmos de búsqueda local para resolver la tarea de, dado un conjunto de peticiones de ficheros, decidir qué servidores van a responder a esa petición.

Definición de tipos

Implementación del estado

El estado que representa las soluciones tiene tiene que almacenar de alguna manera las peticiones que hacen los usuarios, es decir tiene que guardar la relación entre petición, servidor e usuario. Al principio pensamos muchas maneras distintas de hacerlo pero muchas tenían inconvenientes, no por que guardaran menos información ni porque no representaran el estado de manera correcta si no porque no eran ineficientes. Lo primero que almacenamos de manera compartida es la información sobre los servidores, esta no representa el estado pero es necesaria para poder cambiar de estado. Después almacenamos una estructura de datos representada por un vector en el cual cada posición del vector representa un fichero y por posición hemos guardado un conjunto de peticiones. Este conjunto de peticiones lo hemos representado con un *MaxHeap* ya que preveíamos su utilidad para poder aplicar los operadores lo mas rápido posible. Por ultimo guardamos otras cualidades sobre el estado para poder calcular los heurísticos mas rápidamente, como por ejemplo un vector donde cada posición de este tenga el sumatorio de tiempos de transición del servidor de esa posición.

Tipos de estados iniciales

- **initalState1:** Este estado inicial mueve cada peticion al servidor que tarda menos en enviarla, escoge el servidor de entre los posibles que pueden enviar ese fichero a dicho usuario. Hemos implementado esta solución inicial ya que hemos pensado que si por cada fichero lo ponemos en un servidor que ya va muy rapido, entonces estamos ayudando al algoritmo a encontrar un estado muy bueno inicialmente. Aunque equilibremos la carga entre servidores.

- **initialState2:** Este estado mueve cada peticion a un servidor aleatorio de entre los posibles que pueden enviar ese ficher. Hemos elegido esta implementación de solución inicial ya que pensamos basarnos un poco en Simulated Annealing que es estocastico. Al escoger el estado inicial de manera aleatorio estamos provocando una exploración inicial que no esta presente en ningun momento en el Hill Climbing, ya que es un algoritmo que explota y escoge los mejores estados siempre, en otras palabras es puramente greedy. Con esta exploración incial sno estamos seguros de que si obtendremos mejores resultados al o empeoraremos, pero dado que la exploración es mínima lo mas probable es que acabemos obteniendo peores resultados que con el*initialState1*.

Tipos de operadores

- **Move max file:** Este operador selecciona el fichero que tarda mas del servidor que tiene su suma de tiempos de transmisión mas alta y lo mueve a cualquiera de los otros servidores posibles, de manera que tiene un *branch factor* de $k-1$ donde k es el numero de servidores que contienen dicho fichero. Hemos escogido este heurístico porque creemos que al mover un fichero del servidor mas lento, estamos equilibrando la carga entre servidores.

Tipos de heurístico

- **FirstHeuristicFunction:** Este heurístico tiene en cuenta el sumatorio de tres criterios

$$h(state) = h_1(state) + h_2(state) + h_3(state) \tag{1}$$

  donde h_1 es el el tiempo del servidor que tarda mas, h_2 es el total de todos los tiempos de transmisión y h_3 es el equilibrio de tiempos de transmisión entre servidores y esta representada por la varianza.

- **SlowestServerHeuristicFunction** Este heuristico representa el criterio h_1

- **TwoHeuristicFunction** Este heuristico representa el criterio h_1 + el criterio h_1.

Experimento 1

En este experimento, nuestro objetivo es descubrir qué combinación de nuestros operadores es el más efectivo, es decir, con el que obtenemos mejores resultados a partir de una misma inicialización. Las condiciones están detalladas a continuación:

Hipótesis

EN ESTA SECCION HAY QUE PONAR LA HIPÓTESIS, ES DECIR LOS RESULTAMOS QUE PENSAMOS OBTENER

Condiciones iniciales del experimento

- Nº de usuarios: 200
- Nº máximo de peticiones por usuarios: 5
- Nº servidores: 50
- Nº mínimo de replicaciones: 5
- Algoritmo: Hill Climbing
- Estrategia de inicialización: initialState2
- Heurístico usado: SlowestServerHeuristicFunction
- Operador usado: Todos, ya que son el sujeto de la prueba

Resultados del experimento

Experimento 2

El objetivo de este segundo experimento era averiguar que estrategia de estado inicial daba mejores resultados, de esta manera se fijara el mejor método para los siguientes experimentos.

Hipótesis

EN ESTA SECCION HAY QUE PONAR LA HIPÓTESIS, ES DECIR LOS RESULTAMOS QUE PENSAMOS OBTENER

Condiciones iniciales del experimento

- Nº de usuarios: 200
- Nº máximo de peticiones por usuarios: 5
- Nº servidores: 50
- Nº mínimo de replicaciones: 5
- Algoritmo: Hill Climbing
- Estrategia de inicialización: Todas, son el sujeto de la prueba
- Heurístico usado: SlowestServerHeuristicFunction
- Operador usado: **INSERTAR MEJOR OPERADOR ANTERIOR**

Resultados del experimento

Experimento 3

Una vez tenemos la mejor estrategia de inicialización y los mejores operadores, aplicamos el mismo heurístico y las mismas condiciones de prueba pero cambiamos el algoritmo inteligente. El algoritmo que se usa es Simulated Annealing y el objetivo es averiguar que parámetros dan mejores resultados en este algoritmo.

Hipótesis

EN ESTA SECCION HAY QUE PONAR LA HIPÓTESIS, ES DECIR LOS RESULTAMOS QUE PENSAMOS OBTENER

Condiciones iniciales del experimento

- Nº de usuarios: 200
- Nº máximo de peticiones por usuarios: 5
- Nº servidores: 50
- Nº mínimo de replicaciones: 5
- Algoritmo: Simulated annealing
- Parámetros del algoritmo: **INSERTAR PARAMETROS SIMULATED ANNEALING**
- Estrategia de inicialización: **INSERTAR MEJOR INIT ANTERIOR**
- Heurístico usado: SlowestServerHeuristicFunction
- Operador usado: **INSERTAR MEJOR OPERADOR ANTERIOR**

Resultados del experimento

Experimento 4

Esta vez queremos estudiar la evolución del tiempo de ejecución del algoritmo de Hill Climbing según el numero de usuarios y el numero de servidores. Al tenero dos parametros distintos el que hacemos es fijar 1 y probar otra y viceversa. Primero fijamos el numero de servidores en 50 y los usuarios en 100, entonces empezamos a incrementar los usuarios de 100 en 100. Para encontrar la tendencia de los servidores lo primero que hacemos es fijar los usuarios en 200 y los servidores en 50, entonces incrementamos los servidores de 50 en 50.

Hipótesis

EN ESTA SECCION HAY QUE PONAR LA HIPÓTESIS, ES DECIR LOS RESULTAMOS QUE PENSAMOS OBTENER

Condiciones iniciales del experimento

- Nº de usuarios: variable
- Nº máximo de peticiones por usuarios: 5, constante
- Nº servidores: variable
- Nº mínimo de replicaciones: 5, contante
- Algoritmo: Hill climbing
- Estrategia de inicialización: **INSERTAR MEJOR INIT ANTERIOR**
- Heurístico usado: SlowestServerHeuristicFunction
- Operador usado: **INSERTAR MEJOR OPERADOR ANTERIOR**

Resultados del experimento

Experimento 5

En este caso se ha implementado un nuevo heurístico que tenga en cuenta dos criterios distintos, y a partir de esto se calcula la diferencia entre el tiempo total de transmisión y el tiempo para hallar la solución, las condiciones son las mismas que las del primer apartado.

Hipótesis

EN ESTA SECCION HAY QUE PONAR LA HIPÓTESIS, ES DECIR LOS RESULTAMOS QUE PENSAMOS OBTENER

Condiciones iniciales del experimento

- Nº de usuarios: 200
- Nº máximo de peticiones por usuarios: 5
- Nº servidores: 50
- Nº mínimo de replicaciones: 5
- Algoritmo: Hill Climbing
- Estrategia de inicialización: **INSERTAR MEJOR INIT ANTERIOR**
- Heuristico usado: TwoHeuristicFunction
- Operador usado: **INSERTAR MEJOR OPERADOR ANTERIOR**

Resultados del experimento

Experimento 6

Una vez se ha hecho la prueba anterior, hemos repetido la prueba pero usando Simulated Annealing

Hipótesis

EN ESTA SECCION HAY QUE PONAR LA HIPÓTESIS, ES DECIR LOS RESULTAMOS QUE PENSAMOS OBTENER

Condiciones iniciales del experimento

- Nº de usuarios: 200
- Nº máximo de peticiones por usuarios: 5
- Nº servidores: 50
- Nº mínimo de replicaciones: 5
- Algoritmo: SimulatedAnnealing
- Parámetros del algoritmo: **INSERTAR PARAMETROS SIMULATED ANNEALING**
- Estrategia de inicialización: **INSERTAR MEJOR INIT ANTERIOR**
- Heuristico usado: TwoHeuristicFunction
- Operador usado: **INSERTAR MEJOR OPERADOR ANTERIOR**

Resultados del experimento

Experimento 7

En este experimento,tambien hemos medido la diferencia entre el tiempo de transmision y el tiempo en encontrar la solución pero nos hemos centrado en medir el tiempo de transmision segun el numero de replicaciones de los fichero, lo primero que se ha hecho es empezar con 5 replicaciones e ir incrimentando de 5 en 5 hasta llegar a 25 replicaciones por fichero.

Hipótesis

EN ESTA SECCION HAY QUE PONAR LA HIPÓTESIS, ES DECIR LOS RESULTAMOS QUE PENSAMOS OBTENER

Condiciones iniciales del experimento

- Nº de usuarios: 200
- Nº máximo de peticiones por usuarios: 5
- Nº servidores: 50
- Nº mínimo de replicaciones: variable
- Algoritmo: Hill Climbing
- Estrategia de inicialización: **INSERTAR MEJOR INIT ANTERIOR**
- Heuristico usado: TwoHeuristicFunction
- Operador usado: **INSERTAR MEJOR OPERADOR ANTERIOR**

Resultados del experimento

Experimento 8

AQUI COMPARAMEOS LOS RESULTADOS OBTENIDOS, ESTO; PODRIA EXPRESARSE COMO UNA CONCLUSIÓN EN VEZ DE UN EXPERIMENTO?!?

Experimento 9

ESTE ES EL PUNTO EXTRA; NOSE SI FALTA HACER UNA EXPLICACION