

Documentación de la práctica de búsqueda local

Laboratorio de Inteligencia Artificial

2º Cuatrimestre - curso 2018/2019



Grau en Informàtica

Departament de Ciències de la Computació



FIB

Facultat d'Informàtica
de Barcelona

UNIVERSITAT POLITÈCNICA DE CATALUNYA

Índice general

1. Organización, evaluación y entrega	2
2. Objetivos de aprendizaje	3
3. El problema	4
3.1. Elementos del problema	4
3.2. El problema	5
3.3. Criterios de la solución	5
3.4. Las clases java para los escenarios simulados	5
3.5. Tareas a realizar	5
3.6. Experimentos	6
3.7. Competencia de trabajo en equipo	7
4. Guión de la práctica	8
5. Planificación de tareas	11
6. Experimentación	13
6.1. ¿Que es un experimento?	13
6.2. Diseño experimental	13
6.3. Pasos para el diseño de un experimento	13
6.4. Como NO presentar los resultados de un experimento	14
6.5. Ejemplos de experimentación	14
6.5.1. TSP: Influencia de la solución inicial	15
6.5.2. TSP: Influencia del tamaño del problema	17
6.6. El Simulated Annealing	18
6.6.1. Ejemplo	20
7. Rúbrica de evaluación	25
8. Competencia de trabajo en equipo	32
8.1. Competencia de trabajo en equipo: práctica de búsqueda	32
8.2. Competencia de trabajo en equipo: Trabajo de innovación	33

Organización, evaluación y entrega

Esta es la documentación de la práctica de búsqueda local, en este documento tenéis:

- Los objetivos de aprendizaje de la práctica correspondientes al temario de la asignatura
- La descripción del problema que debéis resolver, los elementos que debéis implementar en la solución del problema y los experimentos que deberéis realizar con vuestra solución
- Lo que tenéis que incluir en el informe que deberéis entregar como resultado de la práctica
- La planificación semanal de la práctica incluyendo los objetivos que debéis ir cubriendo cada semana y la dedicación en horas estimada.
- Documentación sobre cómo realizar experimentos
- Documentación sobre como experimentar con el algoritmo de Simulated Annealing
- Rúbrica de evaluación de la práctica

La práctica se debe hacer **preferentemente en grupos de tres**. Si no tenéis grupo de prácticas hablad con vuestro profesor de laboratorio.

La práctica se debe desarrollar en **java** utilizando las clases que tenéis disponibles en la página web de laboratorio.

Planificad bien el desarrollo de la práctica y no lo dejéis todo para el último día, ya que no seréis capaces de acabarla y hacer un buen trabajo. En este documento tenéis indicaciones sobre el desarrollo de la práctica que os ayudará a planificar el trabajo.

En la evaluación de la práctica valoraremos sobre todo la calidad del análisis de cada experimento y los comentarios y conclusiones. Leed el capítulo de la rúbrica de evaluación para una descripción detallada de los criterios de evaluación. La garantía de una buena nota es seguir los criterios que aparecen en la rúbrica.

La entrega del informe y los fuentes de vuestro programa se hará el día **1 de abril** en formato electrónico según las instrucciones que aparecerán en el racó. También deberéis hacer una **reunión presencial** con vuestro profesor de laboratorio durante los días **3, 4 y 5 de abril** para hablar de la práctica y de como habéis trabajado en grupo, recibiréis instrucciones en el racó sobre ello. Tenéis un formulario de evaluación de la competencia de trabajo en equipo al final del enunciado que es el que rellenaréis en el momento de la entrega presencial.

Durante esa entrega también se evaluará la competencia de trabajo en equipo del trabajo de innovación. Tenéis también el formulario que utilizaremos para ello al final del enunciado.

Objetivos de aprendizaje

El objetivo de esta práctica es resolver un problema mediante algoritmos de **búsqueda local**.

En esta práctica se ha de demostrar ser capaz de razonar sobre la naturaleza del problema y plantearlo como un problema de búsqueda local, solucionándolo con la librería de algoritmos que tenéis a vuestra disposición.

Los objetivos específicos que se tienen que cubrir con el desarrollo de la práctica son los siguientes:

- Razonar sobre qué elementos son los necesarios para plantear el espacio de soluciones del problema y como deberían representarse para solucionarlo mediante un algoritmo de búsqueda local. Se han de tener en cuenta las necesidades espacio/temporales de los algoritmos para elegir la representación.
- Determinar qué formas existen para generar una solución inicial del problema y razonar y justificar cuales son las más adecuadas. Se han de elegir varias alternativas para poder experimentar y probar las decisiones empíricamente.
- Determinar qué operadores de transformación sobre la solución son posibles en el problema y qué combinación de estos operadores es adecuada para solucionar el problema considerando el factor de ramificación y la posibilidad de poder generar cualquier solución.
- Determinar y justificar las funciones heurísticas que permitan optimizar la solución según unos criterios específicos. Comparar las funciones heurísticas empíricamente y razonar sobre los resultados.
- Plantear un conjunto de experimentos que muestren la ejecución de los algoritmos de búsqueda Hill Climbing y Simulated Annealing en escenarios diferentes. Escoger los experimentos que demuestren el funcionamiento de los elementos que se han elegido (solución inicial, operadores, función heurística) en diferentes situaciones del problema. Debéis de ser capaces de justificar su elección y comparar lo que se esperaba intuitivamente con el experimento con los resultados reales.
- Experimentar con los parámetros del Simulated Annealing para escoger los más adecuados en los experimentos.
- Comparar el comportamiento del algoritmo de Hill Climbing y el de Simulated Annealing respecto a la calidad de sus soluciones.

Respecto a la **evaluación**, tenéis disponible una rúbrica que indica los criterios que se usarán para valorar la práctica y una descripción de cada uno de los niveles de valoración para cada criterio.

El problema

El cambio climático nos está obligando a encontrar maneras de reducir el gasto energético y el CO₂ emitido. El transporte privado en las ciudades es uno de los mayores generadores de CO₂, además de ser una de las principales fuentes de contaminación del aire.

Esto ha llevado a diferentes iniciativas para reducir el uso del transporte privado, ya sea promocionando los medios de transporte público, como fomentando la mayor ocupación del transporte privado.

Una iniciativa en este sentido es la compartición de coches entre personas. La idea es que de una manera rotativa durante un periodo de tiempo una persona use su coche para trasladar a otras personas a su lugar de trabajo, de manera que haya más coches que tengan la mayoría de sus plazas ocupadas, reduciendo el tráfico, el consumo de combustible y la emisión de contaminantes.

La mayoría de los sistemas de compartición de coche confían en que de manera espontánea las personas que están apuntadas se pongan de acuerdo y se llegue a una solución aceptable, pero obviamente esa solución no tiene por que ser óptima, ni siquiera demasiado buena.

Nos podemos plantear si, usando algoritmos de búsqueda local, podemos obtener soluciones que permitan indicar al conjunto de usuarios cómo deben organizarse, mejorando así la eficiencia del sistema.

3.1. Elementos del problema

Supondremos que hemos puesto en marcha un servicio de compartición de coches en el que cada uno de los usuarios que se apunta al servicio indica las plazas que tiene libres en su coche, la dirección en la que vive y la dirección de su trabajo. Para simplificar supondremos que todos los usuarios tienen siempre dos plazas libres.

Cada mes, de manera rotativa, un grupo de usuarios pone a disposición del resto de usuarios las plazas que tiene libres en su coche. Durante ese mes las personas que comparten su coche adaptan su recorrido para recoger a otros usuarios y dejarlos en sus puestos de trabajo. En su recorrido, el coche puede recoger a más personas que las plazas libres del coche, si los desplazamientos de las personas que se recogen no se solapan entre si.

Supondremos que en el servicio hay N personas apuntadas y que en un mes concreto hay M personas (obviamente $M < N$) que pueden conducir ese mes.

Para simplificar el problema supondremos que la ciudad es un cuadrado de 10×10 kilómetros y que las calles forman una cuadrícula donde cada manzana tiene 100×100 metros. Las direcciones de los usuarios y los destinos corresponderán con cruces de la cuadrícula. El cálculo de la distancia entre dos puntos de la ciudad se hará mediante la función:

$$d(i, j) = |i_x - j_x| + |i_y - j_y|$$

Donde i_x y i_y son las coordenadas x e y del punto i en la cuadrícula.

3.2. El problema

Queremos obtener una solución para este problema para un mes concreto, buscamos generar un plan que indique a cada persona que comparte su coche ese mes, un recorrido que le permita recoger y dejar a un conjunto de usuarios sin sobrepasar las plazas libres de su coche y cumpliendo las restricciones del problema. **Tened en cuenta que solo es una solución la que consigue llevar a todo el mundo.** Es posible que no todas las personas a las que les toque conducir ese mes tengan que conducir, en ese caso serán pasajeros de otros coches.

Una solución deberá indicar:

- Qué personas hacen de conductor en el mes actual.
- Qué recorrido hace cada persona que hace de conductor, es decir, a qué personas recoge y en qué orden las recoge y las deja.

3.3. Criterios de la solución

Para obtener y evaluar la solución usaremos los siguientes criterios y restricciones:

- El dueño del coche ha de poder ser capaz de llegar a tiempo a su trabajo, lo que limita la longitud total del recorrido que debe hacer. Supondremos que la velocidad media del tráfico en la ciudad es de 30 km/h, que el coche iniciará su recorrido a las 7 de la mañana y el usuario ha de estar en el trabajo a las 8 de la mañana como muy tarde.
- La solución ha de recoger y dejar a todas las personas del servicio.
- En un coche no puede haber más de 3 personas en un momento dado (incluyendo al conductor).
- Solo las M personas designadas como conductores para ese mes pueden conducir.
- Tendremos dos criterios para evaluar la calidad de una solución:
 1. Minimizar la distancia recorrida por los conductores.
 2. Minimizar la distancia recorrida por los conductores y minimizar el número de conductores.

3.4. Las clases java para los escenarios simulados

Tenéis disponible un conjunto de clases en java que generan los escenarios simulados. Podéis descargar las clases del aviso del racó donde estaba este enunciado.

La librería que contiene las clases se llama `IA.Comparticion` y contiene dos clases:

1. **Usuarios**, que genera una estructura con todos usuarios que están apuntados al servicio. El constructor recibe tres parámetros, el número de usuarios a generar, el número de posibles conductores del mes y la semilla para el generador de números aleatorios.
2. **Usuario**, que tiene la estructura que representa un usuario, es decir sus coordenadas de origen y destino, y si es conductor este mes o no. La distribución geográfica de las coordenadas de los usuarios es equiprobable. Las coordenadas de un usuario usan como unidades el centenar de metros.

Podéis consultar la documentación de las clases en los ficheros `javadoc` incluidos.

3.5. Tareas a realizar

El desarrollo de la práctica implica realizar las siguientes tareas:

1. Implementar la clase del estado del problema de tal manera que se puedan generar problemas aleatorios. En este caso los elementos que varían son:
 - El número de personas apuntadas al servicio y sus coordenadas de origen y destino.

- El número de personas que pueden conducir este mes.
2. Definir e implementar la representación del estado del problema para poder ser resuelto utilizando las clases del AIMA. Pensad bien en la representación, ha de ser eficiente en espacio y en tiempo.
 3. Definir e implementar dos estrategias para generar la solución inicial. Pensad que en este problema puede ser difícil comenzar desde un estado inicial que sea solución, por lo que tendréis que evaluar la posibilidad de usar estrategias de inicialización que no cumplan todas las restricciones. Pensad también en las implicaciones de esto en los otros elementos de la búsqueda, sobre todo en la función heurística.
 4. Definir e implementar la función generadora de estados sucesores. Esto implica decidir el conjunto de operadores para explorar el espacio de búsqueda. Deberéis pensar y evaluar diferentes alternativas de conjuntos de operadores y justificar la elección de uno de ellos para realizar los experimentos. Deberéis implementar la función generadora de manera diferente para Hill Climbing y Simulated Annealing para que se puedan comparar sus tiempos de ejecución tal como se explicó en clase de laboratorio.
 5. Definir e implementar dos funciones heurísticas, cada una de ellas implementará uno de los criterios de calidad de la solución indicados en el apartado anterior. Pensad que en la segunda función heurística tenéis dos criterios que se miden en unidades diferentes, tendréis que pensar en maneras de combinarlos para obtener una función heurística que tenga en cuenta más o menos por igual los dos criterios. Pensad también que si comenzáis desde el espacio de no soluciones también sería una buena idea introducir en las funciones heurísticas algún criterio que penalice las no soluciones, de manera que la búsqueda se oriente hacia el espacio de soluciones.

3.6. Experimentos

Deberéis hacer los siguientes experimentos:

1. Determinar qué conjunto de operadores da mejores resultados para una función heurística que optimice el primer criterio de calidad (3.3) con un escenario en el que el número usuarios apuntados al servicio (N) es 200 y en número de personas que podrían conducir en el mes actual (M) es 100. Deberéis usar el algoritmo de Hill Climbing. Escoged una de las estrategias de inicialización de entre las que proponéis. A partir de estos resultados deberéis fijar los operadores para el resto de experimentos.
2. Determinar qué estrategia de generación de la solución inicial da mejores resultados para la función heurística usada en el apartado anterior, con el escenario del apartado anterior y usando el algoritmo de Hill Climbing. A partir de estos resultados deberéis fijar también la estrategia de generación de la solución inicial para el resto de experimentos.
3. Determinar los parámetros que dan mejor resultado para el Simulated Annealing con el mismo escenario, usando la misma función heurística y los operadores y la estrategia de generación de la solución inicial escogidos en los experimentos anteriores.
4. Dado el escenario de los apartados anteriores, estudiad como evoluciona el tiempo de ejecución para hallar la solución en función del número de personas que tenemos en el servicio asumiendo que $M = N/2$. Para ello empezad con 200 usuarios e id aumentándolos 100 en 100 hasta que veáis la tendencia. Usad el algoritmo de Hill Climbing y la misma heurística.
5. Dado el escenario del primer apartado, estimad la diferencia entre la distancia total recorrida y el tiempo de ejecución para hallar la solución, usando las dos heurísticas, la que habéis usado en los apartados anteriores y otra que optimice el segundo criterio de calidad propuesto (3.3). Usad el algoritmo de Hill Climbing. En este caso tendréis que buscar como se han de ponderar los criterios de la segunda heurística.
6. Dado el mismo escenario, repetid los experimentos usando Simulated Annealing y las dos heurísticas y comparad los resultados usando la distancia total recorrida y el tiempo de ejecución.

7. El usar la proporción $M = N/2$ es demasiado conservadora. Utilizar un valor menor para M , permitiría hacer que el algoritmo estuviera más cerca de la solución y reduciría el tiempo de exploración. Realizad experimentos con diferentes tamaños de problema que os permita haceros una idea de cuál es aproximadamente la proporción de conductores necesarios para cada tamaño. ¿Depende ese valor del número total de personas en el servicio? ¿Si se fija al inicio el número de posibles conductores a la proporción que habéis hallado, la calidad de la solución se mantiene? ¿se reduce efectivamente el tiempo de cálculo si se empieza con ese valor? Haced los experimentos con el algoritmo y la función heurística que hayan dado los mejores resultados.

Fijaos que no os estamos pidiendo que experimentéis con diferentes valores de M , sino que partiendo del valor $M = N/2$, veáis cuál es el número (o la proporción) de conductores que aproximadamente son necesarios en las soluciones para diferentes totales de usuarios y probéis si usando ese valor aún se obtienen buenas soluciones.

8. **Experimento Especial:** Para fomentar el trabajo continuado en la practica siguiendo la planificación, asignaremos un punto extra sobre la nota de la práctica a los grupos que envíen durante la semana del 18 al 24 de marzo un correo con el resultado de la longitud total del recorrido de los conductores y el número de conductores que se obtiene en el escenario con 200 usuarios y 100 conductores y cuánto tiempo se tarda en hallar la solución en milisegundos (aproximadamente). Para que todos los experimentos sean con las mismas condiciones usaremos como semilla del generador de números aleatorios 1234, para generar los usuarios. Deberéis usar los operadores, inicialización y heurística escogidos con los experimentos 1 y 2.

Los grupos que manden el correo, tendrán que enseñar la ejecución de la práctica con este escenario al profesor de laboratorio durante las sesiones de laboratorio de la semana siguiente. Obviamente ha de dar el mismo resultado.

Pensad que si comenzáis desde el espacio de no soluciones es posible que en algunas ejecuciones, o para algunos parámetros, los experimentos no den una solución válida. Si es así, observad en qué circunstancias sucede, e intentad dar una explicación.

Para cada experimento deberéis hacer como mínimo 10 repeticiones y calcular valores medios. Podéis hacer gráficas y estadísticas que ilustren vuestras conclusiones, obviamente los que lo hagáis tendréis una mejor valoración.

Tened en cuenta que el tamaño del heap de memoria que usa por defecto java es muy pequeño, si os encontráis con problemas de memoria, podéis aumentar el tamaño usando el flag `-Xmx` indicando el tamaño que se quiere, por ejemplo, `java -Xmx256m` hace que el java utilice hasta 256Mb de memoria de heap.

Explicad y justificad todas las decisiones que toméis. Sacad conclusiones de los experimentos que incluyan **comentarios** sobre los resultados obtenidos en cada experimento. Explicad qué esperabais y qué habéis obtenido. Tenéis indicaciones sobre el contenido de la documentación en el guión de la práctica que esta en el siguiente capítulo de este documento.

Mirad también la rúbrica de evaluación de la práctica (capítulo 7).

3.7. Competencia de trabajo en equipo

La competencia de trabajo en equipo se evalúa de dos maneras. La primera es sobre el trabajo de la práctica, en la sección 8.1 tenéis un formulario que rellenaréis en la entrega presencial. Es importante que lo leáis al principio para que tengáis presentes los diferentes criterios para evaluar vuestro trabajo en equipo.

El segundo elemento para evaluar la competencia de trabajo en equipo es el trabajo de innovación. En este trabajo sois vosotros los que planificáis y decidís. Para poder hacer el seguimiento tenéis que añadir al informe de la práctica la información que se os indican en la sección 8.2.

Guión de la práctica

Primera semana: Las clases del AIMA (18 de febrero)

Esta semana veréis en clase de laboratorio cómo utilizar las clases del AIMA para resolver problemas de búsqueda. Estas clases implementan la mayoría de los algoritmos que se han visto en clase y tenéis diferentes ejemplos de su utilización.

Tenéis ejemplos que resuelven problemas de búsqueda heurística y problemas de búsqueda local. Los primeros ejemplos que veréis son problemas de búsqueda heurística, en la práctica resolveréis un problema de búsqueda local, pero desde el punto de vista de realizar la implementación, los elementos que intervienen son los mismos, solo varía el significado que les asigna el algoritmo que se ha de ejecutar.

Fijaos que vosotros no tenéis que implementar los algoritmos, solo tenéis que identificar e implementar los elementos que estos necesitan. Estos elementos son los que se os han explicado en clase de teoría: Estado, operadores de búsqueda y función heurística.

Ejecutad los ejemplos que tenéis en las transparencias de laboratorio y echad un vistazo al código de los ejemplos que tenéis disponibles.

Segunda semana: Toma de contacto con el problema (25 de febrero)

A pesar de que todavía no podáis empezar a desarrollar la práctica deberíais haberos leído el enunciado y haberos hecho una idea sobre el problema.

Una cosa que deberíais hacer es reflexionar sobre el problema que describe el enunciado y preguntaros las mismas cosas que se preguntan en los problemas que hicisteis la primera clase de problemas.

- ¿Qué elementos intervienen en el problema?
- ¿Cuál es el espacio de búsqueda?
- ¿Qué tamaño tiene el espacio de búsqueda?
- ¿Qué es un estado inicial?
- ¿Qué condiciones cumple un estado final?
- ¿Qué operadores permiten modificar los estados?
- ¿Qué factor de ramificación tienen los operadores de cambio de estado?

A partir de las respuestas a esas preguntas os podéis plantear lo que necesitaréis para implementar la práctica con las clases del AIMA.

Un primer ejercicio consiste en que penséis qué estructura de datos debéis implementar para representar el estado. Es fundamental que penséis la representación teniendo en cuenta la mayor eficiencia espacial y temporal, ya que la búsqueda generará una gran cantidad de estados.

Podéis mirar el código de los ejemplos y prácticas anteriores que tenéis disponibles para haceros una mejor idea de lo que necesitareis implementar.

No dudéis en preguntar vuestras dudas a vuestros profesores de laboratorio.

Tercera semana, implementación del estado (4 de marzo)

Esta semana ya deberíais tener claros los elementos que permiten definir el problema de la práctica. Ahora deberíais empezar a plantearos la implementación de la clase que representa el estado.

La implementación del estado incluye la decisión de qué estructura de datos es más adecuada para representar los elementos del problema. Debería ser eficiente ya que la exploración ha de generar un número bastante grande de estados.

Para ganar eficiencia espacial es una buena idea declarar estáticas las partes de la representación que no cambian, de manera que se compartan entre todas las instancias.

Deberéis implementar constructores que generen la solución inicial. El enunciado os pide que busquéis al menos dos estrategias para generarla.

Tenéis que pensar varias cosas sobre cómo generar la solución inicial, entre ellas cuál es el coste de generarla y cómo de buena es.

De cara a hacer los experimentos interesa ver como influye la bondad de la solución en el resultado de la búsqueda. Por ejemplo, podéis usar una estrategia que genera soluciones muy malas y al menos otra que presumiblemente sea mejor según algún criterio, deberéis ver si el número de pasos hasta llegar a la solución final y su calidad es diferente.

Dentro de la implementación del estado también esta incluida la implementación de los operadores de búsqueda. Tenéis que analizar qué conjunto de operadores es más conveniente. Observad que el factor de ramificación es importante, porque influye directamente en el tiempo para hallar la solución. Debéis pensar también que unos operadores que no generen suficientes alternativas pueden dar lugar a soluciones peores por no permitir explorar correctamente el espacio de búsqueda.

Pensad bien en las operaciones que se pueden hacer en el problema. A veces hacen falta varias para poder acceder a todo el espacio de búsqueda, también puede haber conjuntos alternativos de operadores. Deberéis sopesar el factor de ramificación y la conectividad que se obtiene entre las soluciones.

También deberíais ir escribiendo la documentación de la práctica a medida que vais haciendo cosas, no lo dejéis para el final.

Cuarta semana: Implementación de las clases para el AIMA (11 de marzo)

A estas horas ya deberíais tener implementado el estado del problema junto con sus operadores y las estrategias de inicio de la búsqueda.

Ahora necesitáis implementar el resto de clases que el AIMA usa para resolver el problema.

Implementar la clase que genera los estados sucesores es sencillo, solo tenéis que generar para un estado todos los estados accesibles posibles. Como ya tendréis implementados los operadores solo tendréis que decidir cómo se hace la generación de los nodos aplicando los operadores. El orden en que se generan es indiferente, solo tenéis que aseguraros de que para cada nodo se generan todos los sucesores accesibles. Recordad que la estrategia para generar los nodos ha de ser diferente para Hill Climbing que para Simulated Annealing. Para Hill Climbing tendréis que generar todas las posibles aplicaciones de los operadores al estado actual, mientras que para Simulated Annealing tendréis que escoger al azar un operador y generar solo un sucesor aplicando este operador con parámetros también al azar.

Con esto es suficiente, ya que es el algoritmo de búsqueda el que se encargará de hacer la exploración y decidir qué nodos se expanden.

La clase que comprueba si se ha llegado a un estado final es la más sencilla de todas, en el caso de búsqueda local no es posible saber si se ha llegado al estado final, por lo que la función que implementa esta clase ha de retornar siempre **falso**. Podéis copiar esta clase de cualquiera de los ejemplos de búsqueda local que tenéis, todas son iguales.

La clase que debéis pensar más es la que calcula la función heurística.

Tenéis que pensar en qué miden las funciones heurísticas que describe el enunciado y decidir cómo se calculan a partir del estado.

Acordaos de que las funciones heurísticas que implementéis han de minimizarse. Pensad también que la diferencia entre maximizar y minimizar es un cambio de signo.

Quinta semana: Experimentos (18 de marzo)

Durante esta semana deberíais tener ya una implementación funcional de la práctica y comenzar a hacer experimentos.

Deberéis pensar en los diferentes escenarios que se pueden plantear con los elementos que tenéis. Tened en cuenta que el objetivo de los experimentos es obtener información que os permita responder a las preguntas que plantea el enunciado. Tened en cuenta que un mismo experimento os puede dar información para varias preguntas.

Seguid el orden de los experimentos que tenéis en el enunciado, éste os permitirá ir tomando decisiones sobre los diferentes elementos del problema e ir fijándolas para experimentos sucesivos.

Para ajustar los parámetros del Simulated Annealing escoged valores extremos y probad sus efectos. A partir de ellos podéis ir ajustando más sus valores explorando puntos intermedios hasta llegar a un valor que os parezca adecuado según el objetivo del problema. Tened en cuenta el significado de los parámetros para guiaros en vuestra exploración.

Haced suposiciones sobre cómo deberían ser las soluciones en cada experimento y comprobad los resultados que obtenéis. Comparad si los resultados corresponden con vuestras intuiciones e intentad justificar el resultado.

Tened en cuenta que para sacar conclusiones con fundamento deberéis ejecutar cada experimento varias veces. Para poder comparar deberéis ejecutar los algoritmos con datos iguales, para ello podéis generarlos fijando las semillas del generador de números aleatorios al crear los datos.

Podéis justificar la significatividad de los resultados que obtenéis utilizando los conocimientos que habéis adquirido en la asignatura de estadística. También podéis utilizar gráficos para ilustrar lo que sucede en los experimentos.

No os olvidéis de ir escribiendo la documentación a medida que hagáis los experimentos.

Sexta semana: La documentación final (25 de marzo)

En esta semana deberíais tener los resultados de los experimentos y escribir la documentación.

Algo que tenéis que tener presente es que la documentación ha de ser un reflejo de vuestro trabajo ya que es lo que servirá para que califiquemos vuestra práctica. Una mala documentación significa una mala nota.

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 que pueda haber en el enunciado.

Planificación de tareas

Esta planificación os puede servir como orientación para organizar y repartir el trabajo de la práctica. Tened en cuenta que una planificación adecuada beneficiará la calidad de vuestra práctica.

Las horas asignadas al trabajo de la práctica según los créditos ECTS que le corresponde son 25 por cada componente del grupo, más las horas de las clases de laboratorio que son 6. En total la práctica corresponde a un esfuerzo de 93 horas durante seis semanas. Tened en cuenta que son horas ECTS, por lo que este es el tiempo de dedicación de un alumno medio para hacer la práctica a un nivel aceptable (digamos que una nota de notable) para llegar al sobresaliente hace falta algo más de dedicación.

Las tareas a desarrollar dentro de las semanas que indica el guión son las que aparecen en la siguiente tabla. También se indican las horas de dedicación aproximada del grupo en conjunto.

Semana 1 (3h lab + 9 horas trabajo externo)
<ol style="list-style-type: none"> Entender como funciona AIMA Mirar/ejecutar los ejemplos Leer y entender el enunciado
Semana 2 (3h lab + 15 horas trabajo externo)
<ol style="list-style-type: none"> Entender cómo funciona AIMA Pensar la representación y elementos del problema Diseñar la estructura de datos del estado Definir la interfaz de acceso a la estructura Documentar la estructura de datos del estado
Semana 3 (3h lab + 15 horas trabajo externo)
<ol style="list-style-type: none"> Implementación del estado Implementación de la generación del estado inicial Implementación de los operadores de búsqueda Documentación
Semana 4 (3h lab + 12 horas trabajo externo)
<ol style="list-style-type: none"> Función generadora de sucesores Función que indica el estado final Cálculo de las funciones heurísticas Documentación

Semana 5 (3h lab + 12 horas trabajo externo)
<ol style="list-style-type: none"> 1. Planificación de los experimentos 2. Realizar los experimentos y recolectar resultados 3. Analizar los resultados de los experimentos
Semana 6 (3h lab + 12 horas trabajo externo)
<ol style="list-style-type: none"> 1. Documentación

Las horas de laboratorio de las dos primeras semanas os servirán para aprender como funciona el AI-MA. El resto de horas de laboratorio las deberéis dedicar para las consultas al profesor de laboratorio, sincronización entre los componentes del grupo y desarrollo de la práctica.

La tarea de planificación de los experimentos en la quinta semana es muy importante. Una vez decididos los experimentos a realizar, estos se pueden ejecutar en batch y después analizar los resultados.

Es importante que os dividáis las tareas entre los tres miembros del grupo de manera adecuada. Muchas de las tareas se pueden hacer de manera paralela. Y recordad que tres personas delante de un PC durante una hora son una hora de trabajo, no tres.

Experimentación

6.1. ¿Que es un experimento?

“Un experimento es un *procedimiento* mediante el cual se trata de comprobar (confirmar, verificar o inventar) una o varias hipótesis relacionadas con un determinado fenómeno, mediante la manipulación de una o más variables que presumiblemente son su causa.”

(Wikipedia, <http://es.wikipedia.org/wiki/Experimento>)

6.2. Diseño experimental

“El diseño experimental es una técnica estadística que permite identificar y cuantificar las causas de un efecto dentro de un estudio experimental. En un diseño experimental se manipulan deliberadamente una o más variables, vinculadas a las causas, para medir el efecto que tienen en otra variable de interés. El diseño experimental prescribe una serie de pautas relativas a qué variables hay que manipular, de qué manera, cuántas veces hay que repetir el experimento y en qué orden para poder establecer con un grado de confianza predefinido la necesidad de una presunta relación de causa-efecto.”

(Wikipedia, http://es.wikipedia.org/wiki/Diseño_experimental)

6.3. Pasos para el diseño de un experimento

- Observación
- Planteamiento del problema de investigación
- Hipótesis: hipótesis nula (H_0) e hipótesis alterna
- Método (incluye la elección de los sujetos, para la conformación de la muestra; el procedimiento a seguir, es decir, el tratamiento a aplicar a los sujetos; las variables consideradas: variable dependiente, variable independiente, variables extrañas)
- Resultados: aquí se describen cuáles fueron las relaciones observadas entre las variables (si los valores de la variable independiente realmente influyeron significativamente sobre los de la variable dependiente, si hubo tantas variables extrañas como se pensaba o si surgieron otras), para lo cual se añaden a dicha descripción tanto gráficas (de barras, de pastel, etc.) como cuadros.
- Conclusiones

(Wikipedia, http://es.wikipedia.org/wiki/Diseño_experimental)

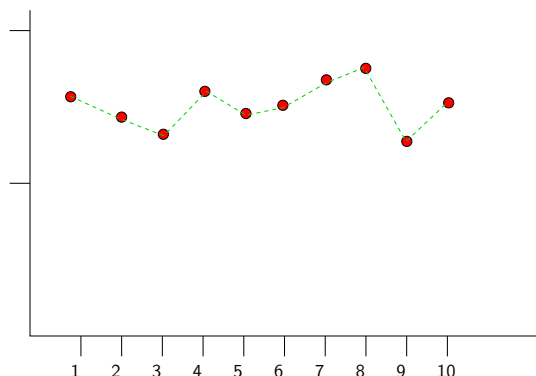


Figura 6.1: Gráfico de las alturas de 10 personas

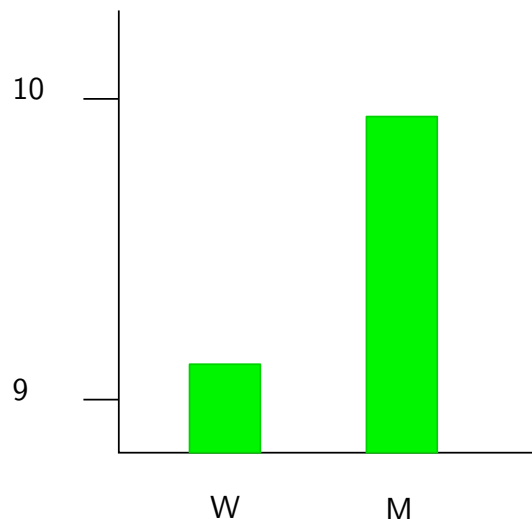


Figura 6.2: Gráfico de la media de la longitud del dedo índice de la mano derecha para 5 hombres y 5 mujeres

6.4. Como NO presentar los resultados de un experimento

Hay muchas maneras de presentar los resultados de un trabajo experimental, pero hay algunas que no son informativas o adecuadas y que deben ser evitadas.

El primer error es incluir tablas con los resultados individuales de cada experimento. Obviamente, nadie se va a mirar cada valor de la tabla, ya que el objetivo del trabajo es resumir y destacar las conclusiones que se pueden extraer de los datos. Si solo hay unos pocos resultados es admisible el añadir un apéndice con los resultados, pero si hay muchos resultados no es algo práctico.

Cuando se presentan las conclusiones de los experimentos, se debe utilizar información estadística que resuma los datos. La estadística se basa en modelos y se pueden obtener conclusiones de sus parámetros. Por ejemplo, si asumimos que los datos del experimento se distribuyen según una normal, la desviación estándar pueden dar información sobre la estabilidad/homogeneidad del experimento. También se puede usar métodos estadísticos para probar hipótesis, por ejemplo para ver si los resultados de dos experimentos diferentes son equivalentes.

Los gráficos son también importantes al mostrar los resultados del trabajo experimental y pueden complementar y representar la información estadística, pero han de ser informativos y no llevar a conclusiones erróneas.

Por ejemplo, supongamos que realizamos un experimento y medimos la altura de un grupo de personas y presentamos la información como en la figura 6.1. Ese gráfico no da ninguna información útil, los datos no son una serie temporal, es solo la tabla de datos representada como una gráfica. Sería mas informativo presentar la media y desviación de las alturas o representarlas como un diagrama de caja (boxplot).

Otro error común es exagerar las diferencias entre medidas para intentar que algo que no es muy significativo lo parezca (este método es usado por la prensa sensacionalista y los políticos). Por ejemplo, podemos hacer una nueva medida a nuestro grupo de personas, en este caso, la longitud del dedo índice de la mano derecha y presentar los resultados separado por sexos, como en la figura 6.2. Podemos argumentar, sin mirar a la escala del eje Y, que el valor para los hombres es mucho mayor que para las mujeres, cuando en realidad usando la escala correcta la diferencia no es tan grande. Es pues recomendable poner en la correcta perspectiva los resultados que presentamos.

6.5. Ejemplos de experimentación

Para ilustrar como hacer una serie de experimentos dentro del tipo de problemas que se solucionan en esta práctica, vamos a hacer algunos experimentos con el problema del viajante de comercio usando la

implementación que hay en las clases de AIMA.

6.5.1. TSP: Influencia de la solución inicial

Un elemento importante en un problema de búsqueda local es la influencia del punto de partida en el coste de la solución final. A priori no se puede determinar si un método de inicio es mejor o peor dado que se desconoce la forma que tendrá el espacio de búsqueda.

Para probar la hipótesis de que hay métodos de inicialización mejores que otros vamos a realizar un experimento probando diferentes métodos y evaluando sus resultados. Consideraremos tres métodos distintos de inicialización: Camino inicial ordenado según el identificador de la ciudad, camino inicial aleatorio y camino obtenido mediante una estrategia avariciosa.

Para poder hacer el experimento en condiciones debemos garantizar que todos los experimentos con cada hipótesis se realizan en las mismas condiciones. Para ello vamos a realizar diez réplicas del experimento donde probaremos cada método de inicialización. Para cada experimento individual utilizaremos la misma semilla de números aleatorios, de manera que cada réplica sea idéntica en las pruebas para cada método de inicialización.

En este caso podemos medir diferentes cosas en cada experimento, pero antes de hacerlo debemos pensar en qué es comparable y qué no.

Por un lado, el valor de la solución de cada experimento no es comparable, ya que cada réplica partirá de condiciones diferentes. En este caso nos puede interesar para cada experimento cual de los métodos obtuvo la mejor solución.

También nos puede interesar el tiempo que tardaron en solucionarse los problemas y el número de pasos que se hicieron hasta llegar a la solución. Estas magnitudes son más comparables entre diferentes réplicas ya que deberían depender del tamaño del problema (en este caso las ciudades). Hay que fijarse también que estas dos medidas están altamente correlacionadas, por lo que nos llevarán a las mismas conclusiones.

Otra dificultad en nuestra experimentación es que uno de los métodos es aleatorio, por lo que diferentes ejecuciones del experimento con condiciones iguales dará diferentes resultados. Para poder comparar con los otros métodos que siempre darán la misma solución en las mismas condiciones podemos hacer la media de los resultados obtenidos de varias ejecuciones.

Podemos resumir las características de este experimento en la siguiente tabla:

Observación	Pueden haber métodos de inicialización que obtienen mejores soluciones
Planteamiento	Escogemos diferentes métodos de inicialización y observamos sus soluciones
Hipótesis	Todos los métodos de inicialización son iguales (H_0) o hay métodos mejores que otros
Método	<ul style="list-style-type: none">▪ Elegiremos 10 semillas aleatorias, una para cada réplica▪ Ejecutaremos 1 experimento para cada semilla para la inicialización ordenada y avariciosa▪ Ejecutaremos 5 experimentos para cada semilla para la inicialización aleatoria y haremos medias de los resultados▪ Experimentaremos con problemas de 40 ciudades▪ Usaremos el algoritmo de Hill Climbing▪ Mediremos diferentes parámetros para realizar la comparación

Una vez realizado el experimento debemos recolectar la información que queremos usar para probar nuestras hipótesis. En la tabla 6.3 podemos ver los resultados de un experimento como el que hemos definido.

A partir de estos datos podemos empezar a dar respuestas a nuestras hipótesis.

Por ejemplo, podemos comparar cada pareja de métodos de inicialización asumiendo que todos son igual de buenos (la hipótesis nula). Para ello podemos considerar que la probabilidad de que un método dé una solución mejor que otro para un experimento se distribuye como una función binomial y podemos comprobar cual es la probabilidad de que la hipótesis nula sea cierta (que entre dos métodos, los dos tienen la misma probabilidad de dar la mejor solución).

Répica	Solución			Pasos			Tiempo		
	Greedy	Ordenado	Aleat.	Greedy	Ordenado	Aleat.	Greedy	Ordenado	Aleat.
1	150	214	188	1	21	26	28	253	239
2	156	172	187,6	1	26	22,2	25	243	212
3	172	184	189,6	7	26	26,6	79	239	246,4
4	104	208	163,6	1	19	24,4	20	181	227
5	150	212	208,8	6	23	24,2	71	211	223,6
6	142	194	192,8	5	27	23,8	51	253	221,2
7	142	202	206,8	1	24	22,8	26	225	216,2
8	138	172	184	6	37	28,6	69	341	267,6
9	134	182	198,4	4	21	23	54	210	220
10	142	184	202	4	26	24	45	237	224,2

Figura 6.3: Valor de la solución, número de pasos y tiempo necesitado para hallarla

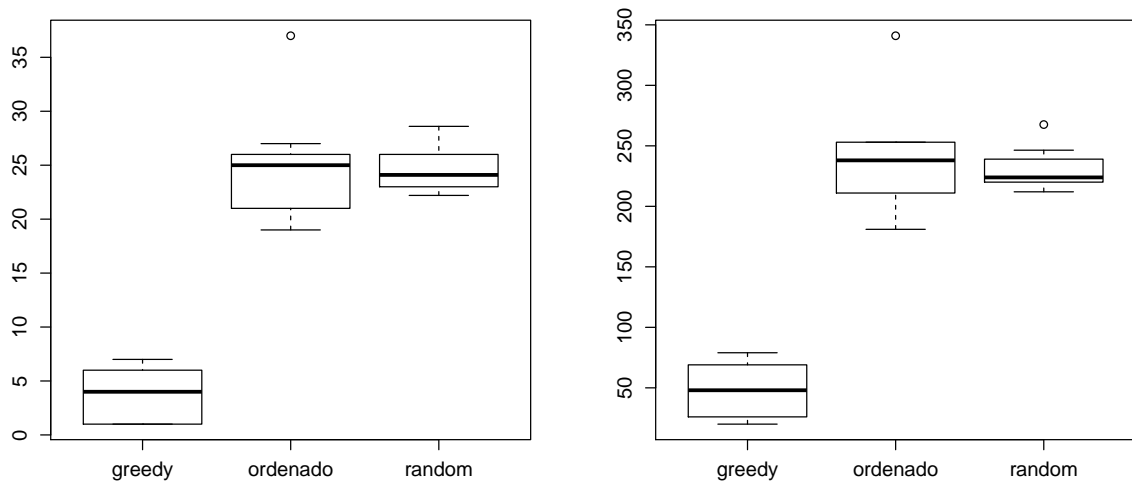


Figura 6.4: Distribución del número pasos y el tiempo de ejecución

Para el caso de la inicialización avariciosa respecto a la ordenada tenemos que siempre la avariciosa da una solución mejor, por lo que deberíamos consultar en las tablas de la binomial cual es la probabilidad de que en 10 experimentos tengamos 10 experimentos donde avariciosa gane, asumiendo que son igual de buenos (tienen una probabilidad de 0,5 de dar el mejor resultado), con ello podemos comprobar que esa probabilidad es 0,001, o sea, bastante remota. Esto querría decir que no es cierto que sean iguales. Sucede lo mismo con la comparación entre el avaricioso y el aleatorio.

Para la comparación entre el ordenado y el aleatorio, tenemos que el ordenado es mejor 6 de las 10 veces. Si consultamos la binomial tendremos que la probabilidad es de 0,205, significativamente más alta que entre estas y la avariciosa.

Podemos comparar también las distribuciones estadísticas del número de pasos y el tiempo en hallar la solución, en la siguiente tabla tenemos la media y desviación de estas variables:

	Greedy	Orden	Aleatorio
Pasos	3.6 (2.41)	25 (4.98)	24.56 (2.96)
Tiempo	46.8 (21,52)	239.3 (42.16)	229.7 (16,8)

En la gráfica 6.4 se puede apreciar mejor la distribución del número de pasos y el tiempo de ejecución. Es obvio que el método avaricioso es mucho mas rápido que los otros dos. La explicación más plausible es

que la solución inicial se encuentra muy cerca de la solución final si usamos este método. Los otros comienzan en una solución mas lejana y tienen que recorrer un camino más largo hasta un óptimo local.

Se nos puede ocurrir también comprobar si el número de pasos (o el tiempo) que necesitan los métodos ordenado y aleatorio son parecidos. Para ello podemos asumir que su distribución es normal y realizar un test de hipótesis para ver si la media de las distribuciones es idéntica. Esto se puede hacer con un test de t-Student para pares de muestras (lo podéis calcular con cualquier paquete estadístico). Si hacemos este test para el número de pasos usando una confianza del 0,95 obtenemos:

```
data: pasosOrdenado and pasosAleatorio
t = 0.3313, df = 9, p-value = 0.748
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
-2.563969  3.443969
sample estimates:
mean of the differences:  0.44
```

Un valor de probabilidad de 0,748 nos dice que es bastante probable que la diferencia entre las medias de las muestras sea 0. Por lo tanto no hay diferencia significativa entre los pasos que necesitamos para llegar a un óptimo con ambas inicializaciones.

6.5.2. TSP: Influencia del tamaño del problema

También nos puede interesar tener una idea del aumento del coste de la búsqueda en función del tamaño del problema. Podemos experimentar con el TSP usando números de ciudades crecientes para hacernos una idea del aumento del tiempo de cálculo. Podemos asumir que el crecimiento del tiempo estará en función del factor de ramificación y del tamaño del espacio de búsqueda, en este caso en esta implementación del problema el factor de ramificación es cuadrático en el número de ciudades (intercambios de todos los pares de ciudad) y el tamaño del espacio de búsqueda es $O(nciudades!)$.

Aquí asumiremos que el tiempo para los problemas de un mismo tamaño es comparable, aunque dependerá mucho de la solución inicial. Haremos varias ejecuciones para cada tamaño y usaremos los valores medios para ajustar la función del tiempo.

Podemos resumir el experimento en:

Observación	El tiempo sigue una función creciente respecto al tamaño del problema
Planteamiento	Escogemos diferentes tamaños de problema y observamos sus tiempos de ejecución
Hipótesis	La función será al menos cuadrática respecto al tamaño del problema
Método	<ul style="list-style-type: none"> ■ Elegiremos 10 semillas aleatorias para cada tamaño del problema ■ Ejecutaremos 10 experimentos para cada tamaño y haremos medias de los resultados ■ Experimentaremos con problemas de 25, 50, 75 y 100 ciudades ■ Usaremos el algoritmo de Hill Climbing con inicialización avariciosa ■ Mediremos el tiempo de cómputo para realizar la comparación

Si hacemos el experimento podemos tener unos resultados distribuidos como los de la figura 6.5. En la siguiente tabla podemos ver la media y desviación del tiempo de ejecución (ms):

Tamaño	25	50	75	100
Media	13,8	98,2	245,7	1196,9
Desviación	5,07	33,34	89,49	807,66

En la figura 6.6 podemos ver el crecimiento del tiempo en función del tamaño. La línea marcada como 1 es el tiempo medio obtenido de los experimentos. La línea marcada como 2 sería el tiempo suponiendo

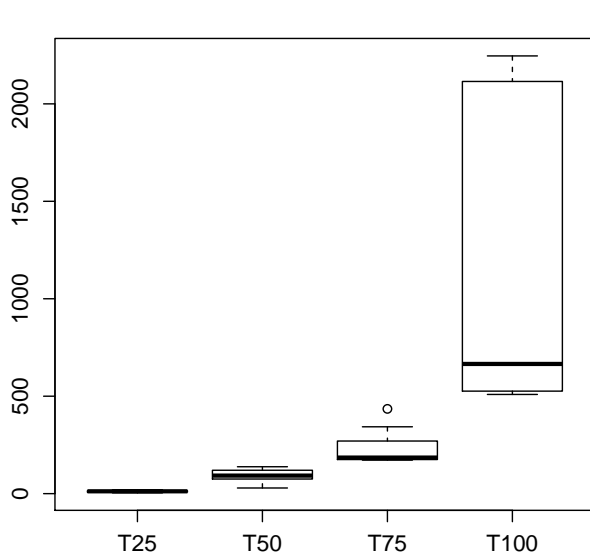


Figura 6.5: Distribucion del tiempo (ms) para cada tamaño del problema

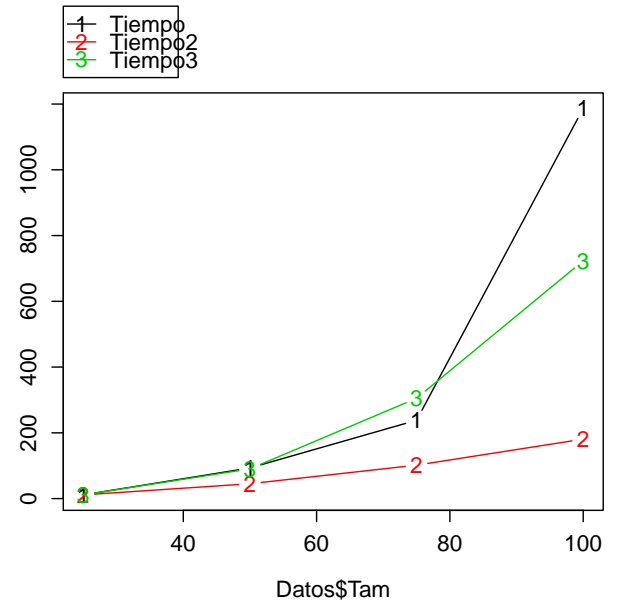


Figura 6.6: Función del tiempo (ms) respecto al tamaño del problema

que crece como una función cuadrática del tamaño del problema. La línea marcada como 3 sería el tiempo suponiendo que crece como una función cúbica del tamaño del problema. Una función que creciera como un polinomio de grado cuatro superaría bastante el crecimiento observado. Sin tomar más medidas es difícil de estimar el tipo de función que sigue, pero podríamos intuir que el crecimiento del tiempo de solución se ve influido tanto por el factor de ramificación, como por el tamaño del espacio de búsqueda, ya que este tiene un crecimiento mayor que cuadrático.

6.6. El Simulated Annealing

Una tarea que hay que realizar en la practica es encontrar los parámetros adecuados para ejecutar el Simulated Annealing. Como se explicó en laboratorio, la implementación que tenéis del algoritmo tiene los siguientes parámetros:

- Número total de iteraciones
- Iteraciones por cada cambio de temperatura (ha de ser un divisor del anterior)
- Parámetro k de la función de aceptación de estados
- Parámetro λ de la función de aceptación de estados

El número total de iteraciones que ha de hacer el algoritmo antes de parar se ha de determinar experimentalmente y, aparte de depender del problema, depende de los valores que se han usado para k y λ como veremos.

El número de pasos por cada cambio de temperatura divide las iteraciones en conjuntos de pasos en los que las condiciones de aceptación son iguales. Dependiendo del problema puede tener más o menos efecto en el resultado.

Para determinar como ajustar los parámetros k y λ primero tenemos que estudiar como es la función que se utiliza para la aceptación de estados. La función para el cálculo de la temperatura del sistema es la siguiente:

$$\mathcal{F}(T) = k \cdot e^{-\lambda \cdot T}$$

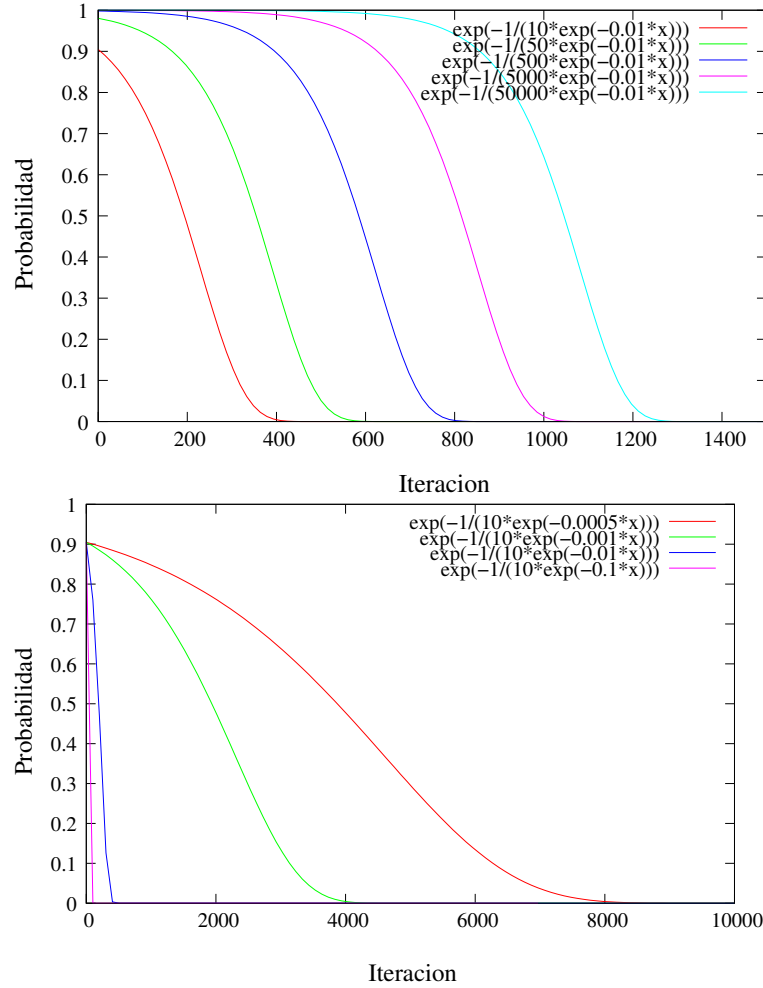


Figura 6.7: Variación del comportamiento variando K y λ

Y la función que determina la probabilidad de aceptación de un estado peor es:

$$P(\text{estado}) = e^{\left(\frac{\Delta E}{F(T)}\right)}$$

Esta es una función en el rango $[0, 1]$, que va variando con la temperatura (T), que en esta implementación es el número de iteración actual, y la diferencia de energía (ΔE) entre el estado actual y el siguiente estado.

Estudiando el comportamiento de la función podemos ver que cuanto mayor es k , más tarda en comenzar a decrecer y cuanto mayor es λ mas rápido descende. Esto lo podemos ver en la figura 6.7, donde para un ΔE de una unidad, podemos ver como varía la función cambiando uno de los parámetros y fijando el otro.

Podemos ver que si vamos variando k , a medida que aumenta, el número de iteraciones en las que la probabilidad de aceptar estados peores es alta va aumentando. Si vamos variando λ , la velocidad a la que esa probabilidad descende va aumentando cuanto mayor es, disminuyendo además el número de iteraciones que hacen falta para llegar a probabilidad 0.

Dado que el comportamiento de la función no varía con el problema, tendremos que estudiar para un problema concreto cuantas iteraciones harán falta para que la exploración pueda sobrepasar el punto en el que la probabilidad se hace 0, que es el momento en el que el algoritmo solo puede mejorar la solución actual.

Si el número de iteraciones que se usa no es suficiente, el algoritmo acabará en una zona de la función en la que aún estamos admitiendo estados peores, por lo que tenemos que asegurarnos que esto no suceda. Tampoco hemos de usar exactamente el punto en el que la función se hace 0, ya que hace falta que el algoritmo tenga un periodo en el que solo se pueda mejorar la solución.

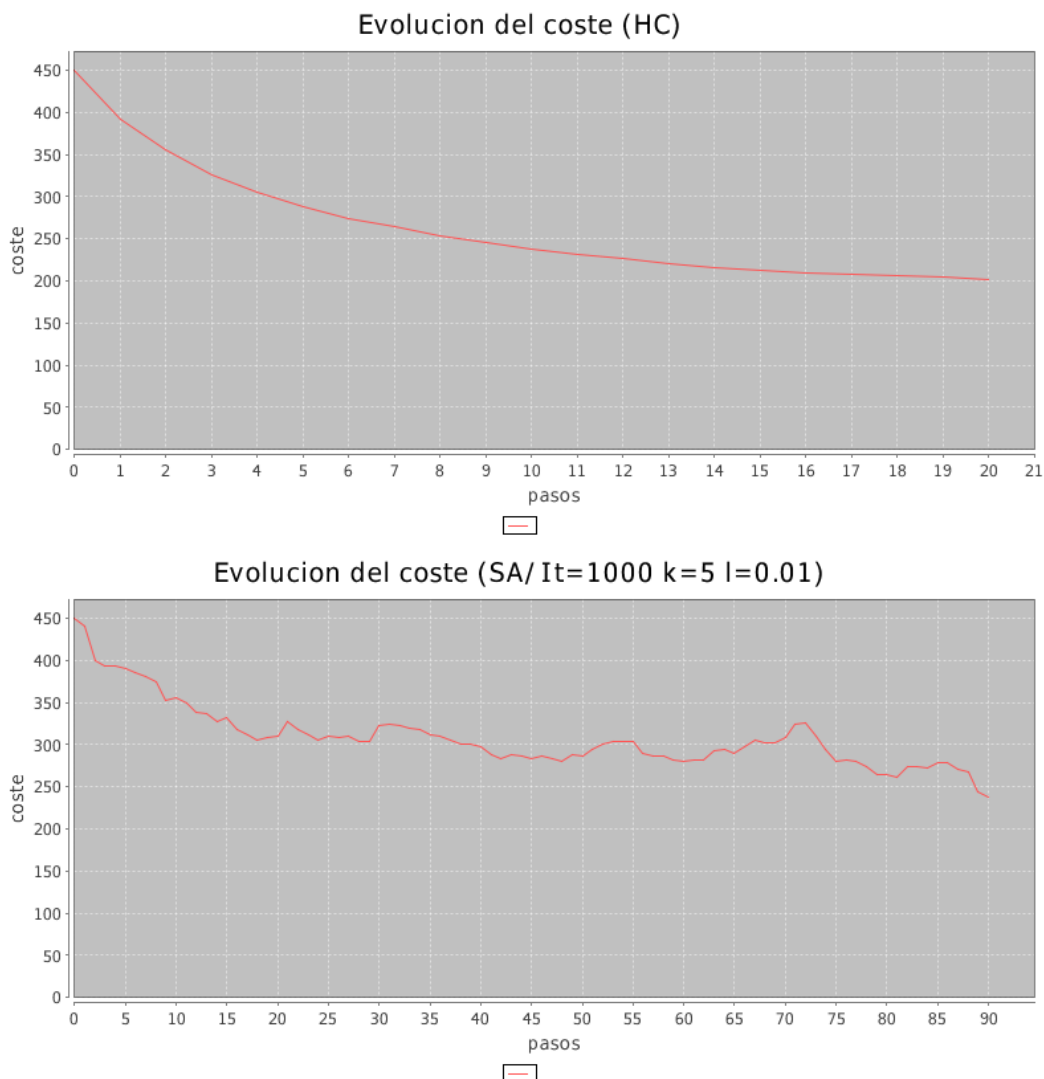


Figura 6.8: Variación del coste para HC y SA

6.6.1. Ejemplo

Vamos a utilizar el problema del viajante de comercio para ajustar los parámetros del Simulated Annealing de manera que se encuentren mejores soluciones que para el Hill Climbing.

Vamos a hacer el experimento para una instancia particular del problema, en este caso fijaremos el número de ciudades a 35¹ y trabajaremos con instancias generadas al azar.

En la figura 6.8 se puede ver la evolución del coste del problema para una instancia particular usando Hill Climbing y Simulated Annealing con 1000 iteraciones, un valor de k de 5 y un valor de λ de 0,01.

Se puede ver claramente el comportamiento de los dos algoritmos en la gráfica, el coste con Hill Climbing va disminuyendo de manera monótona y, en cambio, con Simulated Annealing el coste va subiendo y bajando. Se puede apreciar también que a pesar de que el coste vaya descendiendo en el Simulated Annealing, el coste va subiendo incluso casi al final. Esto puede indicar un número de iteraciones insuficiente. Si por ejemplo las aumentamos a 10000, podemos observar (figura 6.9) como el número de iteraciones efectivas que ha realizado ha sido mayor y que la ultima parte de la gráfica solo descende. También podemos fijarnos en que con los primeros parámetros del simulated annealing, la solución es peor que la del Hill Climbing, pero no con los segundos.

Queda evidente pues la importancia de ajustar correctamente los valores de estos parámetros.

La mayor dificultad para ajustarlos es que la única indicación que tenemos son los resultados que pode-

¹Se ha escogido este número sin ninguna razón particular, pero hay que tener en cuenta que es posible que los parámetros específicos del algoritmo puedan depender también del tamaño del problema.

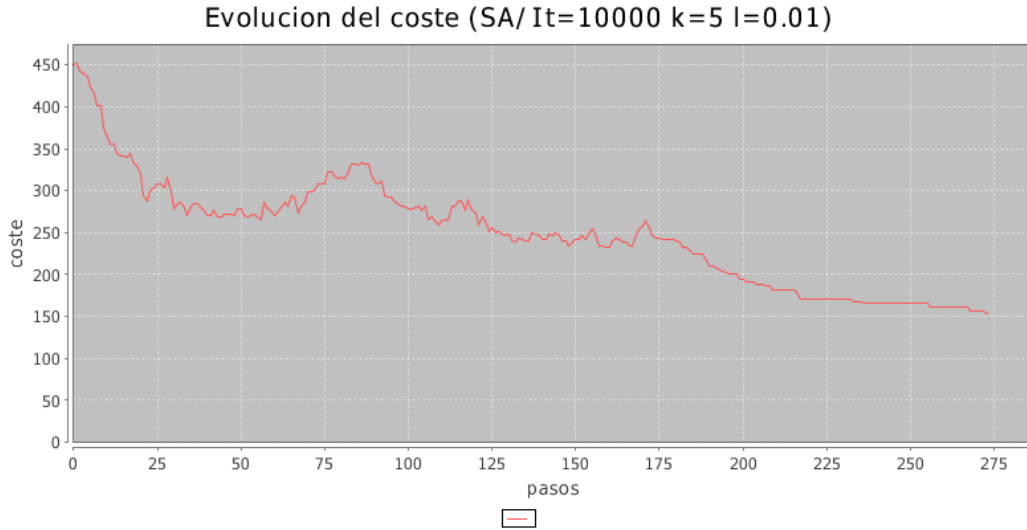


Figura 6.9: Variación del coste para SA con más iteraciones

mos obtener experimentalmente, eso nos obligará hacer pruebas exhaustivas para ver como se comporta el problema.

Una buena práctica para determinar el número de iteraciones mínimo para unos valores concretos de k y λ es averiguar cuál es la iteración en la que la probabilidad de aceptación se hace 0, sabemos que a partir de esa iteración el algoritmo solo acepta mejores soluciones, así que para hallar una buena solución hemos de usar un valor mayor. El cuanto mayor se deberá experimentar.

Los valores concretos de k y λ hacen que el algoritmo se salte más o menos soluciones cercanas al estado inicial y es lo que permite que se pueda llegar más allá que con el Hill Climbing. Este número de estados a descartar depende de la forma que tiene la función heurística, que en principio desconocemos. Es posible que tengamos una función que al principio tenga muchos óptimos locales y después mejore monótonamente o que tenga subidas y bajadas alternadas durante todo su recorrido. Eso nos obligará a experimentar con diferentes comportamientos para la función de aceptación.

Por ejemplo, podemos ir de un extremo en el que los parámetros permitan que se esté mucho tiempo aceptando estados peores y se permitan escoger estados muy malos, al otro extremo en el que siempre se acepten estados no demasiado malos y solo durante un periodo corto. Entre estos tenemos todo un rango de posibilidades.

La única manera de averiguar los parámetros correctos sería usar una exploración exhaustiva, pero podemos usar esos valores extremos para ir acotando un conjunto de parámetros que obtenga buenas soluciones. En el caso de la práctica sería el obtener soluciones mejores que las que nos daría el Hill Climbing.

Por ejemplo, en la figura 6.10 podemos ver la variación del coste usando diferentes valores de k con los que obtenemos comportamientos en los que somos más o menos exigentes con el coste del estado siguiente. Podemos ver también como el número de iteraciones necesarias para converger va aumentando, ya que al aumentar k estamos alejando el punto en el que la probabilidad de aceptación se hace 0.

Finalmente, en la figura 6.11 tenemos la variación del coste cambiando el valor de λ y fijando k . La disminución de este parámetro hace que se alargue el periodo en el que se aceptan estados peores, alargando el número de iteraciones necesarias para converger, y haciendo que la diferencia entre estados sucesivos vaya disminuyendo con el tiempo.

Vamos a hacer una serie de experimentos para averiguar cuales serían los mejores valores de k y λ . Para ello usaremos un número de iteraciones grande para asegurarnos de que lleguemos a converger y probaremos los valores para $k = \{1, 5, 25, 125\}$ y para $\lambda = \{1, 0,01, 0,0001\}$. A partir de estos puntos podremos hacernos una idea de donde se encuentran los mejores valores y podremos hacer una exploración más dirigida a partir de ellos.

La gráfica 6.12 muestra la media de varias ejecuciones para todas las combinaciones de los valores de los parámetros. Se puede observar que el mejor coste medio obtenido es para los parámetros $k = 125$ y $\lambda = 0,0001$. Obtenemos un coste en media alrededor de 130, que mejora sustancialmente la solución obtenida

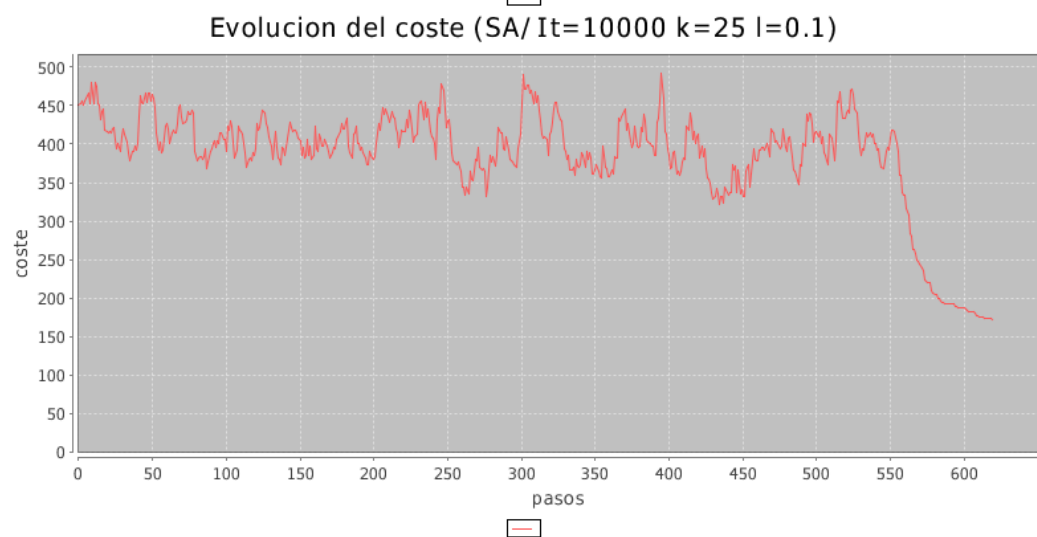
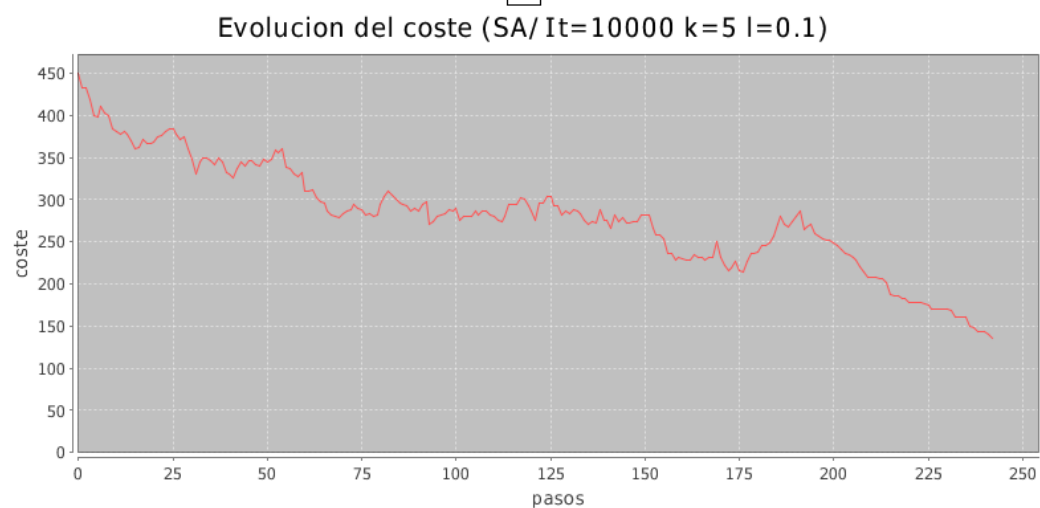
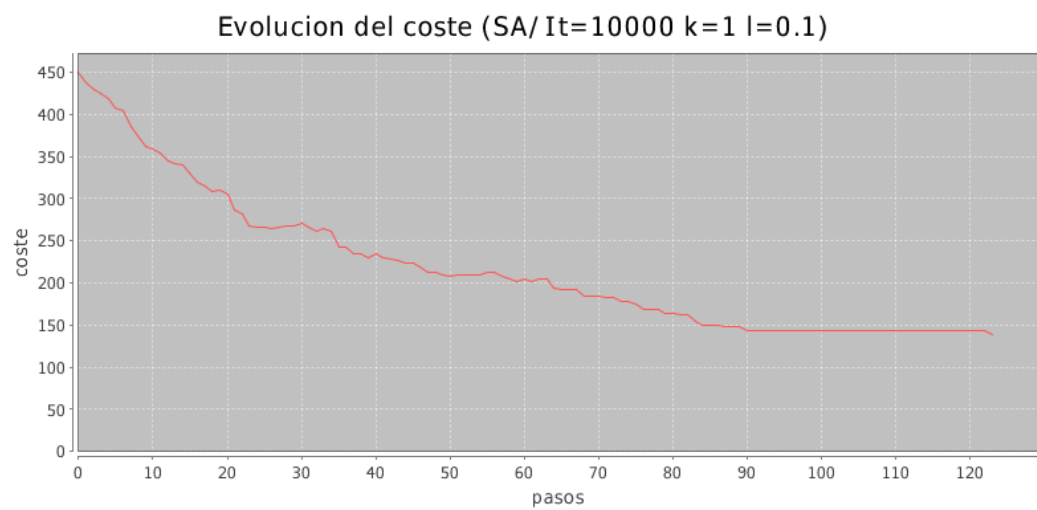


Figura 6.10: Variación del coste para SA variando k

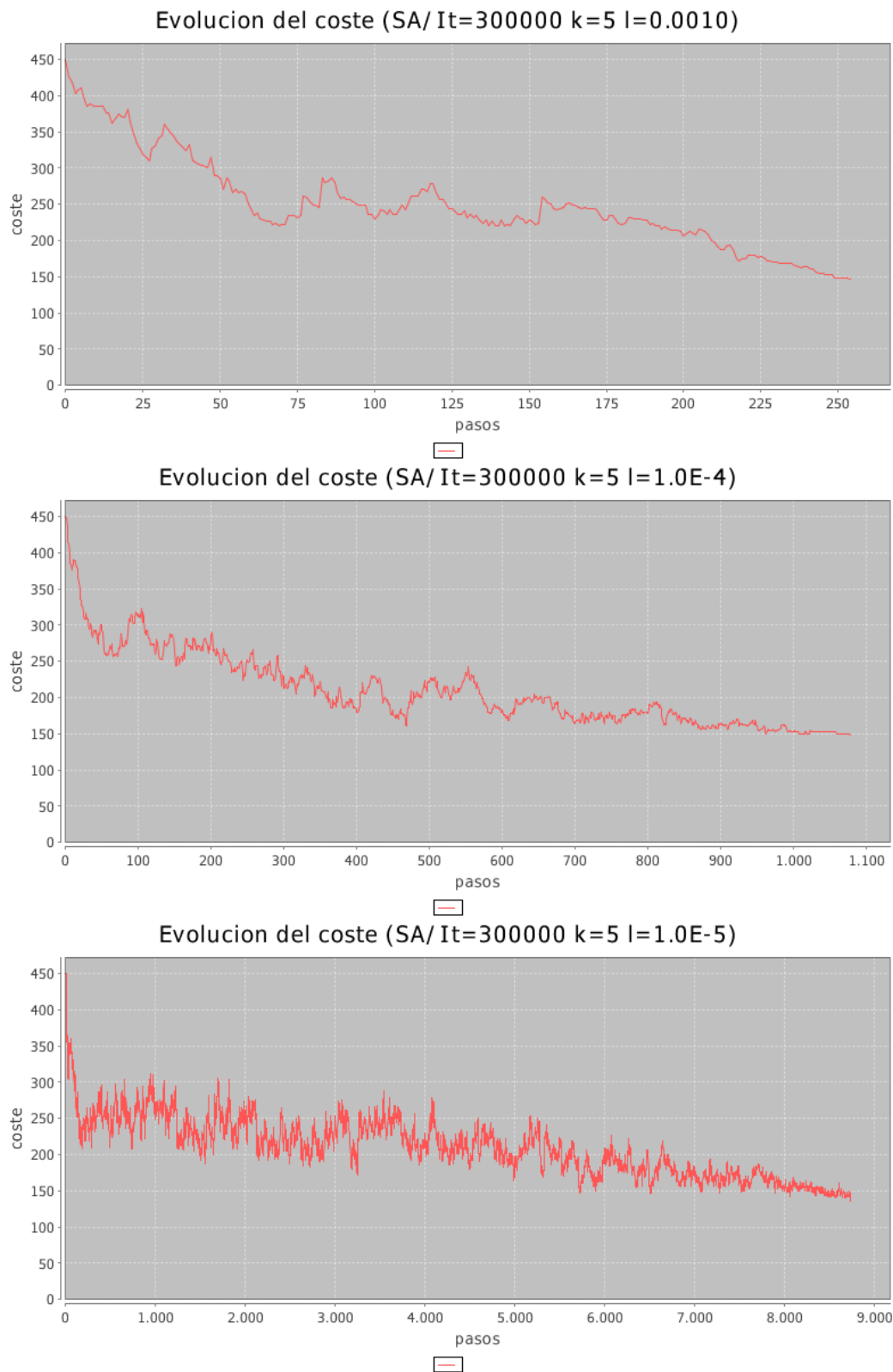


Figura 6.11: Variación del coste para SA variando λ

Experimentos K y Lambda

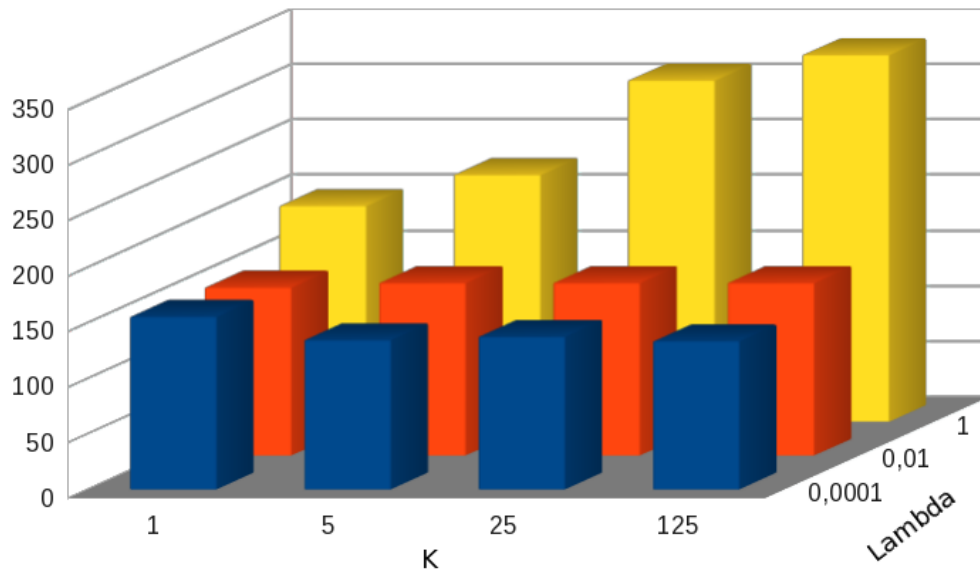


Figura 6.12: Variación del coste para SA para varios valores de k y λ

con Hill Climbing que está en 200.

Esto nos indicaría que deberíamos explorar más alrededor de esos valores. De todas formas también tendría sentido explorar diferentes valores de k para la λ que da el menor valor, ya que el número de iteraciones necesarias para converger es relativamente grande.

La gráfica 6.13 es la evolución del coste para una ejecución del problema con estos parámetros.

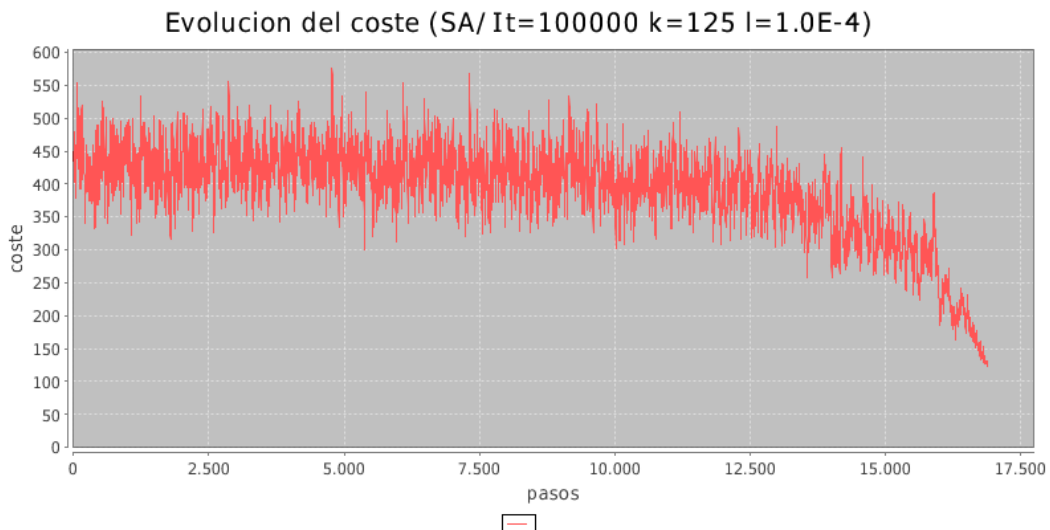


Figura 6.13: Variación del coste para SA para el problema con $k = 125$ y $\lambda = 0,0001$

Rúbrica de evaluación

Esta es la rúbrica de evaluación de la práctica. La corrección se hará según estos criterios y siguiendo las pautas que se detallan para cada nivel de evaluación.

Deberéis seguir estos criterios a la hora de escribir vuestra documentación y explicar qué habéis hecho en el desarrollo de la práctica y como lo habéis hecho.

Valoración			
Criterio	Mal	Regular	Bien
Parte descriptiva Identificación del problema	<ul style="list-style-type: none"> La descripción del problema es una copia de la descripción que hay en el enunciado. No hay una descripción de los elementos del estado del problema. No se justifica porqué es un problema de búsqueda local 	<ul style="list-style-type: none"> La descripción del problema es una copia de la descripción que hay en el enunciado. Identificación y análisis breve de las características del problema. Hay una descripción breve de los elementos del estado del problema. Se justifica porqué es un problema de búsqueda local 	<ul style="list-style-type: none"> La descripción del problema es algo más elaborada que la que ya hay en el enunciado. Identificación y análisis detallado de las características del problema. Hay una descripción detallada los elementos del estado del problema. Se justifica porqué es un problema de búsqueda local
Estado del problema y representación	<ul style="list-style-type: none"> Hay una descripción/justificación poco detallada de la representación del problema. La representación del problema es inadecuada y/o ineficiente. No hay una análisis del tamaño del espacio de búsqueda. 	<ul style="list-style-type: none"> Hay una descripción/justificación poco detallada de la representación del problema. La representación del problema es adecuada y eficiente. Hay un análisis del tamaño del espacio de búsqueda. 	<ul style="list-style-type: none"> Hay una descripción/justificación detallada de la representación del problema. La representación del problema es adecuada y eficiente. Hay un análisis del tamaño del espacio de búsqueda.
Representación y análisis de los operadores	<ul style="list-style-type: none"> No hay una descripción de los operadores (sus condiciones de aplicabilidad y sus efectos). No hay un análisis del factor de ramificación de los operadores. No hay una explicación de la elección de los operadores. 	<ul style="list-style-type: none"> Hay una descripción breve de los operadores (sus condiciones de aplicabilidad y sus efectos). Hay un análisis del factor de ramificación de los operadores. No hay una explicación de la elección de los operadores. 	<ul style="list-style-type: none"> Hay una descripción detallada de los operadores (sus condiciones de aplicabilidad y sus efectos). Hay un análisis del factor de ramificación de los operadores. Hay una explicación de la elección de los operadores.

Valoración			
Criterio	Mal	Regular	Bien
Análisis de la función heurística	<ul style="list-style-type: none"> No hay una explicación/análisis de los factores que intervienen en la heurística del problema. No hay una justificación de las funciones heurísticas escogidas. No hay una explicación de los efectos de las funciones heurísticas en la búsqueda. No hay una justificación de las ponderaciones que aparecen entre los elementos de las heurísticas. 	<ul style="list-style-type: none"> No hay una explicación/análisis de los factores que intervienen en la heurística del problema. Hay una justificación de las funciones heurísticas escogidas. Hay una explicación de los efectos de las funciones heurísticas en la búsqueda. No hay una justificación de las ponderaciones que aparecen entre los elementos de las heurísticas. 	<ul style="list-style-type: none"> Hay una explicación/análisis de los factores que intervienen en la heurística del problema. Hay una justificación de las funciones heurísticas escogidas. Hay una explicación de los efectos de las funciones heurísticas en la búsqueda. Hay una justificación de las ponderaciones que aparecen entre los elementos de las heurísticas.
	<ul style="list-style-type: none"> No hay una descripción del algoritmo para calcular las soluciones iniciales. No hay una explicación/justificación de las elecciones de solución inicial (bondad de la solución, coste de hallar la solución). 		<ul style="list-style-type: none"> Hay una descripción del algoritmo para calcular las soluciones iniciales. Hay una explicación/justificación de las elecciones de solución inicial (bondad de la solución, coste de hallar la solución).
Elección y generación del estado inicial			

Valoración		Mal	Regular	Bien
Criterio Parte experimental Sobre los experimentos en general		<ul style="list-style-type: none"> No son informativos y/o solo se ha hecho una ejecución de cada experimento. No están encaminados a destacar las diferencias entre los elementos comparados. No se ha escogido un criterio uniforme para comparar los experimentos. No se explican, ni analizan los resultados (tablas, estadísticas, gráficos). No se compara lo que se esperaba con lo obtenido. 	<ul style="list-style-type: none"> Son informativos y se han hecho varias ejecuciones de cada experimento. Están encaminados a destacar las diferencias entre los elementos comparados. No se explican, ni analizan los resultados (tablas, estadísticas, gráficos) o el análisis no tiene mucho sentido. No se compara lo que se esperaba con lo obtenido. 	<ul style="list-style-type: none"> Son informativos y se han hecho varias ejecuciones de cada experimento. Están encaminados a destacar las diferencias entre los elementos comparados. Se explican y analizan correctamente los resultados (al menos con tablas, mejor con estadísticas y gráficos). Se compara lo que se esperaba con lo obtenido.
	Influencia de la solución inicial	<ul style="list-style-type: none"> No se muestra claramente la diferencia entre las soluciones iniciales propuestas. 		<ul style="list-style-type: none"> Se muestra claramente la diferencia entre las soluciones iniciales propuestas: <ul style="list-style-type: none"> En función del coste temporal de la búsqueda y el ahorro justificable por partir desde cada solución inicial. En función de la bondad de las soluciones comparando si partir de mejores soluciones iniciales lleva a mejores soluciones finales.
	Influencia de los operadores	<ul style="list-style-type: none"> No se muestra claramente la diferencia entre los conjuntos de operadores. 		<ul style="list-style-type: none"> Se muestra claramente la diferencia entre los conjuntos de operadores: <ul style="list-style-type: none"> En función del coste temporal de la búsqueda. En función de la bondad de las soluciones.

Valoración		Mal	Regular	Bien
Criterio	Influencia de la función heurística	<ul style="list-style-type: none"> ■ No se muestra claramente la diferencia entre las funciones heurísticas escogidas. ■ No se experimentan las ponderaciones de los criterios de las funciones heurísticas. 		<ul style="list-style-type: none"> ■ Se muestra claramente la diferencia entre las funciones heurísticas: <ul style="list-style-type: none"> • En función del coste temporal de la búsqueda. • En función de la bondad de las soluciones. ■ Se realizan experimentos para explorar la influencia de diferentes ponderaciones para los criterios de las funciones heurísticas.
	Comparación de algoritmos	<ul style="list-style-type: none"> ■ No se hace un ajuste de los parámetros del Simulated Annealing. ■ No se hace una comparación de los dos algoritmos de búsqueda en función de los experimentos. 	<ul style="list-style-type: none"> ■ Se ajustan los parámetros del Simulated Annealing pero sin método alguno. ■ Se hace una comparación pobre de los algoritmos de búsqueda (HC,SA) sin ningún criterio definido. 	<ul style="list-style-type: none"> ■ Se ajustan los parámetros del Simulated Annealing explorando sistemáticamente los valores de sus parámetros. ■ Se hace una comparación de los algoritmos de búsqueda en función de: <ul style="list-style-type: none"> • El coste temporal de la búsqueda. • La bondad de las soluciones.
Escenarios propuestos y preguntas del enunciado		<ul style="list-style-type: none"> ■ No se experimentan los escenarios planteados en el enunciado. ■ No se responden a la preguntas planteadas en el enunciado. 	<ul style="list-style-type: none"> ■ Se hacen los experimentos planteados en el enunciado pero las explicaciones sobre los resultados son breves y poco justificadas. ■ Se responden a las preguntas del enunciado pero sin apoyar las explicaciones en los resultados de los experimentos. 	<ul style="list-style-type: none"> ■ Se hacen los experimentos planteados en el enunciado, las explicaciones sobre los resultados son adecuadas y razonadas. ■ Las explicaciones comparan/justifican los resultados que se esperaban con el escenario y los resultados obtenidos con los experimentos. ■ Se responden a las preguntas del enunciado apoyando las explicaciones con los resultados de los experimentos y/o realizando experimentos adicionales.

Valoración		Mal	Regular	Bien
Criterio Calidad de la práctica Documentación		<ul style="list-style-type: none"> La práctica tiene una mala presentación. El documento esta mal estructurado y/o es ilegible. La documentación no demuestra una buena planificación del trabajo. La documentación no refleja el esfuerzo realizado. 		<ul style="list-style-type: none"> La práctica tiene una presentación adecuada. El documento está estructurado según los puntos que indica la práctica. La explicaciones están bien elaboradas y son inteligibles. La documentación demuestra una buena planificación del trabajo y un seguimiento del guión de la práctica. La documentación refleja el esfuerzo realizado.
	Cumplimiento de lo que pide el enunciado	<ul style="list-style-type: none"> Se han simplificado elementos del problema No se han tenido en cuenta todas las restricciones del problema No se han seguido las recomendaciones/indicaciones para la implementación de la práctica 		<ul style="list-style-type: none"> La práctica se ha realizado acorde a lo que pedía el enunciado

Valoración		Mal	Regular	Bien
Criterio	Elaboración de informes de prácticas			
Ortografía y gramática	<ul style="list-style-type: none">El documento está plagado de errores de ortografía y gramaticales. Desde este punto de vista, es un documento impresentable.	<ul style="list-style-type: none">El documento tiene pocas faltas de ortografía y gramaticales.	<ul style="list-style-type: none">El documento no tiene faltas de ortografía, ni errores gramaticales.	
Organización	<ul style="list-style-type: none">El documento está muy mal organizado.No se introduce bien el tema. El desarrollo no sigue una línea coherente. Se pasa de un aspecto a otro sin un orden. No hay un resumen y/o conclusiones.	<ul style="list-style-type: none">Hay algún aspecto claramente mejorable.La introducción no acaba de centrar bien el tema, o los diferentes apartados no acaban de estar bien ligados en una secuencia lógica, o falta un buen cierre con resumen y/o conclusiones.	<ul style="list-style-type: none">El documento está organizado de forma lógica.Las diferentes secciones y subsecciones están bien ligadas, y facilitan el seguimiento del contenido.El documento en una primera parte plantea el problema, en una segunda desarrolla y elabora la solución y en una tercera resume los resultados y plantea las conclusiones.	
Claridad	<ul style="list-style-type: none">El texto es muy difícil de entender; las frases son largas y confusas; constantemente se tienen que releer partes del texto para entenderlas y en varios casos finalmente no se entiende lo que se quiere decir.	<ul style="list-style-type: none">En alguna ocasión hay alguna frase larga y confusa que se tiene que releer varias veces para acabar de entender.	<ul style="list-style-type: none">Los contenidos son muy claros.Las frases son cortas y fáciles de entender a la primera.	
Uso de gráficos	<ul style="list-style-type: none">No se utilizan ni los gráficos, ni las figuras necesarias para facilitar la comprensión, o los que se utilizan no aclaran nada.	<ul style="list-style-type: none">En algún punto del documento se echa en falta algún gráfico o figura que ayude a aclarar los conceptos.Alguna de las figuras o gráficos parece poco clarificadora o innecesaria.	<ul style="list-style-type: none">Todos los gráficos y figuras utilizados tienen sentido y ayudan a entender la explicación.No hay ninguno que sobre, ni tampoco se echa ninguno en falta.	

Competencia de trabajo en equipo

8.1. Competencia de trabajo en equipo: práctica de búsqueda

Con esta práctica se evaluará la competencia de trabajo en equipo. Esto significa utilizar la planificación propuesta para ayudaros a planificar el trabajo y establecer una pautas entre el grupo de trabajo para desarrollar la práctica de manera eficaz.

Estas pautas incluyen el repartir el trabajo, establecer un calendario de plazos de entregas, realizar reuniones periódicas, establecer mecanismos de comunicación y participación entre los miembros del equipo y detectar y resolver conflictos.

No es trabajo en equipo el repartirse el trabajo la primera semana y no verse hasta la semana anterior a la entrega.

El formulario de evaluación que utilizaremos para la competencia es el que tenéis a continuación.

Treball en equip			
Nivell 2: Contribuir a consolidar l'equip, planificant objectius, treballant amb eficàcia i afavorint-hi la comunicació, la distribució de tasques i la cohesió.	Alumnes		
Establir i mantenir les relacions cooperatives, i identificar i marcar pautes per superar-ne les dificultats	1	2	3
Ajuda en la creació de les pautes de treball per a resoldre els conflictes en l'equip			
Té en compte els punts de vista dels altres i retroalimenta de manera constructiva			
Actua constructivament per afrontar els conflictes interns de l'equip			
Establir i planificar els objectius del grup, les responsabilitats i les tasques que s'han de dur a terme	1	2	3
Contribueix a l'establiment dels objectius de l'equip			
Assumeix els objectius de l'equip com a propis			
Proposa un repartiment igualitari de tasques i responsabilitats			
Treballar amb eficàcia i crear oportunitats per motivar la participació dels altres	1	2	3
Compleix els terminis definits en la planificació del treball			
Busca la participació de totes les persones en els debats i resultats			
Intercanviar informació, aportar idees i modificar les propostes de treball	1	2	3
Participa de forma activa en els moments de trobada, compartint la informació, els coneixements i les experiències			
Aporta idees i propostes amb afany de millora			
Accepta i integra la crítica constructiva per a la millora del seu propi treball			

8.2. Competencia de trabajo en equipo: Trabajo de innovación

La competencia de trabajo en equipo también se evalúa con el trabajo de innovación que debéis desarrollar durante todo el cuatrimestre (y no una semana antes de entregarlo). Por ello, adjunto al informe de la práctica también tendréis entregar un informe sobre lo que habéis hecho hasta ese momento.

Pensad que a estas alturas ya deberíais haber decidido el tema del trabajo, haber hecho una planificación/división del trabajo y haber recolectado información para los diferentes puntos del informe del trabajo de innovación.

El documento que deberéis entregar ha de incluir:

- 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