

Examen Parcial de IA

(8 de noviembre de 2010)

grupo 10

Duración: 1 hora

1. (6 puntos) La International Telecommunications Union (ITU) es un agencia de Naciones Unidas que regula el uso de las tecnologías de la telecomunicación y entre otras cosas coopera en la asignación de órbitas para satélites. El incremento del uso de satélites de todo tipo ha llevado a buscar métodos automáticos para asignar posiciones en órbitas que mantengan la seguridad de los satélites y la calidad de su funcionamiento.

Para solucionar el problema, se ha dividido la órbita geosincrónica en regiones donde se asignarán las posiciones orbitales de los satélites. Obviamente, una región del espacio tiene un volumen. Se quiere resolver el problema para cada región.

Hay S diversos tipos de satélites especializados que se desean poner en órbita (televisión, telefonía, científicos, militares, ...). Para cada satélite se tienen dos informaciones, el espacio de seguridad (volumen necesario para que el satélite pueda navegar con el mínimo riesgo de colisión y pueda alimentar sus paneles solares) y el coste de alquiler de la órbita que cobra la ITU al dueño del satélite (este precio depende de cada satélite y no es una función del espacio de seguridad).

Las restricciones que tiene la ITU son que la suma del volumen que han de ocupar los espacios de seguridad de los satélites ha de cubrir como mínimo $1/3$ del volumen de la región si se quieren cubrir las necesidades existentes, pero no puede superar los $2/3$ si no se quieren tener problemas con otros lanzamientos que deben atravesar la órbita geosincrónica o con la basura espacial. También se ha de cubrir un cupo mínimo m_i para cada tipo de satélites. El objetivo del problema es maximizar la ganancia que obtiene la ITU con el alquiler de órbitas.

En los siguientes apartados se proponen diferentes alternativas para algunos de los elementos necesarios para plantear la búsqueda (solución inicial, operadores, función heurística, ...). El objetivo es comentar la solución que se propone respecto a si es correcta, es eficiente, o es mejor o peor respecto a otras alternativas posibles. Justifica tu respuesta.

- a) Se plantea usar Hill-climbing, como solución inicial consideramos que asignamos aleatoriamente al menos la mínima cantidad de satélites para cubrir la cuota de cada tipo hasta ocupar $1/3$ del volumen de la región. Disponemos del operador `añadir-satélite()` que comprueba el posible exceso de ocupación en la región y del operador `quitar-satélite()` que comprueba la posible infra-ocupación de la región. Como heurística usamos la suma para todos los satélites en la solución del producto entre el espacio de seguridad del satélite y el coste de alquiler del satélite.
- b) Se plantea usar Hill-climbing, como solución inicial ordenamos los satélites descendientemente por el coste del alquiler y vamos añadiendo los satélites en ese orden hasta llenar $2/3$ del volumen. Como operador utilizamos cambiar un satélite de un tipo por otro del mismo tipo que no esté ya en la solución. Como función heurística usamos la suma del espacio de seguridad de todos los satélites en la solución multiplicado por la suma del coste del alquiler de todos los satélites.
- c) Se plantea usar algoritmos genéticos. Un individuo es una tira de bits cuya longitud es el número total de satélites que tenemos. Si el bit está a 1 significa que el satélite está en la solución y si está a 0 es que no lo está. La población inicial se genera creando individuos que tengan a 1 los bits de los m_i satélites con mayor coste de alquiler de cada tipo. Como operadores se usan los habituales de cruce y mutación. Como función heurística usamos la suma del coste de alquiler de todos los satélites en la solución.

2. (4 puntos) Queremos planificar cómo debemos integrar diferentes procesos individuales de fabricación de diversos productos en un único proceso. Cada producto necesita utilizar un conjunto

de máquinas en un orden específico que se puede describir mediante una secuencia de proceso. El uso de una máquina es un paso del proceso de fabricación de un producto. Para simplificar supondremos que cada paso tiene la misma duración. Hay que tener en cuenta que un producto puede necesitar un mismo tipo de máquina en diferentes pasos de su proceso de fabricación.

Disponemos de un número fijo de máquinas de cada tipo (m_i). Supondremos que necesitamos fabricar una unidad de cada producto y solo podemos volver a iniciar el proceso una vez que hemos acabado todos los productos, es decir, en el proceso integrado sólo se fabrica una unidad de cada producto a la vez. Obviamente, un paso del proceso integrado se compone de un paso de todos o un subconjunto de los procesos individuales. Queremos minimizar el número de pasos del proceso integrado.

Se nos plantean las siguientes alternativas:

- a) Queremos utilizar A^* para minimizar el número total de pasos del proceso integrado. El estado lo forman los pasos de los procesos individuales que hemos encajado en cada paso del proceso integrado. El operador de cambio de estado consiste en colocar el máximo número de procesos individuales (un paso de cada uno) en el paso actual del proceso integrado, el coste del operador es el número de pasos integrados. La función heurística es la suma de pasos de los procesos individuales que nos quedan por integrar.
- b) Queremos utilizar satisfacción de restricciones, para ello creamos un grafo de restricciones que tiene MAX variables para cada producto, considerando que este número es el máximo número de pasos que tendremos. El dominio de cada variable es la máquina que se ha asignado a ese paso. Para cada variable de un producto, ponemos como restricción que todas las variables anteriores en la secuencia deben respetar el orden de la secuencia de fabricación. Para las variables de todos los productos en un paso de fabricación, ponemos la restricción de que el número de máquinas asignadas a los productos en ese paso no supere los máximos de máquinas de los que se dispone.

Comenta cada una de las posibilidades indicando si resuelven o no el problema y qué ventajas e inconvenientes tiene cada una de ellas. Justifica la respuesta.

Examen Parcial de IA

(9 de noviembre de 2010)

grupo 30

Duración: 1 hora

1. (6 puntos) El comité local de la federación de ajedrez necesita establecer un calendario para la competición anual. En una competición hay C diferentes categorías dependiendo de la edad y del nivel de los jugadores. En cada categoría participan un número J_c de jugadores, lo cual significa un número R_c de rondas eliminatorias (obviamente $\log_2(J_c)$). La competición ha de durar un máximo de 15 días, para cada día de competición se disponen de F franjas horarias, en una franja solo se puede jugar una partida de ajedrez. Obviamente se han de jugar todas las partidas de la competición.

El calendario ha de cumplir un conjunto de restricciones:

- El número de días necesarios para realizar la competición ha de ser el mínimo posible.
- El conjunto de rondas correspondientes a cada categoría se ha de celebrar en el mínimo número de días posible.
- En un día se pueden jugar como máximo P partidas de una categoría.
- Entre dos partidas de una misma categoría en un mismo día debe haber como mínimo dos franjas horarias de separación y como máximo cinco.
- Debe haber como mínimo un día de separación entre dos rondas de una categoría.

En los siguientes apartados se proponen diferentes alternativas para algunos de los elementos necesarios para plantear la búsqueda (solución inicial, operadores, función heurística,...). Comenta muy brevemente la solución que se propone respecto a si es correcta, eficiente o mejor/peor respecto a otras alternativas posibles. Justifica tus respuestas.

- a) Usar Hill-climbing. Como solución inicial se utiliza el calendario vacío. Como operadores usamos asignar y desasignar una partida de una ronda de una categoría a una franja de un día. Como función heurística usamos el número de días total en los que hay partidas asignadas más la suma del número de días que ocupan las rondas de una categoría.
- b) Usar Hill-climbing. Como solución inicial se usa el calendario resultante de añadir de manera secuencial las competiciones de cada ronda de la primera categoría manteniendo una distancia de cinco franjas horarias entre cada partida, sin superar las P partidas diarias y manteniendo un día de separación entre cada ronda. Con el mismo procedimiento se van introduciendo las competiciones del resto de categorías. Como operadores utilizamos mover una partida a una franja no ocupada e intercambiar dos partidas. Como función heurística usamos el número de días total en los que hay partidas asignadas, más la suma del número de días que ocupan las rondas de una categoría, más una constante por el número de franjas que separan en un día las partidas de una misma ronda y categoría.
- c) Usar Algoritmos Genéticos. Asignamos a cada partida a jugar un código binario con la longitud necesaria para representarlas todas, añadimos a la codificación un código con todos los bits a cero. Representamos un calendario como una cadena de bits de longitud igual al producto de F franjas por 15 días por el número de bits necesarios para representar las partidas. Si una franja no tiene una partida asignada contendrá la codificación a cero, si no tendrá el código de una partida.

Para generar la población inicial usamos el método del segundo apartado cambiando el orden en el que escogemos las categorías. Como operadores genéticos usamos solamente el de cruce por un punto, donde el punto de cruce no puede estar en medio de la codificación de una partida. Como función heurística minimizamos el número de franjas vacías entre dos partidas consecutivas.

2. (4 puntos) Queremos transmitir X Kbits entre dos máquinas M_I y M_F , para ello hemos de establecer C canales de comunicación que han de atravesar N máquinas (M_1, M_2, \dots, M_N) que hacen de puntos intermedios ($C \leq N$), los X Kbits los dividimos a partes iguales por cada canal. Conocemos la velocidad en $kbps$ del canal que se puede establecer entre cada par de máquinas (incluidas M_I y M_F), queremos que la velocidad media de un canal sea como mínimo V Kbps y que un canal no atravesase más de P máquinas.

Buscamos la solución que conecte todas las máquinas sin que dos canales pasen por la misma máquina (exceptuando M_I y M_F) tardando lo menos posible en transmitir los X Kbits.

- a) Queremos utilizar A*. Definimos el estado como la asignación de máquinas a canales. El estado inicial consiste en asignar las máquinas M_I y M_F a cada canal. Tenemos un operador que asigna una máquina a un canal y cuyo coste es la velocidad media del canal que empieza en M_I , pasa por la máquina y acaba en M_F , solo podemos asignar una máquina a un canal si no excedemos el valor P . Como función heurística usamos la suma de las velocidades medias de todos los canales que se obtienen poniendo cada máquina que queda por conectar como único nodo intermedio entre M_I y M_F .
- b) Aplicar un algoritmo de satisfacción de restricciones. Las variables son las máquinas, los dominios son los canales donde podemos asignarlas. Supondremos que no tenemos en cuenta las máquinas M_I y M_F ya que estarán asignadas a todos los canales. Como restricciones imponemos que un canal no esté asignado a más de P máquinas y que la velocidad media mínima del canal que incluye las máquinas asignadas sea mayor que V .

Comenta cada una de las posibilidades indicando si resuelven o no el problema, qué errores te parece que tiene cada solución y cómo se podrían corregir, y qué ventajas e inconvenientes tienen cada una de ellas. Justifica la respuesta.

IA Midterm Exam

(November 10th 2010)

Time: 1 hour

1. (6 points) The owners of the “More than bugs” hotel want to reduce their cleaning budget and to maximize their earnings. In order to do this they have decided to have three different rent rates for the rooms. The rate A includes a daily cleaning of the room and has a price of ra euros per day, the rate B includes a cleaning on alternate days and has a price of rb euros per day and the rate C only includes a cleaning the first and the last day and has a price of rc euros per day. The minimum number of days that a room can be rented is two and we will assume that the cleaning days for rate B rooms are organized so the last cleaning of the room is the day the guests leave the room. We have a total of R rooms in the hotel.

To clean a room of rate A costs ca euros per cleaning day, a room of rate B costs cb euros per cleaning and a room of rate C costs ca plus the number of days of the reservation per cc . Rooms that are empty for a period of time are not cleaned.

The health department imposes the constraint that a room has to be cleaned at least 10 times in a month.

The hotel receives petitions for reservations for an entire month, each reservation includes the rate the client wants, the day the reservation begins and the number of days. We will assume that all petitions begin and end inside the month. The goal is to decide what reservations to actually book so the cleaning expenses are minimized and the earnings are maximized.

We want to solve this problem as a local search problem and, in the following sections, different alternatives are proposed for some of the necessary elements to start the search (initial state, operators, evaluation function...). You have to comment the proposed solution with respect to whether it is correct, it is efficient, or it is better or worse than other alternatives. Justify your comments.

- a) We want to use Hill-climbing, the initial solution is the empty solution. As search operators we use **add-reservation** that adds a reservation to a room and **change-reservation** that changes a reservation of a room for other not in the solution. The heuristic function is the sum of all the costs of cleaning the rooms multiplied by the gains obtained from the reservations.
 - b) We want to use Hill-climbing, the initial solution is obtained in the following way, for a room we pick one of the reservations that has the earlier date, the next reservation will be one of the reservations that have the earlier date after the end of the last assigned reservation, we repeat this procedure until we can not assign more reservations to the room. We do this for all the rooms. As search operators we use **change-reservation** that changes a reservation of a room to other not in the solution. The heuristic function is the sum of all the costs of cleaning the rooms minus the gains obtained from the reservations.
 - c) We want to use genetic algorithms, we decide to assign to each reservation a number from 0 to R , the concatenation of this numbers for all the reservations in binary is the code of a solution. The number 0 means that the reservation is not assigned to any room, other number means the room number assigned to the reservation. To generate the initial population we obtain a solution by randomly assigning one reservation to each room (only R reservations have a number different from 0). As genetic operators we use the usual crossover and mutation operators. The heuristic function is the sum for all the reservation in the solution of the quotient between the gains obtained from the reservation and the cleaning costs of the reservation.
2. (4 points) A small airport wants to organize the schedule of flights for a day. The airport can manage 10 flights each hour and the airport is open 15 hours a day. To populate the schedule the airport can pick flights from 10 different companies and each company offer flights to 10 different destinations.

The schedule can not have more than two flights of the same company at the same hour and two flights to the same destination by a specific company have to be at least six hours apart.

The money the airport makes by having a flight to a destination by a company depends on its demand. We want to maximize the money that the airport makes fulfilling the constraints that we have. We can solve the problem in two different ways:

- a)* By using A* defining as state the allocation of flights to the schedule. The initial state is the empty schedule, the final state is when the schedule is full. As search operator we use to assign a flight to a destination by a company to an specific hour of the schedule, the cost would be the money that the airport makes with this flight. As heuristic function we use the number of flights still to be assigned on the state.
- b)* By using a constraint satisfaction. We assume that a flight to a destination by a company only can be assigned at most to three different hours. We create a constraints graph that has as variables the three possible hours a flight to a destination by a company can be assigned. The domain of a variable is the hour the flight to a destination is assigned or empty if it is not assigned to any hour. There is a constraint among the variables for a flight to a destination by a company that forbids that the difference between the hours assigned is less than six and a constraint among all the destinations of a company that forbids that there are more than two flights by the company at the same hour.

Comment on each one of the possibilities about if they are or are not solutions to the problem, what errors you can find in each solution and how they should be corrected and the advantages or disadvantages for each one. Justify your answers.