Examen Parcial de IA

(10 de noviembre de 2008)
grupo 20
Duración: 1 hora

1. (6 puntos) La conocida cadena de supermercados Karpabo desea introducirse en una nueva ciudad y para ello ha de decidir qué locales son los más adecuados para instalar sus supermercados. Para ello disponemos de un mapa de la ciudad que nos indica L posibles locales donde podemos colocar un total de S supermercados (obviamente $L > S$). Además también disponemos del número de supermercados competidores que hay alrededor de cada local en el que se puede ubicar un supermercado.

   El objetivo es colocar todos los supermercados de manera que se minimice la competencia alrededor de cada supermercado y que se maximice la suma de las distancias entre ellos (obviamente disponemos de una tabla con sus distancias).

   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 solucionarlo mediante Hill-Climbing. Como solución inicial se parte de la solución vacía. Como operadores se usa el colocar un supermercado en un local vacío y quitar un supermercado de un local. Como función heurística se usa la suma de las distancias de los locales de cada uno de los S supermercados al resto, multiplicado por la suma de los supermercados competidores de cada supermercado.

   b) Se plantea solucionarlo mediante Hill-Climbing utilizando como solución inicial el colocar aleatoriamente los S supermercados y usando como operadores de búsqueda mover un supermercado al local vacío más cercano respecto a su posición actual. Como función heurística se usa la suma de las distancias de los locales de cada uno de los S supermercados al resto más la suma de los supermercados competidores de cada supermercado.

   c) Se plantea solucionarlo mediante Hill-Climbing utilizando como solución inicial el colocar los supermercados consecutivamente hasta colocar los S y para cada supermercado colocarlo en el local que esté a máxima distancia respecto al local donde se colocó el supermercado anterior. Se plantean como operadores mover un supermercado a cualquier local vacío cuya suma de distancias al resto de supermercados sea mayor que la suma de distancias del local actual. Como función heurística se usa la suma de los supermercados competidores de cada supermercado dividida por la suma de las distancias de los locales de cada uno de los S supermercados al resto.

   d) Se plantea resolverlo mediante algoritmos genéticos. Para representar el problema utilizamos una tira de L bits y como población inicial generamos aleatoriamente n individuos donde en cada uno hay exactamente S bits a 1. La función heurística es la suma de distancias de cada supermercado al resto más una constante por la suma de la competencia de cada local. Como operadores usamos un operador de cruce que intercambia aleatoriamente la mitad de los bits de cada individuo de la pareja cruzada y un operador de mutación que intercambia aleatoriamente dos posiciones de la tira de bits cumpliendo que en una sea un bit a 1 y en la otra un bit a 0.

2. (4 puntos) Queremos saber cómo distribuir un conjunto de plantas en un cultivo. Para poder organizar mejor el cultivo lo hemos dividido en una cuadrícula de $N \times M$. Las plantas que queremos colocar son de T diferentes tipos y tenemos k_t plantas de cada tipo. Cada planta tiene unas necesidades específicas de agua (a_t) y nutrientes (n_t) diarias.

   Cada posición de la cuadrícula puede albergar un máximo de P plantas de cualquier tipo. El sistema de riego permite aportar L litros de agua diarios a cada una de las M columnas de la cuadrícula y de cada posición las plantas pueden consumir hasta G gramos de nutrientes al día.

   Lo que hemos de conseguir es colocar todas las plantas que tenemos, con las restricciones de que el consumo de agua y nutrientes no superen las cantidades diarias y que, para agotar de manera uniforme los nutrientes de las posiciones, la diferencia en consumo de nutrientes entre una posición y sus adyacentes no supere cierto valor C.

   Se nos plantean las siguientes alternativas:

   a) Queremos utilizar satisfacción de restricciones donde las variables son las coordenadas de la cuadrícula del cultivo y los valores el identificador de cada planta a colocar (evidentemente una variable tendrá un conjunto de valores). Las restricciones aparecerían entre las posiciones de cada columna, de manera que las necesidades de agua totales no superen los L litros de agua que se aportan diariamente, habrá una restricción por posición que no permita que el consumo de nutrientes supere el límite G diario y restricciones entre una posición y sus adyacentes de manera que la diferencia de consumo de nutrientes no sea mayor que C.

   b) Queremos utilizar búsqueda local, la solución inicial consiste en colocar todos los elementos en la primera posición de la cuadrícula. El operador de modificación de la solución consiste en mover una planta de una posición a otra. Queremos minimizar la siguiente función heurística: suma del agua necesitada por columnas más suma de consumo de nutrientes por posición más suma del consumo de nutrientes de las posiciones que superan el valor C en su diferencia respecto a sus adyacentes.

   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.

Examen Parcial de IA

(11 de noviembre de 2008)
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 Hill-climbing. Como solución inicial se usa un calendario vacío. Como operadores usamos asignar una partida de una ronda de una categoría a una franja de un día y mover una partida de una franja a otra. Como función heurística usamos el número de días total en los que hay partidas asignadas mas la suma del número de días que ocupan las rondas de una categoría, más una constante por cada asignación en el calendario que viole alguna de las restricciones del problema.

d) 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.

2. (4 puntos) Después de los incendios del verano se nos ha planteado rediseñar la ubicación de los parques de bomberos. Para solucionar el problema dividimos el área en una cuadrícula de $N \times M$, cada posición tiene asignado un factor de accesibilidad A que toma un valor entre 1 y 3 (1 es una zona de fácil acceso, 3 es un área de difícil acceso).

Deseamos ubicar un total de P parques de bomberos, cada parque de bomberos puede tener entre uno y tres camiones. Tenemos un total de C camiones para repartir (obviamente $C > P$).

Definimos el factor de seguridad de una posición como la suma para esa posición y todas las que la rodean del cociente entre el número de camiones de bomberos que hay en la posición y su factor de accesibilidad.

El objetivo es ubicar todos los parques de bomberos y repartir entre ellos todos los camiones de manera que el factor de seguridad global (la suma para todas las posiciones) sea el máximo posible.

a) Queremos utilizar A* de manera que recorremos la cuadrícula de la esquina superior izquierda a la inferior derecha. El estado es la asignación que hemos hecho de parques de bomberos y camiones a las áreas recorridas. Utilizamos como operador poner un parque de bomberos, asignando uno, dos o tres camiones en el caso de ponerlo, el coste del operador es el número de camiones asignados más uno, o no ponerlo con coste uno. La función heurística es el número de áreas que nos quedan por visitar y vale infinito si ya hemos asignado más de C camiones o P parques de bomberos.

b) Queremos utilizar búsqueda local generando una solución inicial en la que repartimos al azar los P parques de bomberos con un camión cada uno. Los operadores son aumentar o disminuir el número de camiones de un parque y mover un parque con todos sus camiones a otra posición. La función heurística que queremos optimizar es el factor de seguridad global más una constante por el número de camiones que quedan por asignar.

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.

Examen Parcial de IA

(12 de noviembre de 2008)

grupo 10

Duración: 1 hora

1. (6 puntos) El ministerio de ciencia y tecnología desea saber como hacer la asignación de ayudas a proyectos de investigación de las C convocatorias de este año. Cada convocatoria de ayudas tiene un presupuesto a repartir entre las peticiones aceptadas. A estas convocatorias los grupos de investigación pueden presentar peticiones (la misma petición puede presentarse a hasta tres convocatorias distintas, pero solo se puede conceder una vez). Una petición tiene un presupuesto solicitado y el ministerio le asigna una prioridad entre 1 y 3 dependiendo del historial del grupo investigador. Hay un total de P peticiones.

   El ministerio quiere asignar la mayor cantidad de dinero posible, pero haciendo que globalmente la proporción de grupos con concesiones de ayuda cumpla las siguientes restricciones: como máximo el 30 % de concesiones ha de ser para los grupos de prioridad 2 y como máximo el 10 % para los de prioridad 3. Obviamente la suma de los presupuestos solicitados de las peticiones asignadas a una convocatoria no puede superar el presupuesto de la convocatoria.

   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 respecto a otras alternativas posibles. Justifica las respuestas.

   a) Usar Hill-climbing, tomando como solución inicial la solución vacía, como operadores tenemos asignar una petición a una convocatoria y desasignar una petición de una convocatoria. Como función heurística usamos:

   $$h'(n) = \sum_{\forall conv_j} \sum_{\forall pet_i \in conv_j} Presupuesto(pet_i) - \sum_{\forall conv_j} Presupuesto(conv_j)$$

   donde $pet_i \in conv_j$ significa que la petición i ha sido concedida en la convocatoria j.

   b) Usar Hill-climbing. Para hallar la solución inicial recorremos todas las convocatorias y asignamos tantas peticiones de prioridad 1 solicitadas a esa convocatoria como podamos, sin exceder su presupuesto. Como operadores utilizamos asignar una petición a una convocatoria, respetando las restricciones de máximos de peticiones concedidas por prioridad, y mover una petición de una convocatoria a otra. Como función heurística usamos:

   $$h'(n) = \frac{\sum_{\forall conv_j} \sum_{\forall pet_i \in conv_j} Presupuesto(pet_i)}{\sum_{\forall conv_j} Presupuesto(conv_j)}$$

   c) Usar Hill-climbing. Para hallar la solución inicial recorremos todas las convocatorias y vamos asignando al azar peticiones solicitadas a esa convocatoria sin exceder su presupuesto. Como operadores utilizamos asignar y desasignar una petición a una convocatoria, respetando las restricciones de máximos de peticiones concedidas por prioridad, y mover una petición de una convocatoria a otra. Como función heurística usamos:

   $$h'(n) = \sum_{\forall conv_j} \left(Presupuesto(conv_j) - \sum_{\forall pet_i \in conv_j} Presupuesto(pet_i) \right)$$

d) Usar algoritmos genéticos. Codificamos el problema con $C \times P$ bits donde cada grupo de P bits corresponde a una convocatoria de manera que un uno significa que la petición ha sido concedida en esa convocatoria. Las soluciones iniciales las generamos utilizando el método del apartado anterior. Como operadores genéticos usamos los operadores habituales de cruce y mutación. Como función heurística usamos:

$$h'(n) = C - \sum_{\forall conv_j} \frac{\sum_{\forall pet_i \in conv_j} Presupuesto(pet_i)}{Presupuesto(conv_j)}$$

2. (4 puntos) Una empresa de transportes quiere planificar sus envíos a través de tren de una ciudad a otra. La empresa tiene una serie de cargas que tienen una fecha tentativa de envío, pero sabemos que tenemos un margen de hasta tres días que podemos usar para retrasarlo. También disponemos del tamaño del envío, que cae en tres categorias (1, 2 y 3), cada una de tamaño mayor que la anterior.

   Tenemos un horario de trenes que nos indica el espacio de contenedor que tenemos disponible en una fecha concreta en diferentes horas del día, este espacio tiene tres categorías de tamaño (1, 2 y 3), acordes con los tamaños de los envios, y en él se puede transportar una carga que quepa, sabiendo que en un espacio de contenedor grande se puede poner una carga más pequeña. El espacio solo se puede reservar entero, de manera que si una carga es más pequeña estaremos desperdiciando una parte.

   El objetivo sería asignar las cargas en los espacios de los trenes de manera que se minimice el espacio desperdiciado y que los envíos se retrasen lo mínimo posible. Supondremos que tenemos suficientes espacios disponibles en los trenes para asignar todas las cargas y que hay muchos más espacios disponibles que cargas. Se nos plantean las siguientes formas de solucionar el problema:

   a) Queremos utilizar A* de manera que ordenamos los espacios de contenedor en los trenes según la fecha en los que están disponibles (en caso de estar disponible en la misma fecha decidimos un criterio de ordenación). Procedemos a asignar las cargas siguiendo el orden establecido para los espacios en los trenes utilizando dos operadores: asignar una carga que quepa en el espacio disponible y que no viole las restricciones de fecha de envío (está dentro del margen de tres días), el coste sería el valor de la categoría del espacio disponible en el tren más uno, o no asignar nada (el coste sería uno). Como función heurística utilizaremos la suma de las categorias de las cargas que quedan por asignar.

   b) Queremos usar hill climbing generando una solución inicial mediante un mecanismo voraz que vaya asignando cada carga al primer espacio disponible que cumpla las restricciones de fecha de inicio teniendo en cuenta el margen de tres días. Como operadores utilizamos intercambiar dos cargas de tren si no se viola ninguna restricción. Como función heurística se utiliza la suma de las categorías de los espacios usados en los trenes, más el número de días de retraso de cada carga respecto a la fecha tentativa de envío.

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 13th 2008)
Time: 1 hour

1. (6 points) The ministry of science and technology needs to allocate the grants for research projects in this year's P funding programs. Each funding program has a budget to distribute among the accepted grant applications. Research groups can apply to these funding programs. They can apply up to three programs, but only one application can be accepted. Each grant application has a proposed budget and the ministry assigns each research group a priority between 1 and 3 depending on their experience. There are A grant applications.

   The ministry wants to allocate as much as money as possible, but maintaining a ratio among the accepted grants of groups of different priorities. There must be a maximum of 30 % of accepted grants of groups of priority 2 and a maximum of 10 % of accepted grants of groups of priority 3. Obviously, the sum of the proposed budgets of grants accepted in a funding program can not be greater that the budget of the program.

   We want to solve this problem as a 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 if it is correct, it is efficient, or it is better or worse than other alternatives. Justify your comments.

   a) Using Hill climbing search, taking as initial state the empty state, as search operators we use assigning or deassigning an application to a funding program. As heuristic function we use:

   $$h'(n) = \sum_{\forall prog_j} \sum_{\forall app_i \in prog_j} Budget(app_i) - \sum_{\forall prog_j} Budget(prog_j)$$

   where $app_i \in prog_j$ means that application i has been accepted in funding program j.

   b) Using Hill climbing search. The initial solution is computed by assigning to each funding program as much as applications of groups of priority 1 as possible without exceeding the program budget. As search operators we use assigning an application to a program if the constraint of maximum ratio of applications granted per priority group allows it and to move an application from a program to another. As heuristic function we use:

   $$h'(n) = \frac{\sum_{\forall prog_j} \sum_{\forall app_i \in prog_j} Budget(app_i)}{\sum_{\forall app_j} Budget(app_j)}$$

   c) Using Hill climbing search. The initial solution is computed by randomly assigning grant applications to funding programs without exceeding the programs budget. As search operators we use assigning or deassigning an application from a funding program if the constraint of maximum ratio of applications granted per priority group allows it and moving an application from a program to another. As heuristic function we use:

   $$h'(n) = \sum_{\forall prog_j} \left(Budget(prog_j) - \sum_{\forall app_i \in prog_j} Budget(app_i) \right)$$

d) Using genetic algorithms. The representation of the solution uses $P \times A$ bits where each group of A bits correspond to a funding program, if the bit value is one this means that the application is granted in that program. The initial solutions are obtained using the methodology of the previous question. As genetic operators we use the usual crossover and mutation operators. As heuristic function we use:

$$h'(n) = P - \sum_{\forall prog_j} \frac{\sum_{\forall app_i \in prog_j} Budget(app_i)}{Budget(prog_j)}$$

2. (4 points) A transportation company needs to plan its shipments from one town to another by train. The company has packages that have a tentative date of delivery, but they have a margin of up to three days that can be used to delay its final delivery. Each package has a size that can fall in one of three categories (1, 2 and 3), each one bigger than the previous one.

We have a train schedule that informs us about the available container space in a specific date and hour. The container space has categories 1, 2 and 3, with sizes corresponding to the categories of package sizes. In this container space we can place a package of equal or smaller category. We have to reserve the whole container space so, if the package is smaller than the container we are wasting space.

The goal is to allocate the packages into the containers of the trains in order to minimize the waste of space and that the deliveries have the minimum possible delay. We will suppose that we have enough containers to send all the packages and there are much more containers than packages. We have two possibilities to solve the problem:

a) We want to use A*. We sort the containers using the date of availability (if the date in two containers is the same we choose another sorting criteria). We proceed to allocate the packages following the order of the containers using two operators: assigning a package to a container if it fits and if the date of availability of the container is between the date of delivery and the three days margin, the cost is the category of the container plus one, or not assigning the package with cost one. As heuristic function we use the sum of the categories of the packages not yet allocated.

b) We want to use hill climbing, generating an initial solution using a greedy algorithm that assigns each package to the first container space available that has a date between the date of delivery of the package and the three days margin. As search operators we use swapping two packages in two container spaces if all constraints are preserved. As heuristic function we use the sum of the categories of the container spaces used plus the sum of the number of days that each package is delayed.

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 it should be corrected and the advantages or disadvantages of each one. Justify your answers.