Examen Parcial de IA

(5 de noviembre de 2007)
grupo 20
Duración: 1 hora

1. (6 puntos) Una empresa del mundo del espectáculo gestiona los contratos de N grupos musicales que se dedican a actuar en las fiestas de los pueblos. Cada grupo musical tiene una tarifa por actuación, de la que una parte se la queda el grupo y otra la empresa. Por otro lado, los municipios celebran sus fiestas patronales en unas ciertas fechas y dedican una parte de su presupuesto festivo, X euros, a pagar las actuaciones de los grupos musicales. El problema es organizar las giras de los grupos de modo que se cubran las peticiones de los pueblos, en total P actuaciones, sin pasarse del presupuesto municipal, se maximice la recaudación de la empresa y se minimicen los kilómetros que al cabo de la temporada ha de realizar cada grupo. Además, el mismo grupo no puede actuar más de un día en el mismo pueblo y cuando acaban en un pueblo, se van al siguiente en el que tienen actuación.

   En los siguientes apartados se proponen diferentes alternativas para algunos de los elementos necesarios para plantear la búsqueda (solución inicial, operadores, función heurística). Hay que comentar la solución que se propone respecto a si es correcta, es eficiente, y es mejor o peor en comparación con otras alternativas. Y hay que justificar todas las respuestas.

   a) Se propone usar Hill-Climbing. La solución de partida coloca, en orden creciente de fecha, un grupo diferente en cada una de las fechas previstas de actuación y cada grupo se asigna una sola vez. El operador disponible es `poner_grupo` y la función heurística es:

   $$h'(n) = \frac{\sum_{\forall grupo_i} Kilometros\_recorridos(grupo_i)}{\sum_{\forall grupo_i} Dinero\_recaudado\_empresa(grupo_i)}$$

   b) Se propone usar Hill-Climbing. La solución de partida se calcula de la siguiente manera: se ordenan los grupos en orden decreciente de beneficio para la empresa y se empiezan a asignar siguiendo ese orden a cada una de las fechas disponibles. Si cuando ya se han colocado todos los grupos una vez todavía quedan fechas sin cubrir, se repite el proceso las veces que sea necesario. Se respeta que el mismo grupo no actúe más de una vez en el mismo pueblo. El operador es `intercambiar_grupo` siempre y cuando no suponga que se supera el presupuesto del municipio y se respete una única actuación por grupo en el mismo pueblo. La función heurística es:

   $$h'(n) = \sum_{\forall grupo_i} Kilometros\_recorridos(grupo_i) - \sum_{\forall grupo_i} Dinero\_recaudado\_empresa(grupo_i) * 100$$

   c) Se propone usar Hill-Climbing. La solución de partida se calcula de la siguiente forma: Se ordenan los grupos en orden decreciente de beneficio, se toma el primer grupo y se le asigna la primera fecha libre sin pasarse del presupuesto del municipio, luego se busca la siguiente fecha libre en la que pueda actuar el mismo grupo, y tal que suponga un desplazamiento mínimo para el grupo y que no supere el presupuesto del municipio, se le asigna el grupo y se repite el proceso hasta que no queden más fechas posibles para ese primer grupo. Se repite exactamente el mismo proceso para los grupos sucesivos hasta que ya no queden fechas por cubrir o no haya más grupos.

   Disponemos de tres operadores `quitar_grupo`, `poner_grupo` e `intercambiar_grupo`. Todos ellos respetan no sobrepasar el presupuesto del municipio y una sola actuación por grupo en un pueblo. El heurístico es el siguiente:

$$h'(n) = \sum_{\forall grupo_i} Kilometros\_recorridos(grupo_i) \times \frac{Dinero\_cobrado(grupo_i)}{Dinero\_recaudado\_empresa(grupo_i)}$$

d) Usar Algoritmos Genéticos. Se asigna un identificador binario de longitud b a cada uno de los grupos, la solución se representa como una tira de $P \times b$ bits. Se utiliza la misma estrategia para generar la solución inicial que en el apartado anterior. Como operadores genéticos se utiliza solamente el operador de cruce, donde los puntos de cruce no pueden ser dentro de la identificación de un grupo. Como función heurística utilizamos también la del apartado anterior.

2. (4 puntos) Queremos determinar cómo abastecer N tiendas a partir de k almacenes. Tiendas y almacenes se encuentran distribuidos dentro de una ciudad y conocemos las coordenadas en las que se ubican. El abastecimiento se realiza mediante camiones que cargan las mercancías necesarias para una tienda, las llevan y vuelven al almacén. Cada tienda tiene una demanda ($D(T_i)$) diferente.

   Lo que nos interesa es que los almacenes tengan aproximadamente la misma cantidad de mercancías, por lo que la suma de las demandas de las tiendas que servimos desde cada almacén ha de ser aproximadamente igual. También nos interesa que los kilómetros que han de recorrer los camiones para abastecer a las N tiendas sean los menos posibles.

   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.

   a) Queremos resolver el problema utilizando búsqueda local generando como solución inicial un estado en el que las tiendas están asignadas al almacén más cercano. Como operadores de búsqueda utilizamos intercambiar una tienda de un almacén a otro. Como función heurística utilizamos la función:

$$h'(n) = \sum_{i=1}^{k} \sum_{\forall T_j \in Abast(A_i)} Dist(T_j, A_i) + \sum_{i=1}^{k} \left| \frac{\sum_{\forall T_j} D(T_j)}{k} - \sum_{\forall T_j \in Abast(A_i)} D(T_j) \right|$$

   Donde $Abast(A_i)$ es el conjunto de tiendas que abastece el almacén A_i, $Dist(T_j, A_i)$ es la distancia entre la tienda T_j y el almacén A_i

   b) Queremos solucionar el problema mediante satisfacción de restricciones, como variables escogemos $k \times N$ variables binarias de manera que cada tienda tiene asociadas k variables que representan el almacén al que está asignada. Como restricción imponemos que solo una de esas variables puede ser cierta. Añadimos también las siguientes restricciones al problema:

   - Para cualquier almacén, la suma de las demandas de las tiendas asignadas a él no puede ser mayor en un 5 % a la suma de las demandas de las tiendas asignadas a cualquiera otro almacén.
   - Para cualquier almacén, la suma de las distancias de las tiendas asignadas a él no puede ser mayor en un 5 % a la suma de las distancias de las tiendas asignadas a cualquiera otro almacén.

Examen Parcial de IA

(7 de noviembre de 2007)
grupo 10
Duración: 1 hora

1. (6 puntos) Deseamos construir un sistema que a partir de documentos textuales extraiga relaciones entre posibles eventos que aparezcan en los documentos y participantes en los mismos. Los documentos a tratar habrán sido preprocesados para marcar en ellos los componentes (usaremos C_j para notar los componentes genéricos):

   - E_i Eventos asociados a verbos, E es el conjunto de eventos y $|E|$ su número. Se indica si es singular o plural.
   - L_i Lugares, L es el conjunto de lugares y $|L|$ su número.
   - F_i Fechas, F es el conjunto de fechas y $|F|$ su número.
   - P_i Personas, P es el conjunto de personas y $|P|$ su número.

   Por ejemplo, el documento "Montilla y Magdalena Álvarez fueron recibidos ayer en la Moncloa por Zapatero mientras llovía a cántaros en Madrid, por la tarde, con Chaves, irán a Sevilla en el AVE" aparecerá tras el preproceso marcada de la siguiente forma:

   "`<P`$_1$ `Montilla>` y `<P`$_2$ `Magdalena_Álvarez>` `<E`$_1$ `recibir+plural>` `<F`$_1$ `ayer>` en la `<L`$_1$ `Moncloa>` por `<P`$_3$ `Zapatero>` mientras `<E`$_2$ `llover+singular>` a cántaros en `<L`$_2$ `Madrid>`, `<F`$_2$ `por_la_tarde>`, con `<P`$_4$ `Chaves>`, `<E`$_3$ `ir+plural>` a `<L`$_3$ `Sevilla>` en el AVE".

   Cada evento E_i debe tener asociados una fecha, F_i, un lugar, L_i, y un conjunto de personas participantes (para cada tipo de evento existirá un mínimo y un máximo, por ejemplo, min(recibir) = 2, max(recibir)= infinito, min(llover)=0, max(llover)=0, min(ir)=1, max(ir)=infinito. Si el evento está en plural el mínimo se incrementa en una unidad (ej. min(recibir+plural) = 3). Cada persona, P_i , lugar, L_i, y fecha, F_i, deben estar asociadas al menos a un evento. Notaremos el número de palabras del documento (tras el preproceso) como $|D|$.

   Por ejemplo, del ejemplo anterior se podrían obtener las siguientes asignaciones:

evento		lugar		fecha		personas	
E_1	recibir	L_1	Moncloa	F_1	ayer	P_1, P_2 , P_3	Montilla, Magdalena_Álvarez, Zapatero
E2	llover	L2	Madrid	F1	ayer		
E3	ir	L3	Sevilla	F2	esta tarde	P_2 , P_4	Magdalena_Álvarez, Chaves

   Asumiremos para simplificar que todas las menciones de componentes C$_j$ (eventos, personas, lugares y fechas) son diferentes. Además $|F| \leq |E|$ y $|L| \leq |E|$. Disponemos de una métrica trivial de distancia entre los componentes, el número de palabras entre ellos (ej. entre P_1 y E_1 la distancia es 2). Bajo estas restricciones deseamos minimizar la suma de distancias de los componentes a los eventos a que están asignados.

   En los siguientes apartados se proponen diferentes alternativas para algunos de los elementos necesarios para plantear la búsqueda (solución inicial, operadores, función heurística). Hay que comentar la solución que se propone respecto a si es correcta, es eficiente, y es mejor o peor en comparación con otras alternativas. Justifica la respuesta ilustrándola con el ejemplo propuesto. Indica un valor aproximado del factor de ramificación de cada caso.

a) Queremos resolver el problema utilizando Hill Climbing a partir de una solución inicial que asigne a cada componente P, L, F al primer evento. Como operador de búsqueda podemos usar 1) el cambio de asignación de un evento a otro y 2) una asignación nueva de un componente a un evento. Como heurística utilizamos la suma de distancias de cada componente a los eventos a que está asignado.

b) Queremos resolver el problema utilizando Hill Climbing a partir de una solución inicial que asigne a cada componente el evento más cercano. Usaremos los mismos operadores y heurística que en el primer apartado.

c) Queremos resolver el problema utilizando Hill Climbing con los mismos operadores y heurística que en el primer apartado a partir de la siguiente solución inicial:

- A cada E_i de E asignarle la F_j de F más próxima priorizando las F no asignadas.
- A cada E_i de E asignarle la L_j de L más próxima priorizando las L no asignadas.
- A cada E_i de E asignarle hasta $\text{mín}(E_i)$ P_j's de P priorizando las más próximas y las no asignadas.
- Si quedan aún P_j's por asignar, asignarlas al E_i de E más próximo.

d) Considerar el mismo caso anterior usando como heurística la suma de dos factores, el primero es la la suma de distancias de cada componente a los eventos a que está asignado. El segundo es la suma para cada componente C_j de las asignaciones que faltan para llegar al mínimo o que superan al máximo multiplicadas por $|D|$.

e) Considerar que dividimos el problema en 3 partes, la asignación de L's, la de F's y la de P's. Las dos primeras las resolvemos como en el apartado anterior. Para la asignación de P's usamos algoritmos genéticos. Como representación usaremos una tira de $|E| \times |P|$ bits donde para cada E_i existe un bit para cada posible P_j. Usaremos como función de fitting la misma de en el apartado anterior y los operadores habituales de cruce y mutación garantizando las fronteras de $|P|$ bits.

2. (4 puntos) Queremos generar los horarios para los vuelos de Virgin Galactic entre la Tierra (n espaciopuertos) y la Luna (un espaciopuerto). Se conoce el número estimado de pasajeros por día de la semana y ruta. Cada ruta tiene que tener un número mínimo de vuelos por semana (vs), preestablecido por Virgin. Además se fija también un número máximo M total de naves espaciales que pueden viajar al día. En cada nave caben p pasajeros. Con los M vuelos diarios deberiamos tener suficiente para poder cubrir los vs vuelos semanales de cada linea. Se plantean las siguientes alternativas para maximizar el número de usuarios transportados por semana:

   a) Queremos usar satisfacción de restricciones, donde hay una variable por cada combinación de rutas y días de la semana, los valores son el número de vuelos de esa combinación. Las restricciones son: La suma del número de vuelos para una ruta en una semana no puede ser menor que vs, la suma del número de naves espaciales para un día no ha de ser superior M y el número total de plazas que se quedan vacías tiene que ser menor que un cierto valor $u \times n$.

   b) Queremos usar búsqueda local, donde se genera una solución inicial colocando suficientes naves espaciales para cubrir el mínimo número de vuelos de cada ruta para cada semana. Los operadores consisten en: mover un vuelo de día y/o ruta y añadir un vuelo a un día y ruta. Se minimiza el número total de plazas que se quedan vacías.

   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

(7 de noviembre de 2007)
grupo 30
Duración: 1 hora

1. (6 puntos) Los propietarios del hotel “Bienestar” trabajan habitualmente con N tour-operadores para los cuales deben reservar siempre un conjunto de habitaciones. A principios de año, cada operador realiza su petición de habitaciones para todo el año. Las peticiones son consideradas como el número máximo a reservar, pero los propietarios saben que no tienen habitaciones suficientes para cubrir todas las peticiones de todos los tour-operadores. Por esta razón, internamente tienen definido un número mínimo a asignar a cada tour-operador. Este número les permite mantener las buenas relaciones. A partir de aquí asignarán a los tour-operadores un número de habitaciones que estará entre el mínimo interno y la petición real recibida.

   Para asignar las habitaciones, los propietarios tienen varias restricciones a respetar. Por un lado, hay un número mínimo de habitaciones que queda siempre bajo la gestión directa del hotel. Por otro lado, se quiere maximizar los beneficios obtenidos de las habitaciones reservadas a los tour-operadores, teniendo en cuenta que los beneficios son distintos dependiendo del tour-operador, e incluso dependiendo de la temporada (alta, media, baja). Adicionalmente, se quiere maximizar la “calidad de la ocupación” para lo cual tienen asignado a cada tour-operador un índice Q que cuantifica la calidad de los turistas que suelen venir a través del operador en cuestión. Este índice se asociará a cada habitación reservada para ese operador. El propio hotel tiene también un índice Q.

   En los siguientes apartados se proponen diferentes alternativas para algunos de los elementos necesarios para plantear la búsqueda (solución inicial, operadores, función heurística,...). Comentar cada apartado indicando si se considera correcta, eficiente, mejor que otra. Justifica tus respuestas.

   a) Se plantea aplicar Hill-climbing usando como solución inicial asignar el mínimo de habitaciones a cada tour-operador. Se define como operador asignar una habitación de una determinada temporada a un tour-operador distinto del que la tiene asignada.

   b) Se plantea aplicar Hill-climbing usando como solución inicial asignar el mínimo de habitaciones a cada tour-operador y del conjunto de habitaciones no asignadas apartar el mínimo para el hotel y asignar el resto al tour-operador con mayor Q. Se define como operador asignar una habitación de una determinada temporada a un tour-operador distinto del que la tiene asignada, siempre y cuando el nuevo tour-operador tenga mejor Q y se respeten todos mínimos.

   c) Usando la solución inicial y el operador definidos en el apartado anterior, se plantea como función de evaluación la suma total de los beneficios obtenidos de cada habitación.

   d) Se plantea aplicar Hill-climbing usando como solución inicial asignar el mínimo de habitaciones a cada tour-operador y del conjunto de habitaciones no asignadas apartar el mínimo para el hotel y asignar el resto al tour-operador que nos da mayor beneficio. Se plantea como función de evaluación la suma total de los beneficios obtenidos por habitación más la suma del índice Q de cada una de ellas.

   e) Se plantea usar algoritmos genéticos donde la representación de la solución es una tira de bits donde hay $3 \times (N+1)$ secuencias bits. Para cada tour-operador y el hotel tenemos asociadas 3 secuencias de bits que representan el número de habitaciones asignadas para cada una de las temporadas. Cada secuencia de bits ha de ser suficiente par codificar en binario el número total de habitaciones del hotel. Los operadores a usar son los habituales de cruce y mutación. Se propone como función de evaluación la suma total de los beneficios obtenidos por habitación más la suma del índice Q de cada una de ellas. La función valdrá infinito cuando no se cumplan los mínimos.

2. (4 puntos) Una empresa pesquera y conservera envía sus N barcos $\{b_i\}$ con capacidades para $C(b_i)$ pescadores a sus D destinos internacionales $\{d_i\}$ $(N > D)$ para 180 dias de pesca. Algunos de sus M pescadores $\{p_i\}$ $(M > \sum C(b_i))$ son asignados a los barcos y son enviados a los destinos o bien con el mismo barco asignado o bien por avión. Sin embargo, el coste por marinero del viaje en avión a un destino d_j, $K_a(d_j)$, es mayor que el coste por marinero del viaje en el barco, $K_b(d_j, b_i)$. Cada pescador p_i proporciona a la empresa una ganancia de $g(p_i)$ euros/dia en media y cuesta a la empresa $k(p_i)$ euros/dia en media.

   A veces, a la empresa le interesa obtener la planificación (asignación de pescadores y barcos a todos sus destinos, y formas de viaje de los pescadores) que le aporta menos pérdidas, otras veces requiere una planificación que cubra todos sus destinos sin pérdidas.

   En los siguientes apartados se proponen diferentes alternativas para algunos de los elementos necesarios para plantear la búsqueda (solución inicial, operadores, función heurística). Hay que comentar la solución que se propone respecto a si es correcta, es eficiente, y es mejor o peor en comparación con otras alternativas. Y hay que justificar todas las respuestas.

   a) Aplicar un algoritmo de búsqueda local. La solución inicial consiste en: 1) asignar secuencialmente $NdivD$ barcos a cada destino, repartiendo los $NmodD$ equitativamente entre los primeros $NmodD$ destinos; 2) asignar aleatóriamente $C(b_i)$ pescadores a cada barco b_i, transportándolos a todos en el barco. Los operadores son: cambiar el destino de un barco y el de, todos sus pescadores, eliminar un pescador de un barco y cambiar el método de transporte de un pescador. Como función heurística usamos:

   $$h'(n) = \sum_{i=1}^{D} \sum_{\forall b_j \in Asig(d_i))} \left(K_a(d_i) \times |A(b_j)| + K_b(d_i, b_j) + 180 \times \sum_{\forall p_i \in O(b_j)} (k(p_i)) - g(p_i)) \right)$$

   donde $O(b_j)$ es la asignación de marineros del barco b_j $(|O(b_j)| \leq C(b_j))$, $A(b_j)$ es la asignación de marineros del barco b_j a ser transportados por avión, y $Asig(d_i)$ son los barcos asignados al destino d_i.

   b) Aplicar un algoritmo de satisfacción de restricciones. Las variables son todos los posibles pares $< p_i, d_j >$ (pescador, destino). Todas las variables tienen como dominio el conjunto de pares $\{< b_i, avion >, < b_i, barco >, < undef, undef >\}$. El primer par denota que el barco b_i se asigna al destino, pero que el pescador llegará en avión. El segundo par denotan que el barco b_i se asigna al destino y que el pescador llegará en él. El tercer valor denota que el pescador no viaja al destino. Las restricciones son:

   $$\forall_i |O(b_i)| \leq C(b_i)$$

   $$\sum_j \sum_{p_i \in O(b_j)} (k(p_i)) - g(p_i)) \geq 0$$