# Examen Parcial de IA

**(4 de abril de 2006)**
**grupo 10**
**Duración: 1 hora**

1. (4 puntos) Queremos planificar cómo debemos integrar diferentes procesos 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 un grafo de proceso. Para simplificar supondremos que cada paso tiene la misma duración. Hay que tener en cuenta que un producto puede necesitar un mismo tipo de maquina en diferentes pasos de su proceso de fabricación.

   Disponemos de un número fijo de máquinas de cada tipo ($M$). 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.

   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 pasos de cada proceso 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 búsqueda local para minimizar el número total de pasos del proceso integrado. Partimos de una solución inicial en la que ponemos todos los procesos uno detrás de otro en secuencia. El operador de modificación de la solución consiste en unir un paso del proceso solución con otro siempre que se respete el orden de ejecución de los procesos individuales. La función heurística es el número de pasos del proceso integrado.

   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.

2. (6 puntos) Tenemos una pequeña flota de $C$ camiones que utilizamos para repartir mercancías y cada día tenemos que determinar qué ruta ha de seguir cada camión para abastecer un conjunto de ciudades por todo el país. El objetivo es que todos los camiones acaben la jornada aproximadamente a la misma hora, por lo que el número de kilometros que ha de recorrer cada camión ha de ser muy parecido, y que recorran en total en mínimo número de kilómetros.

   Disponemos de un mapa de carreteras que nos dice la distancia en kilómetros entre cada ciudad donde hemos de dejar nuestra mercancía, suponemos que todos los camiones parten de la misma ciudad y han de volver a ella al final del día, cargan al principio de la jornada todo lo que han de repartir, han de pasar sólo una vez por cada ciudad.

   Una posible solución a éste problema se puede obtener mediante el uso de algoritmos de búsqueda local. 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 y si es mejor/peor respecto a otras alternativas posibles. Justifica tus respuestas.

   *a*) Usar Hill Climbing. Como solución inicial asignamos al azar a cada camión un número aproximadamente igual de ciudades, recorriéndolas en orden también al azar. Como operadores usamos el intercambiar dos ciudades entre los recorridos de dos camiones e intercambiar las posiciones de dos ciudades en el recorrido de un camión. La función heurística es la siguiente:

$$h'(n) = \sum_{i=1}^{C} \left( \frac{LR_i}{\sum_{i=1}^{C} LR_i} - \frac{1}{C} \right)$$

donde $LR_i$ es la longitud del recorrido del camión $i$.

*b*) Usar Hill Climbing. Como solución inicial asignamos todas las ciudades a un camión, estableciendo el recorrido inicial mediante una estrategia avariciosa de manera que intentemos minimizarlo. Como operadores usamos el mover una ciudad del recorrido de un camión a otro e intercambiar las posiciones de dos ciudades en el recorrido de un camión. La función heurística es la siguiente:

$$h'(n) = \prod_{i=1}^{C} LR_i$$

*c*) Usar Hill climbing. Como solución inicial escogemos las $C$ ciudades más cercanas a la ciudad origen como la primera ciudad a visitar por cada camión, como segunda ciudad en el recorrido de cada camión escogemos la más cercana a la primera que no esté ya asignada, y así sucesivamente hasta asignar todas las ciudades. Como operadores usamos el mover una ciudad del recorrido de un camión a otro, intercambiar las posiciones de dos ciudades en el recorrido de un camión e intercambiar dos ciudades entre los recorridos de dos camiones. La función heurística es la siguiente:

$$h'(n) = \frac{C \cdot (C-1)}{2} - \sum_{i=1}^{C} \sum_{j=i+1}^{C} \frac{LR_i}{LR_j}$$

*d*) Usar algoritmos genéticos. Donde cada ciudad está representada por tantos bits como sean necesarios para codificar el valor $C$, la tira de bits contiene concatenados los bits de todas las ciudades, es decir, representamos en la tira de bits el número del camión que ha de recorrerla. Como operadores utilizamos los operadores habituales de cruce y mutación.

# Examen Parcial de IA

**(5 de abril de 2006)**
**grupo 20**
**Duración: 1 hora**

1. (4 puntos) Deseamos construir un circuito integrado de manera que el calor que generan los diferentes elementos se disperse de la manera mas uniforme posible para evitar que se queme. Para simplificar el problema hemos supuesto que el circuito es una cuadrícula de $N \times M$. Los diferentes elementos que queremos colocar en el circuito tienen $i$ diferentes tipos y tenemos que colocar un número $e_i$ de cada uno de ellos. Cada elemento genera una cantidad de calor específica.

   Cada posición del circuito puede albergar $p$ elementos de cualquier tipo y hemos de colocar todos los elementos.

   Lo que hemos de conseguir es que la suma de los calores generados por los elementos para cada fila y cada columna no supere un valor $c$ y que la diferencia de calor entre una celda y cada una de sus cuatro vecinas no sea mayor que un valor $v$.

   Se nos plantean las siguientes alternativas:

   *a*) Queremos utilizar satisfacción de restricciones donde las variables son las coordenadas de la cuadrícula del circuito y los valores el identificador de cada elemento a colocar (evidentemente una variable tendrá un conjunto de valores). Las restricciones aparecerían entre las celdas según filas y columnas, de manera que su suma no superara el valor $c$, y entre celdas vecinas, de manera que su diferencia no fuera superior a $v$

   *b*) Queremos utilizar búsqueda local, la solución inicial consiste en colocar todos los elementos en la primera celda de la cuadrícula. El operador de modificación de la solución consiste en mover un elemento de una celda a otra. Queremos minimizar la siguiente función heurística: suma del calor de filas y columnas que superan $c$, mas las suma del calor de las celdas que superan el valor $v$ en su diferencia respecto a sus vecinas.

   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.

2. (6 puntos) La sala de cine “Lo nunca visto” desea realizar una planificación estratégica de las películas a proyectar durante un año (52 semanas) siguiendo varios criterios comerciales. Cada película tiene asociada varias informaciones relevantes para este problema: tipo (infantil/adulto), previsión de beneficio semanal, índice de calidad. La programación anual debe contener al menos un 15 % de películas infantiles y un 40 % de películas para adultos. Se desea maximizar el beneficio anual de proyección de películas, pero sin dejar de lado la calidad. Por este motivo, el índice global de calidad no debe ser inferior a una cota predeterminada (Q). En el caso de que una película deba proyectarse más de una semana, dichas semanas deberán ser consecutivas. Las estrategias comerciales exigen que una película no se proyecte más de 8 semanas.

   Teniendo en cuenta el escenario descrito, comenta brevemente las diversas propuestas que se describen en los apartados siguientes. Valora si son correctas o no, si son eficientes o no, si son mejores o peores que otras alternativas. Justifica tus respuestas.

   *a*) Usar un Hill-climbing. Como estado inicial asignamos aleatoriamente una película distinta a cada una de las 52 semanas. Como operador usamos Asignar-película (título, num-semana), que sustituye la película asignada una semana por otra nueva o por alguna de las ya asignadas a otras semanas.

b) Usar un Hill-climbing. Como estado inicial asignamos dos películas infantiles (4 semanas cada una) y seis películas para adultos (4 semanas cada una), el resto de las semanas se asigna aleatoriamente. Como operador usamos el mismo del apartado a.

c) Usar un Hill-climbing. Partimos del mismo estado inicial del apartado b con el control adicional de que el índice de calidad no sea inferior a Q. Al operador de asignación le añadimos como condiciones de aplicabilidad que mantenga las proporciones mínimas de películas infantiles y para adultos, que ninguna película supere las 8 semanas y que se respete la cota inferior Q. Como función heurística usaremos h1'(n): la suma de la previsión de beneficio semanal para cada película proyectada multiplicada por el número de semanas asignado.

d) Usar un Hill-climbing. Como heurístico usaremos h2'(n): h1'(n) + suma del índice de calidad de cada película proyectada multiplicado por el número de semanas asignado.

e) Usar algoritmos genéticos. Cada individuo se representa mediante 52 tiras de bits. La longitud de la tira de bits es suficiente para codificar el número de identificación de todas las películas. Como población inicial se generan n individuos con los criterios de estado inicial descritos en b y c. Como función de fitness usamos h3'(n): h1'(n) + (suma del índice de calidad de cada película proyectada multiplicado por el número de semanas asignado - Q). Como operadores usamos el operador de cruce habitual, pero trabajando siempre con tiras completas de bits (la identificación de una película), es decir, un punto de cruce nunca parte el identificador de una película.

# Examen Parcial de IA

**(5 de abril de 2006)**
**grupo 30**
**Duración: 1 hora**

1. (4 puntos) Para hacer presión ante RENFE, los sindicatos de conductores de Cercanías en Barcelona quieren encontrar la forma de cumplir los requisitos mínimos que se les imponen sobre número de trenes, pero afectando al mayor número de usuarios, para que sus reclamaciones les den a ellos mayor poder en las negociaciones. Para ello han construido una tabla que, para cada hora y línea de cercanías, les dice el número de usuarios que viajan a esa hora por esa línea en cada tren, y el número mínimo de trenes que según RENFE han de pasar durante esa hora por esa línea para cumplir los servicios mínimos. Hay además otra regla que han de cumplir, y es que RENFE también impone un número mínimo $M$ total de trenes que han de circular al día, siendo este número algo mayor que la suma de los trenes de los servicios mínimos antes mencionados.

   Se nos plantean las siguientes alternativas:

   *a*) Queremos utilizar satisfacción de restricciones donde tenemos una variable por cada línea de cercanías y cada hora, y los valores son el número de trenes asignados a cada línea y cada hora. Las restricciones son el número mínimo de trenes para cada línea y hora, el número mínimo de trenes que han de circular durante el día y que el número total de usuarios que se queden sin tren sea mayor que un cierto valor $U$.

   *b*) Queremos utilizar búsqueda local, donde se genera una solución inicial colocando suficientes trenes para cubrir los servicios mínimos de cada línea y asignando aleatoriamente los trenes restantes entre todas las líneas. El operador de modificación de la solución consiste en mover un tren de una hora a otra en la misma línea, o mover un tren de una línea a otra. Queremos maximizar el número de usuarios que se verán afectados por la falta de trenes.

   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.

2. (6 puntos) Para planificar un examen de una asignatura con $N$ estudiantes matriculados $e_1 \ldots e_N$, la FIB debe reservar $S$ aulas $a_1 \ldots a_S$ (cada aula $a_i$ con $X_i \times Y_i$ sillas), suficientes para que quepan dichos estudiantes. Sin embargo:

   1. La UPC quiere obtener la mayor eficiencia en la gestión de aulas. Dado que el número de estudiantes que finalmente se presentan a un examen es $M \leq N$, la UPC requiere que el número de aulas realmente ocupadas a la hora del examen, $R \leq S$, sea el menor posible.
   2. Los profesores quieren que la ocupación de sillas para diestros/siniestros por parte de estudiantes siniestros/diestros sea mínima.
   3. Los profesores quieren que la posibilidad de copia entre alumnos sea mínima, teniendo en cuenta que se dispone de un factor de confianza, $\alpha_i \in [-1, 1]$, para cada estudiante, $e_i$, tal que cuanto mayor es dicho factor, menos tendencia a copiar tiene el estudiante.

   Una posible solución a éste problema se puede obtener mediante el uso de algoritmos de búsqueda local. 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 y si es mejor/peor respecto a otras alternativas posibles. Justifica tus respuestas.

a) Usar Hill-climbing. Como solución inicial se utiliza el conjunto de las $S$ aulas vacías. Como operadores se usan: *quitar_estudiante(x,y,a)* con aplicabilidad cierta si existe estudiante en la silla $< x, y >$ del aula $a$, y *añadir_estudiante(x,y,a)* con aplicabilidad cierta si la silla $< x, y >$ del aula $a$ está libre.

b) Usar Hill-climbing. Como solución inicial se distribuyen los $M$ estudiantes en las $S$ aulas. Como operadores se usan los mismos que en (a). Como heurístico se usa:

$$Q = card(\{e_i \mid e_i \textit{ en silla adecuada}\})$$
$$h_1(n) = R + Q + \sum_{\forall a_k} \sum_{\forall e_i, e_j \in a_k | e_i \neq e_j} (\alpha_i + 1) * (\alpha_j + 1) * d(e_i, e_j)$$

donde $R$ es el número de aulas con alumnos y $d(e_i, e_j)$ es la distancia euclídea entre las sillas que ocupan los alumnos.

c) Usar Hill-climbing. Como solución inicial se usa la distribución aleatoria de los estudiantes diestros/siniestros en sillas diestras/siniestras de todas las aulas, y los sobrantes se distribuyen aleatoriamente por todas las aulas. Como operador se usa *mover_estudiante(x,y,a,x',y',a')*. Éste se podrá aplicar cuando las sillas $< x, y >$ y $< x', y' >$ sean del mismo tipo. Como heurístico se usa:

$$h_2(n) = \frac{(Q + \sum_{\forall e_i, e_j | e_i \neq e_j} (\alpha_i + 1) * (\alpha_j + 1) * d_1(e_i, e_j))}{R}$$

$$d_1(e_i, e_j) = \begin{cases} d(e_i, e_j) & e_i, e_j \in a_k \\ \infty & \textit{en otro caso} \end{cases}$$

d) Usar Algoritmos Genéticos. Consideramos que cada estudiante se identifica con un número $e_i \in [1 \ldots M]$. Representamos cada configuración posible de conjunto de aulas como una secuencia de números $e_i$. Cada posición representa una silla del conjunto de aulas y el posible estudiante que la ocupa. Las primeras posiciones de la secuencia representan las sillas diestras de todas las aulas, y las últimas posiciones representan las sillas siniestras. Las sillas de $a_i$ aparecen antes que las de $a_{i+1}$. Como operadores se usan un operador de mutación y un operador de cruce por tres puntos fijos: uno dentro de la subsecuencia de sillas diestras, otro en la concatenación de las sillas diestras con las sillas siniestras, y el último punto en la secuencia de sillas siniestras. La función de fitness es la siguiente:

$$h_3(n) = \frac{(Q + \sum_{\forall a_k} \sum_{\forall e_i, e_j \in a_k | e_i \neq e_j} (\alpha_i + 1) * (\alpha_j + 1) * d(e_i, e_j))}{R}$$

# Examen Parcial de IA

**(4 de abril de 2006)**
**grupo 40**
**Duración: 1 hora**

1. (4 puntos) El área de Parques y Jardines del Ayuntamiento de Barcelona quiere modernizar su flota de mini-furgonetas con una nueva, experimental, impulsada por hidrógeno, que tiene la ventaja de ser altamente ecológica y fácil de mantener pero que tiene una autonomía muy limitada. Para resolver el problema de la autonomía se han colocado 3 surtidores de hidrógeno por diferentes partes de la ciudad para que la mini-furgoneta pueda repostar.

   Queremos planificar el recorrido diario de esta mini-furgoneta por los diferentes parques de la ciudad de forma que pueda pasar por todos los parques una vez, tardando el mínimo tiempo posible y pasando por uno de los surtidores cada vez que se le esté acabando el hidrógeno. Como datos para esta planificación disponemos de una tabla que nos indica las posiciones de todos los parques y jardines a visitar y de los tres surtidores, así como la distancia entre cada uno de estos puntos en la ciudad. Sabemos también que la mini-furgoneta experimental tiene un tanque de $n$ litros de hidrógeno y que su consumo es de $x$ litros de hidrógeno por kilometro.

   Se nos plantean las siguientes alternativas:

   *a*) Queremos utilizar A* para minimizar el número de kilometros recorridos por la furgoneta y reducir el número de repostajes al mínimo. Utilizaremos como función de coste la longitud del camino (usando la tabla de distancias antes mencionada), donde la función heurística vale infinito si la mini-furgoneta no tiene suficiente combustible para ir del punto actual a un surtidor, y en caso contrario es la suma de las distancias de los puntos por recorrer al punto actual. El operador aplicable es pasar del punto actual a otro punto.

   *b*) Queremos utilizar busqueda local para minimizar el número de kilometros recorridos por la mini-furgoneta y reducir el número de repostajes al mínimo. Partimos de una solución inicial en la que intercalamos un paso por el surtidor de hidrógeno entre parque y parque. Se dispone de dos operadores de modificación de la solución: uno para eliminar del camino un paso por el surtidor, y otro para modificar el orden en el que visitamos alguno de los puntos del camino (ya sea parque o surtidor). La función heurística es la longitud del camino recorrido + 2 kilómetros extra de penalización por cada paso por un surtidor.

   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.

2. (6 puntos) Se ha de celebrar un congreso multitudinario en la ciudad y los organizadores del evento han de planificar las comidas de los 3000 asistentes. Disponen de una lista de restaurantes y de cada uno de ellos saben:

   - A qué distancia se encuentra del lugar de la celebración del congreso,
   - A cuántas personas pueden dar de comer y
   - El precio del menú.

   Su problema es encontrar una asignación de las 3000 personas a restaurantes de modo que no se supere el tope de personas que admite cada restaurante, se minimice la suma de las distancias recorridas por todos los asistentes y se minimice la cantidad de dinero que la organización va a dedicar al pago de la comida. Se puede suponer que sólo hay que planificar 1 comida.

   Una posible solución a éste problema se puede obtener mediante el uso de algoritmos de búsqueda local. 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 y si es mejor/peor respecto a otras alternativas posibles. Justifica tus respuestas.

*a*) Usar Hill Climbing. Como solución inicial se utiliza una estrategia avariciosa poniendo a las personas en los restaurantes más baratos hasta colocarlas a todas. Como operador se utiliza mover una persona de un restaurante a otro. La función heurística es el producto entre la suma de las distancias recorridas por todas las personas y la suma de los precios de sus comidas.

*b*) Usar Hill Climbing. Como solución inicial se distribuyen aleatoriamente todas las personas entre los restaurantes. Como operador se utiliza mover un grupo de personas de un restaurante a otro, moviendo tantas como quepan en el restaurante destino. La función heurística es la suma de las distancias recorridas por todas las personas multiplicada por una constante $D$ más la suma de los precios de sus comidas multiplicada por una constante $P$.

*c*) Usar Hill climbing. Como solución inicial se utiliza una estrategia avariciosa poniendo a las personas en los restaurantes con el cociente precio/distancia más bajo. Como operador se utiliza mover un grupo de personas de un restaurante a otro, moviendo tantas como quepan en el restaurante destino. La función heurística es la suma para todas las personas del producto entre la distancia recorrida por una persona y el precio de su comida.

*d*) Usar algoritmos genéticos. Representaremos la solución como una tira de bits donde asignamos a cada restaurante tantos bits como se necesiten para representar el número máximo de personas que caben. Utilizamos los operadores habituales de cruce y mutación. La función heurística es la suma de las distancias de todos los restaurantes que tienen alguna persona, dividida por la suma de los precios de las comidas de todas las personas.