IA Examen Parcial

(5 de noviembre de 2012)
grupo 10
Duración: 1 hora

1. (6 puntos) Una empresa chocolatera desea comercializar una nueva caja de bombones que cumpla cierto conjunto de características: el precio ha de ser el mínimo posible, el beneficio que se obtenga ha de ser el máximo posible, ha de tener un peso limitado (no más de 500 g, ni menos de 400 g) y ha de estar compuesta de una combinación de los diferentes tipos de bombones que fabrica, estos pertenecen a K tipos diferentes, cada uno de ellos con un precio, un peso y un beneficio por bombón diferentes. La caja ha de contener no más de N y no menos de M bombones de cada tipo.

   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) Se plantea solucionarlo mediante Hill-climbing utilizando como solución inicial llenar la caja de bombones de un solo tipo hasta llegar al peso mínimo y como operadores añadir un bombón sin pasar del peso máximo y cambiar un bombón por otro de un tipo distinto. La función de evaluación es la suma del precio de los bombones multiplicada por la suma de los beneficios.

   b) Se plantea solucionarlo mediante Hill-climbing utilizando como solución inicial asignar un número aleatorio de bombones de cada tipo dentro del rango [M,N]. Como operadores tenemos cambiar un bombón de un tipo por otro de otro tipo siempre que se mantengan las cantidades máximas y mínimas por tipo y no se violen las restricciones de peso. Se plantea utilizar como función de evaluación de las soluciones la suma del beneficio de los bombones de cada tipo que pasa de M unidades dividida por la suma de precios de los bombones de cada tipo que pasa de M unidades.

   c) Se plantea utilizar algoritmos genéticos donde la representación de la solución es la concatenación de K grupos de N bits cada uno, donde cada bit determina si el bombón está o no en la caja. Las soluciones iniciales se obtienen llenando la caja con M bombones de cada tipo. Se usan los operadores habituales de cruce y mutación. La función de evaluación es la suma de los precios penalizando la solución con el valor $+\infty$ cuando el peso está por encima o por debajo de los límites de peso o se violan los límites de número de bombones por tipo.

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 tiene tres categorías (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 envíos, y en él se puede transportar una carga que quepa, sabiendo que en un espacio de contenedor 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 en los contenedores que se usan 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 categorías de las cargas que quedan por asignar.

   b) Queremos usar satisfacción de restricciones para resolver el problema, para ello elegimos como variables las cargas y para cada carga calculamos su dominio como todos los espacios donde se podrían colocar cumpliendo las restricciones de fechas y de tamaño. Tenemos una restricción entre todas las parejas de cargas, que obliga a que la relación de orden entre las fechas tentativas de entrega de las cargas sea la misma que entre las fechas asignadas, es decir, que los paquetes que se han de enviar antes siempre se envíen antes. Tenemos también otra restricción que obliga a que la suma de los espacios desperdiciados no sea mayor que una constante D.

   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 2012)
grupo 30
Duración: 1 hora

1. (6 puntos) Una empresa de distribución de electricidad necesita calcular cada día qué centrales de producción ha de tener en marcha para abastecer la demanda de sus clientes. La empresa dispone de C centrales de producción eléctrica ubicadas en diferentes puntos de la geografía, con una capacidad de producción CA_i cada una. El coste diario de tener en marcha cada central es CO_i.

   El contrato que tiene la empresa con cada uno de sus CL clientes indica la cantidad de electricidad (E_i) que se les debe suministrar cada día. La distancia entre los clientes y las centrales (d_{ij}) supone una pérdida en la eficiencia del suministro, esta pérdida se calcula como un factor fijo P multiplicado por esa distancia.

   El objetivo es determinar el conjunto de centrales que se han de tener en marcha y a qué clientes han de servir, de manera que se minimice el coste de tenerlas en marcha y la pérdida debida a la distancia, teniendo en cuenta que la cantidad de electricidad que reciben los clientes de una central no ha de sobrepasar su capacidad de producción y que se ha de servir toda la demanda de los clientes. Asumiremos que los valores asignados al problema siempre permiten hallar una solución.

   Una posible solución a este 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) Se plantea solucionarlo mediante Hill-climbing utilizando como solución inicial asignar a cada cliente al azar una central. Como operadores se utiliza el mover un cliente de una central a otra e intercambiar dos clientes entre dos centrales. La función heurística es la siguiente:

   $$h(n) = \sum_{i=1}^{C} (CO_i \times marcha(i)) - \sum_{i=1}^{C}\sum_{j=1}^{CL} asig(i,j) \times d_{ij}$$

   donde $asig(i,j)$ es una función que retorna uno si el cliente i esta asignado a la central j y cero en otro caso y $marcha(i)$ retorna uno si la central tiene clientes asignados y cero si no.

   b) Se plantea solucionarlo mediante Hill-climbing utilizando como solución inicial un algoritmo avaricioso que ordena ascendentemente las centrales según su coste y los clientes según su demanda, y va asignando clientes a las centrales según ese orden. Cuando se sobrepasa la capacidad de una central se pasa a la siguiente. El operador utilizado es intercambiar dos clientes entre dos centrales siempre que no se supere la capacidad de suministro de alguna de ellas. La función heurística es:

   $$h(n) = \left(\sum_{i=1}^{C} CO_i\right) \times \left(\sum_{i=1}^{C}\sum_{j=1}^{CL} P \times asig(i,j) \times d_{i,j}\right)$$

   c) Se plantea utilizar algoritmos genéticos donde la representación de la solución es una tira de bits. Esta está compuesta por la concatenación de C grupos de CL bits, donde cada grupo representa la asignación o no de cada cliente a la central como un 1 o un 0. Las población inicial la obtenemos mediante el mismo mecanismo que en el apartado anterior. Se utilizan los operadores habituales de cruce y mutación. La función heurística es:

   $$h(n) = \left(\sum_{i=1}^{C} CO_i \times marcha(i)\right) + \left(P \times \sum_{i=1}^{C}\sum_{j=1}^{CL} asig(i,j) \times d_{i,j}\right)$$

2. (4 puntos) Tenemos un problema de bioinfomática que consiste en lo siguiente: Tenemos una cadena de ADN de una longitud fija y un conjunto de subcadenas que tenemos que encajar en cualquier posición de la cadena donde coincidan las letras. Cada subcadena tiene una longitud (l_i). El objetivo es buscar el mínimo subconjunto de las subcadenas que permita cubrir la mayor longitud posible de la cadena, sin que haya ningún solapamiento entre las posiciones que cubren las subcadenas escogidas.

   a) Queremos utilizar A*, consideramos el estado como las subcadenas que tenemos asignadas. El estado inicial es no tener ninguna cadena asignada. Tenemos un operador que asigna una subcadena a alguna de las posiciones en las que coincide con la cadena, el coste de este operador es uno. Como función heurística usamos la suma de las longitudes de los trozos de la cadena que no están cubiertos.

   b) Queremos utilizar satisfacción de restricciones. Consideramos que las subcadenas son las variables y los dominios son todas las posiciones donde se pueden encajar las subcadenas en la cadena. Como restricciones tenemos que para cada par de cadenas no puede haber solapamiento entre las posiciones que corresponden a sus asignaciones y que la suma de las longitudes de las subcadenas que estén asignadas sea en la solución mayor que un valor L.

   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 7th 2012)
Time: 1 hour

1. (6 points) A power distribution company needs to compute each day what power stations to use to fulfill the consumption needs of its clients. The company has C power stations geographically distributed with a production capacity of CA_i each. The daily functioning cost of each power station is CO_i.

   The company has a contract with each one of its CL clients that says that the amount of daily energy that has to provide is E_i.

   The distance between each client and the power stations (d_{ij}) results in a loss of efficiency, this loss is computed as a constant factor P multiplied by that distance.

   The goal is to determine what power stations to use and how to assign the clients to these power stations. The total functioning cost of the used stations and the efficiency loss due to the distance has to be minimized. The amount of power needed by the clients assigned to a power station can not exceed its capacity and the power demand of all clients must be fulfilled. We will assume that the values assigned to the problem always guarantee a solution.

   We want to solve this problem as a local search problem. 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 each proposed solutions with respect to whether it is correct, it is efficient, or it is better or worse than other alternatives. Justify your comments.

   a) Using Hill climbing. As an initial solution we assign randomly each client to a power station. As search operators we use to move one client from a power station to another and to swap two clients between two power stations. The heuristic function is:

   $$h(n) = \sum_{i=1}^{C} (CO_i \times open(i)) - \sum_{i=1}^{C} \sum_{j=1}^{CL} assign(i,j) \times d_{i,j}$$

   where $assign(i,j)$ is a function that returns one if the client i is assigned to the power station j or zero otherwise and $open(i)$ is a function that returns one if the power station has at least one client assigned and zero otherwise.

   b) Using Hill climbing. As an initial solution we use a greedy algorithm that first sorts in ascending order the powers stations by their cost and the clients by their demand and then assigns clients to power stations in that order. When the capacity of a power station is exceeded we move to the next one. The search operator is to swap two clients between two power stations if no capacity is exceeded. The heuristic function is:

   $$h(n) = \left(\sum_{i=1}^{C} CO_i \right) \times \left(\sum_{i=1}^{C} \sum_{j=1}^{CL} P \times assign(i,j) \times d_{i,j} \right)$$

   c) We use genetic algorithms. The representation of the solution is a string of bits. This string is composed by the concatenation of C groups of CL bits, where each group represents for each client if it is assigned to that power station as a 1 or a 0. The initial population is computed using the same method as in the previous solution proposal. The usual crossover and mutation operators are used. The heuristic function is:

   $$h(n) = \left(\sum_{i=1}^{C} CO_i \times open(i) \right) + \left(P \times \sum_{i=1}^{C} \sum_{j=1}^{CL} assign(i,j) \times d_{i,j} \right)$$

2. (4 points) We have the following bioinformatics problem: We have a string representation of a DNA strand of a specific length and a set of substrings that we have to place in any position of the string with the same letters. Each substring has a length (l_i). The goal is to find the minimum subset of substrings that covers the most of the string but without any overlapping among the positions that are covered by the substrings.

   a) We use A*, we consider as the state the substrings that are assigned. The initial state is the empty state. We have an operator that assigns a substring to any of the positions where its letters match with the string, the cost of the operator is one. As heuristic function we use the sum of the lengths of the parts of the string that are still not covered.

   b) We use constraints satisfaction. We consider that the substrings are the variables and the domains are all the positions where the substring matches with the string. As a constraints we have that for each pair of strings it is forbidden that they overlap when assigned to their matched positions on the string and that the sum of the lengths of the substrings that are assigned in the solution has to be larger that L.

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