Examen Parcial de IA

(5 de noviembre de 2019)
grupo 10
Duración: 90 min

1. (4 puntos) Una empresa de productos orgánicos nos pide un sistema inteligente para distribuir un conjunto de plantas en un cultivo. Para organizar mejor el cultivo se ha dividido en una cuadrícula de $N \times M$ posiciones. Las plantas que quieren colocar son de T diferentes tipos y tienen 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.

   El sistema inteligente ha de colocar todas las plantas, 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 cualquiera de sus adyacentes no supere cierto valor C.

   Tras un análisis inicial del problema un compañero de nuestra empresa nos plantea dos estrategias distintas para resolverlo:

   a) Usar el algoritmo de A*. El estado es la asignación de plantas a las casillas de la cuadrícula. Usamos como operador el añadir una planta de un tipo a una posición de la cuadrícula siempre que no se supere el número P de plantas en la posición y no se superen los consumos de agua y nutrientes indicados en el enunciado. El coste del operador es la suma de necesidades de agua y nutrientes de la planta colocada. La función heurística es la suma de agua y nutrientes de las plantas que quedan por colocar o infinito si en la solución actual hay alguna posición en la que la diferencia de consumo entre una posición y cualquiera de sus adyacentes es mayor que C.

   El algoritmo A^* está pensado para encontrar la solución óptima y no solo para maximizar o minimizar según uno o más criterios. Utilizar un algoritmo con un coste tan elevado en un problema en el que no se requiere encontrar el óptimo es una mala decisión, ya que introducimos una complejidad algorítmica innecesaria.

   La definición del estado para resolverlo mediante A* es correcta, la asignación de plantas a casillas de la cuadrícula. El camino lo compondrían la secuencia de asignaciones de plantas a las casillas. Sin embargo, esta representación es demasiado sintética, y haría falta añadir a la representación del estado información auxiliar sobre el estado de nutrientes por casilla o los litros que quedan por columna para facilitar el cálculo de las condiciones a comprobar.

   El operador permite avanzar hacia una solución asignando plantas a posiciones, y mantiene la restricción de que no haya mas de P plantas en una casilla y comprueba que no se superen las restricciones de consumo de agua y nutrientes. No se menciona la restricción sobre la diferencia con las casillas adyacentes, pero se tiene en cuenta en la función heurística al hacerse infinito cuando se viola esa restricción. El factor de ramificación es $O(T \times P \times N^2)$.

   El coste del operador es el consumo de agua y nutrientes. Dado que el problema no pide que se optimice ninguna de las características del problema este criterio es tan válido como cualquier otro. Al tener unidades distintas (litros y gramos) quizás se debería añadir una ponderación.

   La función heurística es a priori admisible ($h = h^* =$ el coste de las plantas que nos quedan por colocar). Teniendo en cuenta que vamos a colocar todas las plantas, el coste de todas las soluciones será el mismo, por lo que la suma del coste de los operadores (g) con la estimación de lo que que queda (h) da siempre una constante, aportando un guiado nulo al algoritmo A* que provocará una búsqueda en anchura ordenada por los consumos más

grandes primero (ya que reducen más el valor de h, que se usa para desempatar entre todas las f iguales).

El heurístico controla la restricción respecto a las celdas adyacentes evitando continuar la exploración de los caminos que no respetan esta restricción

En definitiva, esta solución es equivalente a hacer una búsqueda en anchura, no se puede beneficiar de las características de A* dado que no se pretende optimizar nada.

b) Usar un algoritmo de satisfacción de restricciones. El grafo de restricciones tendría como variables 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.

La aplicación de un algoritmo de satisfacción de restricciones podría hallar una solución al problema, ya que no se pide la optimización de ninguno de los criterios.

Como representación se plantea elegir las posiciones como variables, Con este planteamiento hay relativamente pocas variables, pero los dominios de todas las variables serán enormes (el conjunto de todos los posibles subconjuntos de identificadores de planta de tamaño 0 hasta $NumeroTotalPlantas$, de forma que al reducir el dominio a un valor corresponde al subconjunto de plantas que asignaríamos a esa casilla). Podríamos reducir el tamaño de los dominios si consideramos el tipo de planta en lugar de su identificador, pero tendríamos que controlar que no asignemos mas que el número que tenemos para cada tipo (con la solución propuesta se supone que eso se ha controlado al crear los identificadores de plantas).

Todas las restricciones que se plantean son las que debe satisfacer la solución:

- R1: las columnas no deben superar el límite L de consumo de agua. Esta restricción sería global entre todas las variables que pertenecen a casillas de la misma columna. Al necesitar comprobar un límite superior ($< L$) puede ser usada durante la asignación de variables.
- R2: la asignación no debe superar el consumo máximo G de nutrientes. Esta restricción iria desde cada variable hasta sí misma, cosa que PSR no puede usar. Por ello la forma de incorporarla seria haciendo un pre-procesado de los dominios de todas las variables, eliminando los subconjuntos de plantas cuyo consumo supere G;
- R3: La diferencia de consumo de una posición con las adyacentes no debe tampoco superar el límite C. En este caso esto se puede hacer con restricciones binarias que van desde una casilla a cada una de sus adyacentes, lo cual es muy corrrecto para PSR.

Hay además dos condiciones más que plantea el enunciado y que la solución propuesta no tiene en cuenta:

- R4: se puede colocar un máximo de P plantas en una casilla. Al igual que en el caso de G, seria una restriccion hacia la propia variable de casilla. Por ello la forma de incorporarla seria modificando los dominios de todas las variables de forma que solo se creen los subconjuntos de 0 a P plantas posibles (y solo los que cumplen G).
- R5: un identificador de planta solo puede aparecer asignado a una sola casilla. Un ejemplo de conocimiento de sentido común que hay que añadir al algoritmo para que no nos devuelva resultados que no nos interesan. En este caso la forma de introducirlo seria con un grafo completo de restricciones binarias entre todo par de variables de forma que no puedan tener ninguna planta compartida.

Como se puede ver el planteamiento propuesto es muy complejo, y no hay otros planteamientos más simples que funcionen. Por ejemplo, otro planteamiento sería suponer que las

plantas son las variables y su dominio son las $N \times M$ posiciones a las que podriamos asignarlas. En este caso tenemos un conjunto aceptable de variables con dominios de tamaño correcto. Pero en esa representación se complicaría mucho representar las restricciones sobre G, P, L y C, que acabarían siendo todas globales y difíciles de calcular.

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.

2. (6 puntos) Los alcaldes de las poblaciones del Baix Llobregat se han unido para conseguir el compromiso de la Generalitat de construir una nueva línea de metro que pueda servir a zonas que aún no cubre la red actual de metro y trenes. Para realizar el proyecto los ingenieros del Area Metropolitana de Barcelona han pre-seleccionado L lugares potenciales donde construir las estaciones de metro y han determinado, a partir del análisis de los datos de movilidad, el número de usuarios diarios que usaría cada estación potencial. Pero por limitaciones presupuestarias, la línea de metro solo puede tener E estaciones, a escoger entre los L lugares potenciales. El objetivo es determinar qué E lugares escoger para las estaciones y el orden en el que se han de conectar de manera que se sirva al mayor número de personas y el recorrido total de la línea sea el menor posible.

   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 elegimos una solución sin estaciones. Como operadores de búsqueda usamos `añadir-estación`, que añade una estación al final del recorrido de la solución e `intercambiar-estaciones` que intercambia dos estaciones dentro del recorrido. La función heurística es:

$$h(n) = \sum_{i=1}^{E-1} dist(E_i, E_{i+1}) + \sum_{i=1}^{E} Personas(E_i)$$

   Donde $dist(E_i, E_{i+1})$ es una función que indica la distancia (en metros) entre la estación E_i y la siguiente, y $Personas(E_i)$ es una función que devuelve el número de usuarios diarios que usaría la estación E_i.

Plantear como solución un Hill Climbing para este problema es adecuado a priori, ya que nos piden encontrar una solución maximizando o minimizando una serie de criterios sin necesidad de obtener el óptimo.

La solución inicial planteada no es solución (la solución vacia no cumple el tener E lugares seleccionados como estaciones). El coste de generación es $O(1)$ pero su calidad es mala, dejando todo el trabajo de construir una primera solución a la búsqueda. De tener una buena combinación de operadores y heurístico podría compensarse este problema, pero no es el caso. Además para este problema no es difícil construir soluciones iniciales mejores (al azar, que tengan E lugares seleccionados) con costes lineales, por lo que no es recomendable empezar con la solución vacía.

El operador `añadir-estación` no comprueba que no pongamos más de E estaciones, además no explora todas las posibles soluciones al restringir en que posición se puede poner la nueva estación. Esto último se podría dar como correcto dado que reduce el factor de ramificación ($O(L)$) y el otro operador permite generar las opciones que se evitan.

El operador `intercambiar-estaciones` es correcto. Al no alterar el número de estaciones en la solución, ésta seguirá siendo solución si lo era. El factor de ramificación es $O(E^2)$.

El problema de los operadores escogidos es que, aunque en teoria nos permiten explorar todo el espacio de soluciones, la exploración se ve limitada por el hecho de que una estación no se puede eliminar de la solución una vez ha sido añadida.

La función heurística multiplica un criterio que queremos maximizar (*personas*) por otro que queremos minimizar (*distancia*), y por ello no nos sirve ni maximizarla ni minimizarla. Lo mejor sería plantear una resta entre el sumatorio de personas y el de distancias, y como tienen diferentes unidades (individuos y metros) que esta resta fuera ponderada por un factor corrector. La otra cosa necesaria en esta función heurística es añadir una penalización a las no-soluciones, relacionada con el número de estaciones que faltan o sobran en la solución.

b) Usar Hill climbing. Como solución inicial escogemos una estación al azar, usamos esta estación como referencia y escogemos como siguiente estación la más cercana a esta, pasamos a tomar como referencia esta nueva estación y buscamos la más cercana que no esté en la solución, repetimos lo mismo hasta tener E estaciones. Como operadores de búsqueda usamos el `cambiar-estación` que intercambia una estación de la solución por otra que no esté en la solución e `intercambiar-estaciones` que intercambia dos estaciones dentro del recorrido. La función heurística es:

$$h(n) = \frac{\sum_{i=1}^{E-1} dist(E_i, E_{i+1})}{\sum_{i=1}^{E} Personas(E_i)}$$

Como se ha dicho en el partado anterior, a priori Hill Climbing es adecuado para este tipo de problema.

La solución inicial ahora si es solución al cumplir las restricciones (tener E estaciones), pero tiene una calidad indeterminada (aunque intenta minimizar distancia no garantiza calidad respecto a las personas servidas) El coste de generación es $O(L^2)$ (o si se quiere ser más preciso, $O(L \times E)$, ya que hay que escoger E veces un lugar entre los L posibles).

En este caso la combinación de operadores cubre todo el espacio de soluciones, ya que podemos introducir lugares que no estan en la solución con el operador `cambiar-estación` y generar las reordenaciones necesarias de los lugares dentro del recorrido con el operador `intercambiar-estaciones`.

El operador `cambiar-estación` no comprueba restricciones, pero al mantener siempre el número E de estaciones en la solución nos matiene dentro del espacio de soluciones. El factor de ramificación es $O(E \times L)$.

El operador `intercambiar-estaciones`, al igual que en el apartado anterior, no altera el número de estaciones en la solución, y por lo tanto nos mantiene dentro del espacio de soluciones. El factor de ramificación vuelve a ser $(O(E^2))$.

En el heurístico se combinan los dos criterios del enunciado, las personas (a maximizar) y la distancia (a minimizar). Al colocar la distancia en el numerador y las personas en el denominador, en este caso este heurístico que se propone se ha de minimizar. El problema es que este cociente puede darnos valores similares con un número de personas bajos y distancias reducidas o con números de personas elevados y distancias enormes, y es difícil de controlar la influencia de cada uno de los criterios. Por ello sería mejor una resta ponderada de personas menos distancia. En este caso el heurístico no necesita añadir una penalización a las no soluciones, ya que en esta propuesta siempre se está dentro del espacio de soluciones.

c) Se plantea utilizar algoritmos genéticos. Asociamos a cada lugar posible con un número del 1 al L. La codificación de la solución se realiza considerando que tenemos una tira de bits de longitud $E \times \log_2(L)$, donde concatenamos los identificadores de los lugares que hay en la solución, el orden en el que aparecen en la codificación es el orden del recorrido. Para generar la población inicial se escogen E lugares al azar de los L y se colocan al azar en la

solución. Como operadores genéticos usamos los operadores habituales de cruce y mutación. La función heurística es:

$$h(n) = \alpha \times \sum_{i=1}^{E-1} dist(E_i, E_{i+1}) + (1 - \alpha) \times (-\sum_{i=1}^{E} Personas(E_i))$$

Donde α es un valor entre 0 y 1 que permite dar más o menos importancia a cada criterio.

Plantear como solución un algoritmo genético para este problema es adecuado a priori, ya que nos piden encontrar una solución maximizando o minimizando una serie de criterios sin necesidad de obtener el óptimo.

La representación escogida permite representar cualquier solución del problema (un recorrido compuesto por E identificadores de lugar), pero también permite representar muchas no-soluciones (identificadores de lugar repetidos en un mismo recorrido, identificadores de lugar inválidos si L no es una potencia de 2 y por lo tanto $log_2 L > L$).

La función generadora de la población inicial escogida al parecer solo genera un individuo, pero al tener un comportamiento aleatorio si la llamamos tantas veces como individuos queramos en la población inicial nos dará individuos diferentes, con una buena variabilidad, y todos serán solución ya que tendrán E identificadores de lugar. El coste de generación será $O(N \times E)$, siendo N el tamaño escogido de la población inicial.

Los operadores de cruce y mutación estandar escogidos, dada la posibilidad de que la representación codifique no-soluciones, pueden sacarnos del espacio de búsqueda fácilmente (en ambos casos tanto el cruce como la mutación pueden crear identificadores duplicados de lugar, o identificadores inválidos). Y el heurístico no lo controla adecuadamente, como veremos a continuación.

La función de evaluación propuesta por fin plantea una resta ponderada que deberíamos minimizar, pero eso no es adecuado como función de fitness de un algoritmo genético (busca siempre maximizar la función de fitness). Por ello se debería girar la resta para que sea el sumatorio de personas menos el de distancias. En este caso faltaría también añadir una penalización para las no-soluciones asociada al número de identificadores repetidos y el número de identificadores inválidos (y modificar las funciones $Personas(E_i)$ y $dist(E_i, E_{i+1})$ para que no den error al pasarles como parámetro un identificador E_i inválido).

Las notas se publicarán el día **25 de noviembre**.