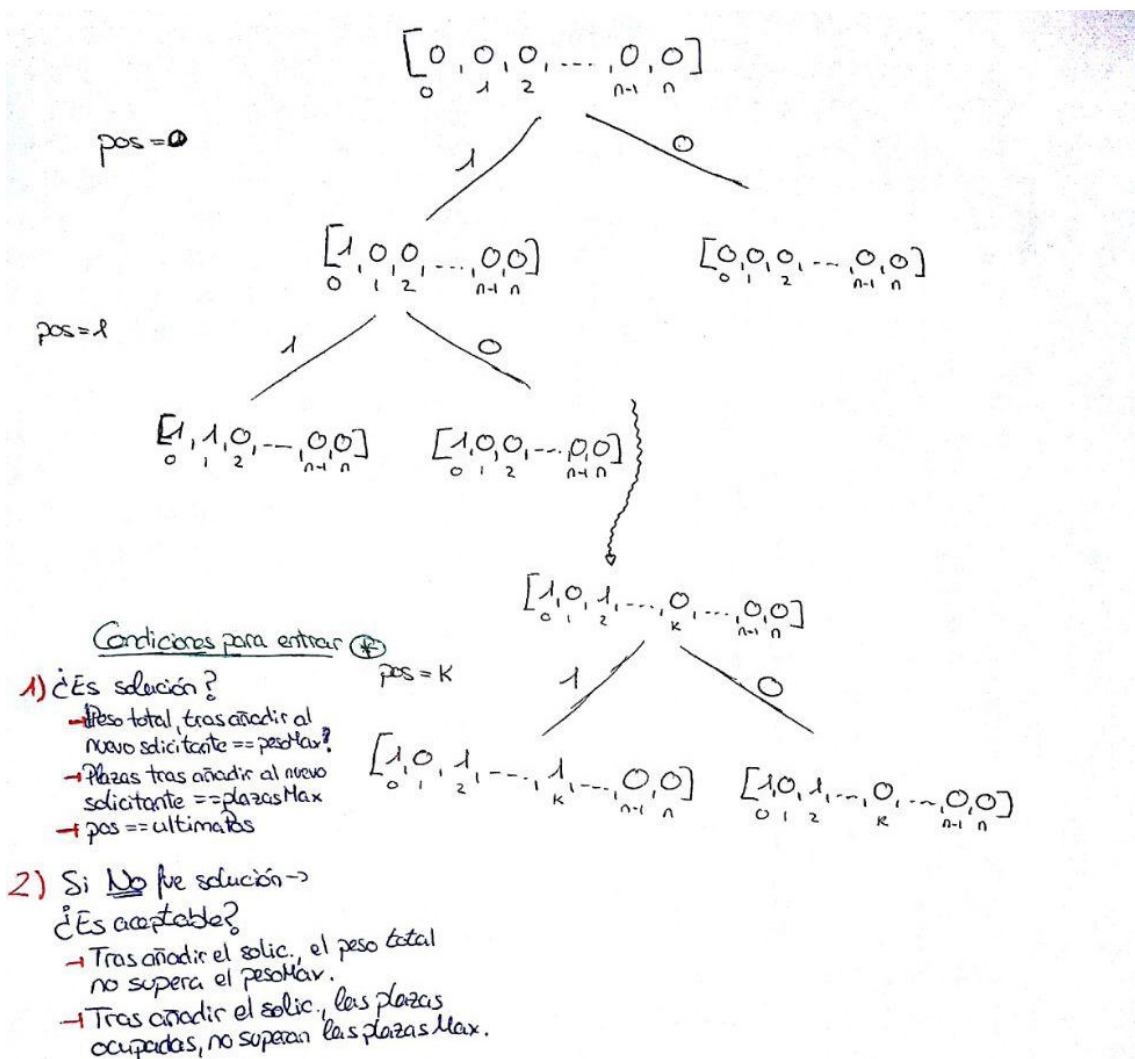


DOCUMENTACION DA

PRACTICA 2

BACKTRACK 01

- Árbol de ensayos:



El ensayo inicial será un array de booleanos, con el tamaño de la lista de solicitantes completa (n).

Estos booleanos en un inicio estarán a 'false' (0), que implica que no hay ninguna plaza asignada para ningún solicitante, pero conforme se avance en el árbol y en la lista de solicitantes (pos), estos valores se irán actualizando a medida que añadamos nuevos solicitantes a nuestro posible ensayo óptimo.

El proceso que sigue la evaluación de un solicitante será: preguntar primero si añadiéndolo obtenemos una posible solución, es decir, que llegamos al límite de plazas, de peso, o de la lista.

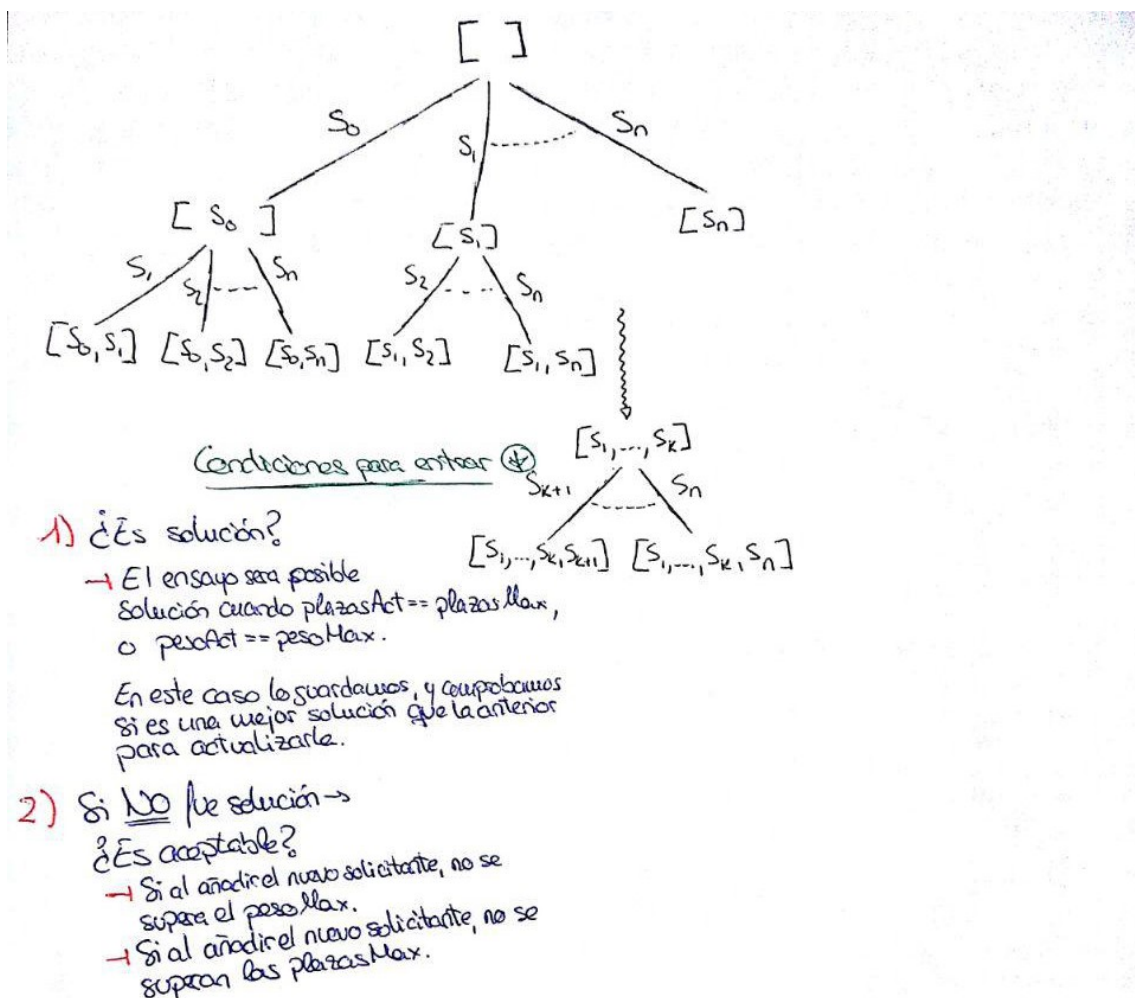
En caso de ser solución se comparará con la mejor solución hasta ahora, y se actualizará la información de ser necesario. Y en caso de no ser una posible solución el ensayo con el nuevo solicitante, comprobamos si es aceptable que entre al ensayo.

Pero no podrá entrar cualquier solicitante al ensayo, ya que el problema está limitado en número de plazas máximo, y peso máximo, que puede llevar el avión. Las plazas totales no podrán superar 'plazasMax', por lo tanto a la hora de evaluar un nuevo solicitante habrá que tener en cuenta cuantos tenemos ya añadidos; y el peso total no podrá superar 'pesoMax'. Además, se añade otro control, que será en caso de estar evaluando al último solicitante de nuestra lista, como posible entrada, lo añadamos o no, habremos llegado a una posible lista óptima, ya que no hay más elementos a evaluar.

He de añadir, que además de los casos de entrada de cada solicitante, también están las ramas con 0, que simulan no coger el solicitante, por lo tanto no alteran apenas la situación.

BACKTRACK N-ario

- Árbol de ensayos:



En este caso los ensayos estarán formados de objetos 'Solicitante', y el inicial estará vacío.

Además en este caso, desde el ensayo inicial tendremos disponible todo el abanico de nodos para decidir desde cual queremos empezar; pero hay que tener en cuenta que una vez seleccionado un solicitante 'k', solo podremos acceder a los siguientes desde 'k+1' hasta 'n' para rellenar el ensayo, no será posible coger otro solicitante que quedó atrás.

En este caso el ensayo en vez de ser del tamaño de la lista de solicitantes, tendrá un tamaño de 'plazasMax' y eso nos marcará uno de los límites del problema; mientras que el otro límite lo marcará 'pesoMax' que también lo evaluaremos conforme se vayan a añadir los solicitantes.

Los criterios de entrada para los nuevos solicitantes serán los siguientes: Primero miraremos si al añadir ese nuevo solicitante a la lista se forma una posible solución, esto es, que se llene por completo el ensayo con todas las plazas posibles ocupadas o que se iguale el peso actual al añadir el nuevo solicitante al pesoMax; o que sea el ultimo solicitante de la lista entre o no.

En caso de no cumplir alguno de estos criterios, el ensayo aun no es una posible solución, y se evaluará si el nuevo solicitante es aceptable para entrar a la lista.

Igual que antes, tendrá que cumplir ciertos requisitos, que son: al añadir el nuevo solicitante no superamos las 'plazasMax', y que tampoco que supere el 'pesoMax'.

Además he de añadir que en los dos algoritmos para hacer una mejor poda, se han ordenado previamente los solicitantes conforme a sus pesos, de tal forma que en cuanto el que estemos estudiando no entre por exceso de peso, no miraremos ninguno mas ya que sabemos que tampoco entran.

MUESTRAS DE TIEMPOS:

- Lista de 10 solicitantes:
 - Dinámica: 67 ms.
 - Backtrack 01: 9ms.
 - Backtrack 1N: 6 ms.

- Lista 30 solicitantes:
 - Dinámica: 70ms.
 - Backtrack 01: 154ms.
 - Backtrack 1N: 94ms.

- Lista 80 solicitantes:
 - Dinámica: 100ms.
 - Backtrack 01: >15 mins.
 - Backtrack 1N: >15 mins.

- No he hecho mas pruebas con los de backtrack, ya que son algoritmos muy costosos, sin embargo el de dinámica aun aguanta bastantes solicitantes mas.
 - Lista de 300 solicitantes: 118 ms.
 - Lista de 500 solicitantes: 123ms.
 - Lista de 2500 solicitantes: 308ms.
 - Lista de 4000 solicitantes: 448ms.
 - Lista de 5000 solicitantes: StackOverflowError.