Programación Dinámica - Noviembre 2015

Número de decisiones : S

Significado de la decisión i-ésima: secuencia de decisiones $<d_1, d_2 \dots d_S>$

d_1 : número de solicitudes atendidas en la sede 1.

d_2 : número de solicitudes atendidas en la sede 2.

. . .

d_i : número de solicitudes atendidas en la sede i-ésima.

Habrá un número de alternativas variable para la solución i-ésima, que dependerá de Sol[i], que es el nº de solicitudes en la sede i. También se considera que un monitor solo puede como máximo atender 20 solicitudes, en cada sede. $d_i \in \{0, 1, 2 \dots, \min(Sol[i], 20)\}$

Función objetivo y restricciones: maximizar $\sum_{d=1}^{S} d_i$ $\quad \sum_{i=1}^{S} FACT[i][d_i] \le E$

Demostración del principio de Optimalidad:

Sea $<d_1, d_2, \dots d_S>$ la secuencia óptima de decisiones para el problema Gimnasio (S, E), siendo su valor (número de pet. atendidas) asociado:

$$\sum_{i=1}^{S} d_i \quad \text{sujeto a} \quad \sum_{i=1}^{S} FACT[i][d_i] \le E$$

Vamos a prescindir de la última decisión, es decir d_S. La subsecuencia que resulta es, $<d_1, d_2, \dots d_{S-1}>$, es la solución óptima del subproblema Alumnos (S-1, E - FACT[S][d_S]) cuyo valor es:

$$\sum_{i=1}^{S-1} d_i \quad \text{sujeto a} \quad \sum_{i=1}^{S-1} FACT[i][d_i] \le E - FACT[S][d_S]$$

Supongamos que $<d_1, d_2, \dots d_{S-1}>$ no es la solución óptima del problema Alumnos (S-1, E - FACT[S][d_S]), sino que hay otra subsecuencia, $<d_1^*, d_2^*, \dots d^*_{S-1}>$ que es mejor que esta, que mejora el resultado para el subproblema Alumnos (S-1, E - FACT[S][d_S]), en este caso se cumpliría:

$$\sum_{i=1}^{S-1} d_i < \sum_{i=1}^{S-1} d_i^* \quad \text{y además}$$

$$\sum_{i=1}^{S-1} FACT[i][d_i^*] \le E - FACT[S][d_S]$$ por tanto la secuencia cumple

$$\sum_{i=1}^{S-1} d_i + d_S \le \sum_{i=1}^{S-1} d_i^* + d_S$$

por lo que mejora el valor de la secuencia $<d_1, d_2, \dots d_S>$. En consecuencia, ésta no sería la solución óptima del problema Alumnos (S, E), en contra de lo supuesto inicialmente.

Luego, se cumple el principio de Optimalidad.

Definición recursiva y primera llamada a la función:

$$Alumnos(S,E) = \begin{cases} 0 & \text{si } S = 0 \\ \max\limits_{0 \le d_S \le \min(Sol[S], 20) \,/\, E - FACT[S][d_S] \ge 0} \{ Alumnos(S-1, E - FACT[S][d_S]) + d_S \} & \text{si } S > 0 \end{cases}$$

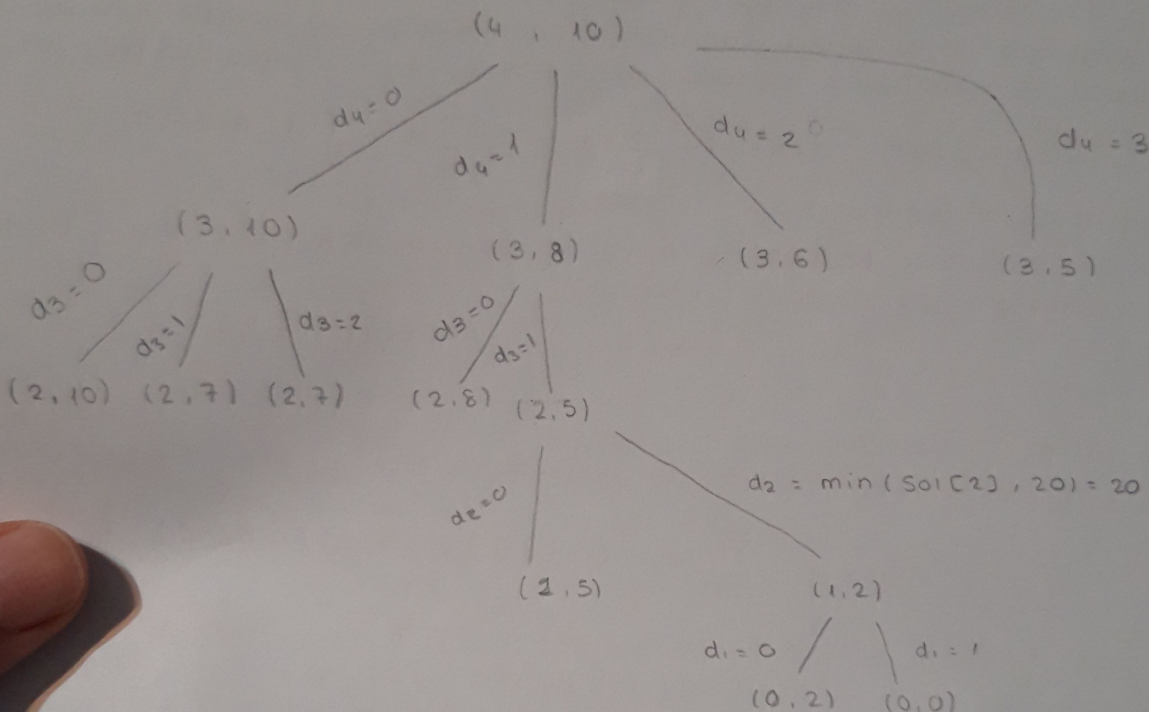
La primera llamada a la función Alumnos (S, E) se realiza con S, nº de sedes total y E presupuesto disponible inicialmente.

Arbol de llamadas para un ejemplo.

S = 4 , E = 10 , Sol[1..4] = {15, 32, 2, 3} y FACT[1..4][0..20]

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	0	2	9	2																	4
2	0	2	1	1																	3
3	0	3	3	8			...							...							5
4	0	2	4	5																	9

d_1



Tipos de estructuras necesarias, dimensión y forma de relleno.

Dado que la función recursiva tiene dos parámetros se precisan dos estructuras de almacenamiento. Éstas tendrán E + 1 y S + 1 filas y columnas ya que $1 \le e \le E$ y $1 \le s \le S$.

En una de las estructuras se guarda el valor asociado a cada subproblema, osea el máximo de solicitudes atendidas. En la otra estructura se guardará la alternativa que proporcione el máximo para cada subproblema.

Las estructuras de almacenamiento se inicializan con los resultados de los problemas triviales, que corresponde a llenar la fila de cada matriz con cero ya que los prob. son del tipo Alumno(0, e)

Para el relleenado de cada matriz: dado que para solucionar el subproblema Alumnos(s, e) se precisa conocer la solución de los subproblemas tipo Alumno(s-1, x) con ambas matrices se rellenan por filas en sentido ascendente.