Explicación a detalle

Viu

Organizar sesiones de doblaje

BRAYAN SEBASTIAN SALAS RICO

2025

**Explicación a detalle**

Definición del problema:

Se precisa coordinar el doblaje de una película. Los actores del doblaje deben coincidir en las tomas en las que sus personajes aparecen juntos en las diferentes tomas. Los actores de doblaje cobran toda la misma cantidad por cada día que deben desplazarse hasta el estudio de grabación independientemente del número de tomas que se graben. No es posible grabar más de 6 tomas por día. El objetivo es planificar las sesiones por día de manera que el gasto por los servicios de los actores de doblaje sea el menor posible. Los datos son:

Numero de actores: 10

Numero de tomas: 30

Tomas máximas por día: 6

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Actores** | | | | | | | | | |  |
| **Tomas** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **Total** |
| **1** | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 5 |
| **2** | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 3 |
| **3** | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 3 |
| **4** | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 4 |
| **5** | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 3 |
| **6** | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 4 |
| **7** | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 4 |
| **8** | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 3 |
| **9** | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 3 |
| **10** | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 4 |
| **11** | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 5 |
| **12** | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 5 |
| **13** | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 3 |
| **14** | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 3 |
| **15** | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 3 |
| **16** | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 2 |
| **17** | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 |
| **18** | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 2 |
| **19** | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 |
| **20** | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 4 |
| **21** | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 2 |
| **22** | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 4 |
| **23** | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 |
| **24** | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 2 |
| **25** | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 4 |
| **26** | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 4 |
| **27** | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 2 |
| **28** | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 2 |
| **29** | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 3 |
| **30** | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 2 |
| **Total** | 22 | 14 | 13 | 15 | 11 | 8 | 3 | 4 | 2 | 2 |  |

Glosario

A: total de actores.

D: total de días.

T: total de tomas.

a: actores.

d: días.

t: tomas.

td: tomas máximas por día.

Nota:

Si bien las respuestas entregadas tanto en este documento como en el documento notebook son con formulas generales o basadas en el ejemplo original (10 actores, 30 tomas), hago un ejemplo mas pequeño con el cual es más fácil llegar a conclusiones y explicaciones.

Ejemplo simplificado:

El siguiente ejemplo lo que busca es minimizar la complejidad del ejercicio inicial para llegar a fórmulas y conclusiones generalizadas.

Numero de actores: 5

Numero de tomas: 4

Tomas máximas por día: 2

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Autores** | | | | | |
| **Tomas** | 1 | 2 | 3 | **4** | **5** | **Total** |
| **1** | 1 | 1 | 0 | 0 | 0 | 2 |
| **2** | 0 | 1 | 1 | 1 | 0 | 3 |
| **3** | 1 | 0 | 1 | 0 | 0 | 2 |
| **4** | 1 | 1 | 0 | 1 | 1 | 4 |
| **Total** | 22 | 14 | 13 | 15 | 11 |  |

Dada la información anterior podemos decir que las posibles combinaciones son:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Dia 1 | | Dia 2 | | Dia 3 | Dia 4 |
| t1 | t2 | t3 | t4 | - | - |
| t1 | t3 | t2 | t4 | - | - |
| t1 | t4 | t2 | t3 | - | - |
| t1 | t2 | t3 | | t4 | - |
| t1 | t3 | t2 | | t4 | - |
| t1 | t4 | t2 | | t3 | - |
| t2 | t3 | t4 | | t1 | - |
| t2 | t4 | t1 | | t3 | - |
| t3 | t4 | t1 | | t2 | - |
| t1 | | t2 | | t3 | t4 |

Y que su ponderación (suponiendo que a cada actor le paguen 10) según el esquema anterior sería el siguiente:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Dia 1 | Dia 2 | Dia 3 | Dia 4 | TOTAL |
| 40 | 50 | - | - | 90 |
| 30 | 50 | - | - | 80 |
| 40 | 40 | - | - | 80 |
| 40 | 20 | 40 | - | 100 |
| 30 | 30 | 40 | - | 100 |
| 40 | 30 | 20 | - | 90 |
| 40 | 40 | 20 | - | 100 |
| 50 | 20 | 20 | - | 90 |
| 50 | 20 | 30 | - | 100 |
| 20 | 30 | 20 | 40 | 110 |

Según la información de la tabla anterior, las mejores opciones serian:

[(t1, t3), (t2, t4)] o [(t1, t4), (t2, t3)]

Entendiendo cada tupla como un día (ósea en total dos días), y cada T como toma.

Función objetivo:

Dada la información obtenida por el anterior ejemplo y el ejemplo original, podemos decir que:

Entendiendo que, lo que buscamos es minimizar la suma total de todas las combinaciones actor-día en las que el actor se presente a grabar.

a: Es cada actor iniciando desde el 1.

A: Es la cantidad máxima de actores.

d: cada uno de los días iniciando desde 1.

D: el total de días de grabación, por lo general suelen ser menos que las tomas, sin embargo, entendiendo que podría ser una toma por día, la cantidad máxima de dias va a ser igual a la cantidad máxima de tomas (D = T).

y: Es una variable binaria, que indica si el actor a asiste el día d.

Para el caso simplificado:

Para el caso original:

Complejidad de algoritmo por fuerza bruta:

El algoritmo genera todas las opciones posibles de conjuntos, pero entendiendo que el ejercicio tiene subconjuntos, y que a su vez estos no tienen una longitud fija, si no que varia entre 1 y 6 (cantidad de tomas al dia), lo cual también debe tenerse en cuenta en las posibilidades. Consecuencia de esto, el algoritmo cuenta con diferentes for anidados, que hacen aumentan el orden de complejidad.

El algoritmo sería **O(n!)** el orden mas grande visto en clase.

Complejidad de algoritmo por ramificación y poda (Branch and bound):

La complejidad en el mejor de los casos puede ser lineal O(n), entendiendo el mejor de los casos como que el algoritmo encuentre en primera instancia la mejor combinatoria (esto no suele ocurrir), en el peor de los casos puede llegar a ser de orden exponencial (2^n), esto varia mientras más se demore el algoritmo en encontrar una combinación satisfactoriamente baja (en el caso de minimización) para descartar rápidamente las siguientes.

Peor caso:

Mejor caso: