Optimización de Cronograma de Rodaje para un Programa de Televisión o Película

Ana Melissa Alonso Reina C-412 Alejandro Ramírez Trueba C-411

February 10, 2025

1 Introducción

Este proyecto se enfoca en la planificación y gestión integral de todos los recursos necesarios para rodar un programa de televisión o una película de manera eficiente y ajustada al presupuesto. Para lograrlo, se consideran múltiples factores que suelen complicar la producción:

- **Disponibilidad de actores:** Cada actor o actriz tiene un horario limitado que puede solaparse con otras actividades o producciones, por lo que la asignación de escenas debe coincidir con su presencia.
- **Restricciones de localizaciones:** Muchas locaciones están disponibles solo ciertos días o requieren permisos especiales, lo que obliga a programar secuencias específicas durante ese margen. Además, las localizaciones deben ser usadas en el momento del día que corresponde con las escenas que se filman en ellas, porque si una escena es en el patio de la escuela al atardecer no se puede filmar por la noche.
- **Continuidad narrativa:** Algunas escenas deben rodarse antes que otras para mantener la coherencia de la trama; esto implica respetar un orden parcial en la grabación.
- **Presupuesto fijo:** A menudo, los gastos totales no pueden superar un monto predeterminado, por lo que cualquier cambio en la planificación debe considerar su efecto sobre el costo final. También se puede considerar el gasto de combustible al dirigirse de una localización a otra para filmar en más de una en el mismo día.

La meta principal es minimizar el tiempo total requerido para completar el rodaje y evitar sobrecostos, todo ello manteniendo la calidad de la producción y cumpliendo con los plazos de entrega pactados. Para abordar estos retos, se pueden proponer distintos enfoques, como la aplicación de técnicas de programación dinámica, metodologías de divide y vencerás que segmenten el problema en episodios o actos, o algoritmos codiciosos que prioricen la filmación de escenas con mayor complejidad y valor de producción. Cada enfoque tendrá sus fortalezas y limitaciones, pero todos deben lidiar con la complejidad combinatoria que nace de la gran cantidad de restricciones interconectadas.

Este problema es de gran importancia práctica en el ámbito audiovisual, ya que la correcta organización de recursos humanos y técnicos puede ahorrar miles de dólares al evitar retrasos o

cambios de última hora. Además, garantizar que todos los elementos necesarios (actores, locaciones, equipo) estén disponibles en el momento oportuno reduce la probabilidad de rehacer escenas o duplicar esfuerzos. Por otra parte, la industria audiovisual trabaja a menudo bajo calendarios muy ajustados, por lo que toda mejora en la eficiencia del programa de rodaje se traduce en grabaciones más ágiles y un mayor cumplimiento de las fechas de entrega.

Paper 1

2 Problema de Programación de Filmación (FS).

En la producción de películas, uno de los retos logísticos más significativos es la programación eficiente de los días de rodaje. Este desafío no solo implica consideraciones artísticas y técnicas, sino también la gestión óptima de recursos humanos y financieros. Uno de los costos más críticos y a menudo subestimados en este contexto es el asociado con los "días de espera" del talento, que incluye actores, extras y a veces personal técnico especializado.

2.1 Definición del Problema

Los días de espera son aquellos en los cuales el talento está presente en el set de rodaje pero no participa activamente en la filmación de escenas. Durante estos días, aunque no están trabajando, los actores y otros talentos deben ser compensados, generando un gasto considerable para la producción. El objetivo del artículo[3] es secuenciar los días de rodaje de manera que se minimicen estos días de espera, reduciendo así los costos innecesarios sin comprometer la eficacia del proceso de producción.

2.2 Desafíos Específicos

- **Coordinación de múltiples actores:** Cada actor puede tener diferentes días en los que es necesario que participe, dependiendo del guion y la logística de las locaciones.
- **Restricciones de disponibilidad:** A menudo los actores tienen compromisos previos o limitaciones contractuales que restringen su disponibilidad.
- **Complejidad combinatoria:** El número de combinaciones posibles de días de rodaje aumenta exponencialmente con el número de actores y días involucrados, lo que convierte la programación en un problema combinatorio complejo.

2.3 Importancia de la Optimización

Dada la naturaleza competitiva y costosa de la industria cinematográfica, la eficiencia en la programación puede traducirse en ahorros significativos y en una mayor capacidad para invertir en otros aspectos de la producción como efectos especiales, mejores locaciones o talentos de mayor renombre. Además, una planificación óptima mejora el ambiente de trabajo al reducir los tiempos de inactividad forzada, lo que potencialmente puede afectar el desempeño y la satisfacción del talento.

3 Modelación del Problema FS

3.1 Entradas del Problema

- **Matriz DODM (Day Out of Days Matrix):** Matriz binaria $T^0 \in \{0,1\}^{m \times n}$, donde:

$$t_{ij}^0 = \begin{cases} 1, & \text{si el actor } i \text{ está requerido el día } j, \\ 0, & \text{en otro caso.} \end{cases}$$

- **Vector de Costos:** $C \in \mathbb{R}^m$, donde c_i es el costo por día de espera del actor i.

3.2 Función Objetivo

Dada una permutación σ de los días de rodaje, el costo total de días de espera se calcula como:

$$K(\sigma) = \sum_{i=1}^{m} c_i \left[l_i(\sigma) - e_i(\sigma) + 1 - \sum_{j=1}^{n} t_{ij}^0 \right],$$

donde:

- $e_i(\sigma)$: Primer día en que el actor i está requerido en la secuencia σ.
- $l_i(\sigma)$: Último día en que el actor i está requerido en σ.
- $\sum_{j=1}^{n} t_{ij}^0$: Total de días que el actor i está requerido.

4 Demostracion NP-Hard del Problema de Programación de Filmación

En esta sección, demostraremos que el problema de programación de filmación (FS) es NP-Hard mediante una reducción desde el problema de **Optimal Linear Arrangement (OLA)**, el cual es NP-Hard[1]. El problema FS es **NP-Hard**, ya que incluso su versión restringida, donde cada actor es requerido exactamente en dos días y el vector de pago es unitario, es reducible polinomialmente a OLA. Esto implica que la existencia de un algoritmo pseudopolinomial es improbable [2].

Problema de Optimal Linear Arrangement (OLA)

Entrada: Un grafo no dirigido $G = (V, E)$ y un entero positivo B. **Pregunta:** ¿Existe una función biyectiva $\sigma : \{1, 2, \ldots, |V|\} \to \{1, 2, \ldots, |V|\}$ tal que

$$\sum_{\{v_i, v_j\} \in E} |\sigma(i) - \sigma(j)| \leq B?$$

Teorema

El problema FS es NP-Hard.

Demostración

Dada una instancia de OLA con $G = (V, E)$ y B, construimos una instancia del problema FS de la siguiente manera:

1. **Actores como aristas:** Cada arista $E_i \in E$ corresponde a una fila i en la matriz DODM T^0.
2. **Días como vértices:** Cada vértice $V_j \in V$ corresponde a una columna j de T^0, donde:
$$t_{ij}^0 = \begin{cases} 1, & \text{si } V_j \text{ es un extremo de } E_i, \\ 0, & \text{en otro caso.} \end{cases}$$
3. **Costos unitarios:** El vector de pago se define como $c_i = 1$ para todo i.

Afirmamos que la respuesta a OLA es "sí" si y solo si la instancia de FS tiene un costo de días de espera $K(\sigma) \leq B - |E|$.

$$\text{Observamos que } \sum_{j=1}^{|V|} t_{ij}^0 = 2 \quad \forall i.$$

Para una permutación σ, el costo de días de espera es:

$$\begin{aligned} K(\sigma) &= \sum_{i=1}^{|E|} 1 \times [l_i(\sigma) - e_i(\sigma) + 1 - 2] \\ &= \sum_{i=1}^{|E|} [l_i(\sigma) - e_i(\sigma)] - |E| \\ &= \sum_{\{v_i, v_j\} \in E} |\sigma(i) - \sigma(j)| - |E|. \end{aligned}$$

Si existe una permutación σ para OLA con $\sum |\sigma(i) - \sigma(j)| \leq B$, entonces $K(\sigma) \leq B - |E|$. La conversa también es cierta. Esto completa la reducción polinomial y demuestra que FS es NP-Hard. □

5 Algoritmos y Análisis

5.1 Algoritmo de Ramificación y Poda (Branch-and-Bound)

Descripción: El algoritmo de ramificación y poda comienza con la construcción de un árbol de búsqueda, donde cada nodo representa una secuencia parcial de los días de rodaje. La construcción de la secuencia se realiza desde los extremos hacia el centro, determinando primero los días extremos y luego los intermedios. Se calcula una cota inferior para cada nodo, representando el costo mínimo de días de espera si la secuencia de ese nodo se completara hasta formar una secuencia total.

Construcción de cotas: Las cotas inferiores se calculan usando la información de los días ya programados en la secuencia parcial. Se considera el número mínimo de días de espera que un actor debe incurrir, dado que algunos días de rodaje ya están fijados en la secuencia y su presencia es requerida en días aún no programados.

- **Cota inferior:** Para un nodo P, se calcula como:

$$K_{\text{lower}}(P) = \sum_{i=1}^{m} c_i \left[D_i^e(P) + D_i^l(P) \right],$$

donde $D_i^e(P)$ y $D_i^l(P)$ son días de espera obligatorios para el actor i en la secuencia parcial P.

Complejidad y Correctitud:

- **Tiempo:** La complejidad es $O(n!)$, reflejando que en el peor caso, se deben explorar todas las permutaciones posibles de los días de rodaje.
- **Correctitud:** La poda elimina del árbol de búsqueda aquellas secuencias que definitivamente no pueden superar al mejor costo encontrado hasta el momento, asegurando que no se descarten soluciones óptimas.

5.2 Método Heurístico

Descripción: Este método heurístico incluye dos pasos principales:

1. **Paso 1 - Ruta Más Atractiva:** Construcción de una secuencia inicial seleccionando los días que minimicen el incremento inmediato del costo total de días de espera, empezando desde los extremos hacia el centro.
2. **Paso 2 - Intercambios de Pares:** Mejora de la secuencia inicial mediante intercambios de pares de días para reducir el costo total. El proceso se repite hasta alcanzar un óptimo local.

Complejidad y Correctitud:

- **Tiempo:** $O(n^3)$ por iteración, debido a los intercambios de pares ($O(n^2)$) y los cálculos de costo por intercambio ($O(n)$).
- **Correctitud:** No garantiza la solución óptima global pero es eficaz para obtener soluciones cercanas al óptimo local con un error promedio menor al 10%.

Paper 2

6 Descripción del Problema

El artículo[4]. estudia el problema de **programación de talentos para rodajes cinematográficos**, extendiéndolo al considerar los costos de alquiler de locaciones. Se busca determinar la secuencia óptima de filmación de escenas minimizando los costos totales asociados a los actores y locaciones. A diferencia de trabajos previos, este modelo incorpora variabilidad en los costos de alquiler de locaciones a lo largo del horizonte de planificación.

Se considera un conjunto de escenas $S = \{s_1, s_2, ..., s_n\}$ y un conjunto de actores $A = \{a_1, a_2, ..., a_m\}$. Cada escena s_j requiere un subconjunto de actores $A_j \subseteq A$ y tiene una duración d_j. Los actores reciben un pago diario desde su primera aparición en el rodaje hasta la última, independientemente

de si participan en todas las escenas de ese período. Además, cada locación tiene un costo de alquiler variable a lo largo del tiempo.

El objetivo es determinar una secuencia de filmación y las fechas de inicio de cada escena para minimizar el costo total, considerando:

- Costos de los actores basados en su período de contratación.
- Costos de alquiler de locaciones, que varían a lo largo del horizonte de planificación.
- Restricciones de precedencia entre escenas.
- Cumplimiento del horizonte de planificación.

7 Formulación Matemática

El problema se modela como un **problema de programación entera mixta (MILP)** con las siguientes variables de decisión:

- x_{ij}: variable binaria que indica si la escena s_j se filma después de s_i.
- y_{ij}: variable binaria que indica si la escena s_i comienza en el día j.
- t_j: día de inicio de la escena s_j.
- e_i, l_i: primer y último día en que el actor a_i es requerido.

El modelo optimiza la función objetivo:

$$\min \sum_{i=1}^{m} c_i(l_i - e_i + 1) + \sum_{i=1}^{n} \sum_{j=1}^{p} y_{ij} L_{ij}, \tag{1}$$

donde c_i representa el pago diario del actor a_i y L_{ij} el costo de alquiler de locación si la escena s_i comienza el día j.

Las restricciones aseguran la secuenciación correcta de las escenas, la asignación de fechas dentro del horizonte de planificación y el cálculo correcto del tiempo de contratación de los actores.

8 Método de Solución

Dado que el problema es NP-hard, se propone una **búsqueda local iterada (ILS)** para resolver instancias grandes de manera eficiente. El algoritmo consta de:

- **Construcción de solución inicial:** inserción secuencial de escenas minimizando costos de actores.
- **Búsqueda local:** intercambio y reubicación de escenas para mejorar la solución.
- **Perturbación:** modificaciones aleatorias para escapar de óptimos locales.
- **Criterios de aceptación:** balance entre exploración y explotación mediante umbrales dinámicos.

9 Resultados Experimentales

Se generaron 30 instancias aleatorias con diferentes tamaños y se compararon las soluciones obtenidas por el modelo MILP y el algoritmo heurístico ILS. Los resultados muestran que:

- MILP encuentra soluciones óptimas solo en instancias pequeñas (hasta 10 escenas).
- ILS obtiene soluciones de alta calidad en tiempos significativamente menores.
- La heurística supera al MILP en muchas instancias grandes en términos de calidad y estabilidad.

10 ILS

El artículo describe el método Iterated Local Search (ILS) con los siguientes componentes clave:

10.1 Construcción de la Solución Inicial

- Se utiliza una estrategia de inserción secuencial simple para construir el horario inicial.
- Intenta colocar cada escena en un orden que minimice los costos de los actores.

10.2 Heurística de Búsqueda Local

- Utiliza dos estructuras de vecindario:
  - Relocate: Mueve una escena de su posición actual a otra posición.
  - Swap: Intercambia las posiciones de dos escenas.
- Evalúa los movimientos utilizando una función objetivo aumentada que aproxima el costo total.

10.3 Estrategia de Perturbación

- Aplica modificaciones aleatorias al horario para escapar de mínimos locales.
- Se seleccionan aleatoriamente dos estrategias:
  - Intercambio aleatorio: Intercambia las posiciones de dos escenas.
  - 2-opt aleatorio: Invierte una subcadena aleatoria de la secuencia de escenas.

10.4 Criterios de Aceptación

- Utiliza una regla de aceptación por umbral:
  - Se acepta una nueva solución si su costo es como máximo $(1 + \Delta)$ veces el mejor costo actual.
  - El umbral Δ disminuye gradualmente con el tiempo.

10.5 Condición de Terminación

- El algoritmo se detiene cuando se alcanza un número fijo de iteraciones.
- La heurística de inserción inicial mejora la calidad de la solución inicial, haciendo que la convergencia sea más rápida.
- Los movimientos de intercambio y reubicación permiten una mejor exploración del espacio de búsqueda.
- El paso de perturbación ayuda a escapar de mínimos locales realizando modificaciones controladas.
- La estrategia de aceptación por umbral permite soluciones temporales peores, mejorando la optimización global.
- Fijar el número de iteraciones proporciona un tiempo de ejecución predecible, incluso si no siempre encuentra la mejor solución.

11 Programación Lineal Entera Mixta (MILP)

11.1 Lo que describe el artículo:

- El modelo MILP tiene como objetivo minimizar el costo total asociado con los salarios de los actores y los costos de alquiler de locaciones.
- Variables de decisión:
  - x_{ij}: Variable binaria que indica si la escena j está programada después de la escena i.
  - y_{ij}: Variable binaria que indica si la escena i comienza en el día j.
  - t_i: Variable entera que representa la fecha de inicio de cada escena.
  - e_i, l_i: Variables enteras que representan el primer y último día que se necesita a un actor.
- Función Objetivo:
  - Minimizar la suma de los costos de los actores $\sum c_i(l_i - e_i + 1)$ y los costos de alquiler de locaciones.
- Restricciones:
  - Asegura una secuencia de rodaje válida.
  - Asegura que cada escena comience exactamente un día.
  - Garantiza que los actores estén disponibles cuando se les necesite.
  - Respeta el horizonte de planificación.

Demostración de NP-Hard de TSP-LC: OLA ¡= TSP-LC:

Para demostrar que es NP-Hard nos basaremos en el problema OLA al igual que hicimos para demostrar que FS era NP-Hard. Dado que recibo un grafo no dirigido G = (V, E) y un entero positivo B como entrada de OLA, voy a construir la matriz que recibe TSP-LC (las filas son los actores y las columnas las escenas en las que trabajan) tal que por cada arista que pertenzca a E voy a tener una fila en la matriz y por cada vértice de G voy a tener una columna. En una celda de esta matriz ponemos una X solamente si el vértice que corresponde a la columns es uno de los extremos de la arista de la fila. Declaro el costo de renta de cada localización como 0 para cualquier fecha y le asigno a cada escena, representada por las columnas de la matriz, una duración de exactamente un día. Para todos los actores los salarios correspondientes(ci) se fijarán en 1. De esta manera la función objetivo solo necesita minimizar:

$$\sum_{i=1}^{m} c_i \left[l_i(\sigma) - e_i(\sigma) + 1 \right],$$

Por tanto, la función objetivo quedaría:

$$\begin{aligned} \sum_{i=1}^{|E|} 1 \times [l_i(\sigma) - e_i(\sigma) + 1] &= \sum_{i=1}^{|E|} [l_i(\sigma) - e_i(\sigma)] + |E| \\ &= \sum_{\{v_i, v_j\} \in E} |\sigma(i) - \sigma(j)| + |E|. \end{aligned}$$

Si existe una permutación σ para OLA con $\sum |\sigma(i) - \sigma(j)| \leq B$, entonces se debe cumplir que el resultado de la función objetivo sea $\leq B + |E|$, por ser el mínimo. La conversa también es cierta. Esto completa la reducción polinomial y demuestra que el problema es NP-Hard. □

12 Análisis de Complejidad Temporal y Correctitud

12.1 Complejidad Temporal

La complejidad de los métodos propuestos se desglosa así:

Componente	**Complejidad Teórica**
Modelo MILP	$O\big(2^{n \cdot p} \cdot m\big)$ (exponencial en el peor caso)
Iterated Local Search (ILS)	$O\big(T \cdot (I + P) \cdot n^3\big)$, donde T = número de iteraciones, I = tamaño del vecindario

Table 1: Complejidad teórica de los enfoques presentados en el paper

Detalles:

- **MILP:** En el artículo se utiliza un *solver* comercial (CPLEX) para resolver la formulación MILP. Debido a la naturaleza NP-hard del problema, el tiempo de ejecución puede crecer exponencialmente con respecto a n, p y m. Para instancias pequeñas, el solver encuentra soluciones óptimas en un plazo razonable; sin embargo, a medida que n y p aumentan, la resolución puede tardar mucho más tiempo.

- **ILS:** La complejidad viene dada por el número de iteraciones T y las operaciones en cada búsqueda local y etapa de perturbación. El paper demuestra empíricamente que esta aproximación heurística obtiene buenas soluciones en menor tiempo que el método exacto al crecer el tamaño del problema.

12.2 Correctitud

La correctitud de los métodos se fundamenta en:

1. **Cumplimiento de restricciones**: El modelo MILP del paper incluye todas las ecuaciones necesarias para representar:
   - El orden de las escenas y su encadenamiento (restricciones tipo *precedencia*).
   - La planificación dentro del horizonte de tiempo, respetando los límites de cada escena.
   - Los costos de actores y locaciones, garantizando que se sumen correctamente en la función objetivo.

   Al resolver dichas restricciones con un solver que sigue métodos de *branch-and-bound* y técnicas de corte, se asegura la factibilidad de la solución hallada.

2. **Optimalidad de la solución**:
   - **MILP:** El paper muestra que, para instancias pequeñas o medianas, el método exacto con CPLEX converge a soluciones óptimas, confirmadas por la verificación de *gaps* muy reducidos (o nulos) respecto al *bound* inferior calculado.
   - **ILS:** Aunque el *Iterated Local Search* no ofrece garantía de optimalidad, el paper señala que, en la práctica, se acerca a soluciones de alta calidad en menos tiempo para instancias de mayor tamaño. Se justifica que el ILS mantiene la corrección en el sentido de generar siempre soluciones factibles (cumplen las mismas restricciones que el MILP), pero no garantiza el óptimo global.

13 Resultados Experimentales

En esta sección se presentan los resultados obtenidos al ejecutar las implementaciones inspiradas en los algoritmos descritos anteriormente. Con el fin de reproducir, hasta donde es posible, la metodología del paper, se han desarrollado dos códigos principales:

1. Un método *exacto* basado en un modelo MILP simplificado que asigna días a cada escena suponiendo un orden predeterminado.

2. Una *heurística ILS (Iterated Local Search)* que, de forma análoga a lo expuesto en el paper, introduce una solución inicial, luego aplica movimientos de vecindad y perturbaciones, y finalmente se repite un cierto número de iteraciones.

Si bien no se cuenta con la misma formulación MILP completa ni con CPLEX, se han respetado las ideas centrales del documento, incluyendo la manera de generar instancias aleatorias con el número de escenas, actores y horizonte de planificación en rangos similares. De esta manera, los

resultados permiten observar el comportamiento de un enfoque exacto simulado en comparación con la búsqueda local iterativa, siguiendo la esencia de la evaluación propuesta en el paper.

Los algoritmos en el artículo fueron implementados en C++ compilado con GCC 9.3.0, utilizando CPLEX 12.10 para resolver el MILP, ejecutando las pruebas en un ordenador de escritorio con sistema operativo Kubuntu 20.04, equipado con un procesador AMD Ryzen 3700X a 4.0GHz y 16GB de RAM. Debido a la diferencia de recursos computacionales se decidió proceder con tres instancias para demostrar la aplicabilidad de los algoritmos.

A continuación, se exhiben las tablas que recogen los valores obtenidos para distintas instancias generadas. La primera tabla 2 recopila los datos correspondientes al método "MILP" simulado, mientras que la segunda tabla 3 expone los valores que caracterizan el desempeño de la heurística ILS en cada instancia.

Table 2: Resultados del método MILP

id	**n**	**m**	**p**	**Z(MILP)**	**gap(MILP)(%)**	**time(MILP)(s)**
0	25	14	189	3.16387e+06	0	0.05
1	9	18	46	1.02375e+06	0	0
2	10	18	47	4.72324e+05	0	0

Table 3: Resultados del método ILS (DLS)

id	**Zbest**	**gap'(%)**	**gap(%)**	**time(ILS)(s)**
0	3.03644e+06	-4.03	0.3	138.22
1	9.88486e+05	-3.44	0	18.91
2	4.23896e+05	-10.25	0	14.42

- **id:** Identificador de la instancia. Cada instancia se numera de manera secuencial.
- **n:** Número de escenas que componen la instancia.
- **m:** Número de actores requeridos en total.
- **p:** Horizonte de planificación (en días) disponible para filmar todas las escenas.
- **Z(MILP):** Costo total hallado por el "método MILP" simulado (en este código, se considera el orden natural de escenas y se resuelven únicamente los días de inicio).
- **gap(MILP)(%):** Brecha del método MILP frente a un posible óptimo. En este ejemplo, aparece en 0% por simplificación.
- **time(MILP)(s):** Tiempo, en segundos, que tardó la parte "MILP" en ejecutarse (muy reducido al ser un subproblema simple).
- **Zbest:** Mejor costo encontrado por el algoritmo de ILS en sus 10 corridas.

- **gap'(%):** Diferencia porcentual entre *Zbest* del ILS y *Z(MILP)*. Un valor negativo indica que ILS obtuvo una solución de menor costo que el MILP simulado.
- **gap(%):** Brecha promedio de las 10 corridas de ILS con respecto a la mejor. Si es 0, todas las corridas encontraron el mismo valor óptimo local.
- **time(ILS)(s):** Tiempo promedio de ejecución (en segundos) del ILS en cada una de sus corridas.

Las tres filas representan distintas instancias, cada una con diferentes dimensiones (*n, m, p*), salarios y costos de locación aleatorios. Se observa, por ejemplo, que en la primera instancia (id = 0), el método MILP simple reporta un costo de 3.16×10^6, mientras que el ILS logra 3.03×10^6, mostrando un gap'(%) de -4.03%. El *time(ILS)(s)* de 138.22 indica el promedio de segundos que tomó resolver dicha instancia a lo largo de las 10 corridas. En la segunda y tercera instancias (id = 1 y 2), se aprecian situaciones análogas, con *gap*(%) = 0 en las corridas de ILS, lo que significa que todas las ejecuciones encontraron la misma solución óptima local.

14 Soluciones que toman en cuenta la distancia entre los lugares de filmación $(\mathbf{n}^3)y(n^2)$

Referencias

References

[1] Garey, M. R., Johnson, D. S., & Stockmeyer, L. (1976). Some Simplified NP-Complete Graph Problems. *Theoretical Computer Science*, 1(3), 237–267.

[2] Garey, M. R., & Johnson, D. S. (1979). *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman.

[3] T. Cheng, J. Diamond, y B. M. Lin, "Optimal scheduling in film production to minimize talent hold cost," *Journal of Optimization Theory and Applications*, vol. 79, no. 3, pp. 479–492, 1993.

[4] 1st Thu Trang Hoa y 2nd Minh Anh Nguyen, "An Iterated Local Search for the Talent Scheduling Problem with Location Costs," *Phenikaa University*, Hanoi, Vietnam, 2024.