Solving The Rehearsal Scheduling Problem

Camilo Valenzuela, Martín Villanueva, German Ortiz, Gonzalo Moya
Universidad Técnica Federico Santa María
Avenida España 1680, Valparaíso, Chile
cvalenzu@alumnos.inf.utfsm.cl mavillan@alumnos.inf.utfsm.cl gortiz@alumnos.inf.utfsm.cl
gemoya@alumnos.inf.utfsm.cl

EVALUACIÓN

Resumen:	___/5
Palabras Claves:	___/5
Introducción:	___/5
Estado del Arte:	___/15
Modelo Matemático:	___/35
Experimentación:	___/15
Análisis de Resultados:	___/10
Conclusiones y Trabajo Futuro:	___/10
Referencias:	___/5
Nota Final:	___/105

***Resumen*—Se resolverá "The Rehearsal Problem", problema que trata sobre la asignación de músicos de una orquesta, a distinta piezas de ensayos, cada pieza tiene una duración distinta, donde los músicos que no tocan deben esperar que los otros músicos finalicen de tocar la pieza. Se busca minimizar el tiempo de espera entre cada pieza. (5 ptos.)**

***Palabras Claves*—Palabras claves del proyecto, se puede considerar como si se usara un buscador, debería ser encontrado al usar esas palabras. Deber ser específico respecto al problema y maneras de resolver. (5 ptos.) Rehearsal scheduling problem ?**

I. INTRODUCCIÓN

"The Rehearsal Problem" trata la planificación de los ensayos para un concierto, el cual consiste de nueve piezas de música de diferentes duraciones, donde cada una involucra una combinación distinta de músicos. Estos músicos pueden llegar pueden llegar a los ensayos inmediatamente antes de que les toque ensayar, e irse inmediatamente después de participar en su última pieza. El fin del problema es diseñar un orden específico de las piezas musicales con el fin de minimizar el tiempo de espera de los músicos para tocar en su ensayo, en otras palabras el tiempo en el cual los ensayistas estan presentes pero no estan tocando.

II. ESTADO DEL ARTE

El problema se inició en la Universidad de Lancaster, Reino Unido, en los años 70. Se sabe que el problema fue formulado por un miembro del personal del departamento de ciencias de la administración, que a la vez era miembro de una orquesta de aficionados, mientras esperaba su turno para tocar durante un ensayo.

El problema se plantea de la siguiente forma. Un concierto consta de nueve piezas de música, de diferente duraciones cada una, con una combinación diferente de los cinco miembros de la orquesta.

Las personas llegan a los ensayos inmediatamente antes de la primera pieza en la que tocarán y se van inmediatamente después de la última pieza pieza que tocan. El problema consiste en organizar el orden que minimice la espera de las personas que van a tocar; es decir, la suma del tiempo total que esperan las personas, para poder tocar, debe ser mínimo.

En el año 1993, Cheng, Diamond y Ling[1] plantearon un problema parecido. El problema ocurre cuando se está grabando una película. Diferentes días de grabación requieren diferentes elencos, y a los elencos se les debe pagar mientras estén presentes (ya sea esperando o actuando). Por lo tanto, el objetivo en este problema, es minimizar el tiempo de espera del elenco, para que así, se les pague lo menos posible por 'tiempo no producido'.

El primer enfoque que se conoce para resolver éste problema, era un algoritmo de Branch and Bound, que es NP-duro, entonces, para reducir el tiempo de cálculo de Branch and Bound, (Cheng et al., 1993) se desarrolló un método heurístico[1]. Nordström y Tufekci (1994)[2], propusieron un algoritmo que combina una heurística llamada 'pairwise interchange' con un simple algoritmo de genética, para resolver el problema propuesto por Cheng (1993)[1], éste algoritmo superó al propuesto por Cheng et al. (1993)[1], en términos de calidad de las soluciones y en el tiempo computacional.

García de la Banda, Stucky y Chu (2011)[3], aplicaron programación dinámica al problema de producción de cine. Kochetov (2011)[4] desarrolló tres nuevos algoritmos heurísticos, estos son, un 'annealing algorithm', 'un stochastic tabu search algorithm'y un algoritmo de búsqueda local genética. Estas heurísticas propuestas, pudieron encontrar la solución a los pocos minutos. Bomsdorf y Derigs (2008)[5], estudiaron un problema similar al del rodaje de la película, ellos tomaron en cuenta otros factores un poco más realistas, como tiempo de trabajo, escena anteriormente grabada, disponibilidad de recursos, disminuir el tiempo de finalización ('makespan') y minimizar los costos de los recursos. Se aplicó una técnica de búsqueda codiciosa indirecta ('greedy indirect search technique') para resolver el problema. Éste método, proporcionó horarios más rápidos y mejor que el método de programación manual.

Smith (2003)[6], tomó el problema de la programación de producción de cine y lo adaptó al problema de programación de la orquesta (problema que estamos estudiando). Éste problema fue resuelto como un problema de satisfacción de restric-

ciones. Gregory, Miller y Prosser (2004)[7], utilizaron técnicas de planificación y verificación de modelos para resolver el problema. Se compararon las dos técnicas, con la propuesta por Smith (2003)[6], el resultado mostró que la técnica de planificación superó la técnica de satisfacción de restricciones y la verificación de modelos en términos de tiempo de cálculo.

Hasta entonces, los 'Rehearsal Problem' anteriores se basaban en el caso donde las piezas musicales se encuentran secuenciadas desde la primera hasta la última. Sakulson and Tharmmaphornphilas (2011)[8], considerando piezas musicales de igual duración de ensayo, a través de una casilla. Ellos optaron por disponer de varios días para los ensayos, y en base a esto poder minimizar el número de días en que los músicos deben estar presentes. Además con la intención de minimizar el tiempo de espera de cada músico en cada día en particular.

Para nuestro caso, distinto del anterior, se considerarán piezas musicales de distinta duración, con limitación tanto en los slots de ensayo como en los días de ensayo. También se asume que la paga de los músicos es sólo para el día en que asisten a ensayar. Distinto del modelo presentado por Cheng et al. (1993)[1], quien considera pagos de músicos distintos, ahora se presentará el problema con montos de pago iguales para los músicos, es decir que la paga por día es igual para cada músico. Lo anterior permite reducir el problema a minimizar los días que deben presentarse los músicos a ensaya, buscando además la satisfacción de los músicos, reduciendo el tiempo de espera en el día de cada uno, para tocar la pieza que deben.

III. Modelo Matemático o LP

III-A. Formulación Estándar

A continuación se presenta una solución general al *Rehearsal Scheduling Problem* ocupando programación lineal entera. Este enfoque consiste en asignar las m piezas que conforman el ensayo (en donde participan p músicos) en una secuencia de n *slots* (casilleros) que minimicen el tiempo de espera. Se consideran los indices (i, j, k) de modo tal que

Indices:

$i \in \{1, \ldots, m\} = \mathcal{I}$ indices de ensayos
$j \in \{1, \ldots, n\} = \mathcal{J}$ indices de slots
$k \in \{1, \ldots, p\} = \mathcal{K}$ indices de músicos

en donde $n \geq m$, pues se considera un slot por cada unidad de tiempo, esto es, deben haber tantos slots como unidades de tiempo total tome practicar todos los ensayos. De este modo un ensayo que tome q unidades de tiempo, deberá ser asignado a una secuencia consecutiva de q slots.

Los parámetros del problema son los siguientes:

Parámetros:

$piezas_{ki} = 1$ ssi músico k toca en el ensayo i, 0 eoc.
δ_i duración del ensayo i

de forma natural se definen las variables de decisión del problema:

Variables:

$$s_{ij} = \begin{cases} 1 \text{ ssi ensayo } i \text{ va en el slot } j \\ 0 \text{ eoc.} \end{cases}$$

$$plays_{kj} = \begin{cases} 1 \text{ ssi músico } k \text{ debe tocar en slot } j \\ 0 \text{ eoc.} \end{cases}$$

$$start_{ij} = \begin{cases} 1 \text{ ssi ensayo } i \text{ es ensayado en o antes de slot } j \\ 0 \text{ eoc.} \end{cases}$$

$$end_{ij} = \begin{cases} 1 \text{ ssi ensayo } i \text{ es ensayado en o después de slot } j \\ 0 \text{ eoc.} \end{cases}$$

$$r_{kj} = \begin{cases} 1 \text{ ssi músico } k \text{ está presente en slot } j \\ 0 \text{ eoc.} \end{cases}$$

$$a_{kj} = \begin{cases} 1 \text{ ssi músico } k \text{ arriva en o antes de slot } j \\ 0 \text{ eoc.} \end{cases}$$

$$l_{kj} = \begin{cases} 1 \text{ ssi músico } k \text{ se retira en o después de slot } j \\ 0 \text{ eoc.} \end{cases}$$

$$w_{kj} = \begin{cases} 1 \text{ ssi músico } k \text{ debe esperar en slot } j \\ 0 \text{ eoc.} \end{cases}$$

como lo que se quiere es minimizar el tiempo de espera total la función objetivo (1) es

$$\text{Min } z = \sum_{k=1}^{p} \sum_{j=1}^{n} w_{kj} \tag{1}$$

notar que se suma sobre slots pues cada uno representa una unidad de tiempo.

A continuación se detallan las restricciones del modelo:

- **Asignación a slots.** La restricción (2) permite determinar los slots donde debe tocar cada músico

$$p_{kj} = \sum_{i=1}^{m} s_{ij} \pi_{ki} \quad \forall k \in \mathcal{K}, j \in \mathcal{I} \tag{2}$$

  básicamente s_{ij} permite sabe si el ensayo i va en el slot j, de ser así y si además el músico toca en el ensayo i, entonces significa que debe tocar en el slot j.
- **Arrivo de músicos**. Para determinar si el músico k llega (ha llegado) el slot j se tienen las siguientes dos restricciones

$$a_{kj} \leq a_{k,j+1} \quad \forall k \in \mathcal{K}, j \in \{1, \ldots, n-1\} \tag{3}$$

$$a_{kj} \geq p_{kj} \quad \forall k \in \mathcal{K}, \forall j \in \mathcal{J} \tag{4}$$

  Primero, (3) indica recursivamente que el músico k ya ha llegado en el slot $j+1$, si ha llegado en slot el slot j o alguno anterior. Como complemento a dicha recursión (4) impone que si el músico toca en el slot j, entonces ha llegado.
- **Salida de músicos**. Para determinar si el músico k se va al final (o después) del slot j se tienen las siguientes dos restricciones

$$l_{kj} \geq l_{k,j+1} \quad \forall k \in \mathcal{K},\ \forall j \in \{1,\ldots,n-1\} \tag{5}$$

$$l_{kj} \geq p_{kj} \quad \forall k \in \mathcal{K},\ \forall j \in \mathcal{J} \tag{6}$$

Muy similar al caso anterior, (5) el músico se va al final del slot j (o después), si se va en el slot $j+1$ o alguno posterior. La restricción (6) dice que el músico se puede ir al final del slot j (o después) si toca en dicho slot.

- **Presencia de músicos**. Un músico está presente en el slot j, si llega en el slot j (o antes) y se va al final (o después) del slot. Esto quede representado en (7)

$$r_{kj} \geq a_{kj} + l_{kj} - 1 \quad \forall k \in \mathcal{K},\ \forall j \in \mathcal{J} \tag{7}$$

- **Cuando un músico espera.** Un músico k espera en el slot j, si está presente pero no tiene que que tocar. Lo anterior se refleja en la restricción (8)

$$r_{kj} - w_{kj} \leq p_{kj} \quad \forall k \in \mathcal{K},\ \forall j \in \mathcal{J} \tag{8}$$

- **Asignación a slots.** Las siguientes restricciones permiten hacer la correcta asignación de ensayos a slots. Las restricciones (9) y (10) dicen respectivamente, que a cada slot sólo puede asignarse un ensayo, y que cada ensayo debe asignarse a tantos slots como duración tenga.

$$\sum_{i=1}^{m} s_{ij} = 1 \quad \forall j \in \mathcal{J} \tag{9}$$

$$\sum_{j=1}^{n} s_{ij} = \delta_i \quad \forall i \in \mathcal{I} \tag{10}$$

Las restricciones (11)-(15) fuerzan a que la asignación de un ensayo i sea en δ_i slots consecutivos, pues un ensayo individual no debe divirse en partes. La restricción (11) asegura que el ensayo i se realice en el slot j, si dicho ensayo comienza en el slot j (o antes) y termina al final del slot j (o después)

$$s_{ij} \geq start_{ij} + end_{ij} - 1 \quad \forall i \in \mathcal{I},\ \forall j \in \mathcal{J} \tag{11}$$

la restricción (12) indica que el ensayo i ha comenzado en el slot $j+1$, si ha comienza en j o antes y (13) sirve como base de la recursión anterior, estableciendo que si el ensayo i va en en el slot j, entonces dicho ensayo ya ha comenzado en dicho slot (o antes)

$$start_{ij} \leq start_{i,j+1} \quad \forall i \in \mathcal{I},\ \forall j \in \{1,\ldots,n-1\} \tag{12}$$

$$start_{ij} \geq s_{ij} \quad \forall i \in \mathcal{I},\ \forall j \in \mathcal{J} \tag{13}$$

la restricción (14) dice que el ensayo i termina en el slot j (o después), si termina en el slot $j+1$ (o después) y (15) sirve de base a la recursión, estableciendo que si el ensayo i va en el slot j, entonces dicho ensayo termina al final del slot j (o después)

$$end_{ij} \geq end_{i,j+1} \quad \forall i \in \mathcal{I},\ \forall j \in \{1,\ldots,n-1\} \tag{14}$$

$$end_{ij} \geq s_{ij} \quad \forall i \in \mathcal{I},\ \forall j \in \mathcal{J} \tag{15}$$

III-B. Extensión

Planteamiento de un modelo mejorado y/o extendido. Explicar el sentido que tiene la extensión y su explicación formal correspondiente. Justificar el nuevo modelo con respecto al estándar. **(15 puntos)**

IV. Experimentación

IV-A. Entorno (Hardware y Software)

Se debe detallar el *software* y *hardware* utilizado. Además, configuraciones y parámetros utilizados tanto en el modelo como en el *Solver* utilizado (En este caso el solver a utilizar será LINGO). **(5 puntos)**

IV-B. Resultados modelo estándar

Reportar resultados obtenidos, además de los tiempos de ejecución. **(5 puntos)**

IV-C. Resultados modelo extendido

Reportar resultados obtenidos, además de los tiempos de ejecución. **(5 puntos)**

V. Análisis de Resultados

Análisis de resultados del modelo estándar y del modelo extendido de manera independiente. Interpretar el comportamiento de los resultados. Luego realizar una comparación entre ambos modelos. **(10 puntos)**

VI. Conclusiones y trabajo futuro

Que se puede rescatar de todo lo anterior, sus resultados e inferencias. Preguntas claves ¿Qué se aprendió sobre la problemática?, ¿Qué se podría hacer a futuro?. **(10 ptos.)**

VII. Referencias

De donde obtuvo la información. Si fue sacada de una página web colocar el enlace directo a la información (es decir, google y wikipedia, este último al menos sin ninguna otra especificación, NO son referencias válidas y su mención será penalizada con 0 ptos. en este ítem), si se obtuvo de un paper usar el titulo, autores y año de publicación y si fue sacada de un libro usar el título, nombre del autor, edición y las páginas correspondientes. **(5 ptos.)**.

*En esta sección solo van las referencia, no se incluye ningún tipo de texto adicional. Y **muy importante**: deben aprender a referenciar libros, papers, Links de internet, etc. Es su trabajo averiguar el formato adecuado.*

REFERENCIAS

[1] T. C. E. Cheng, J. E. Diamond, and B. M. T. Lin. Optimal scheduling in film product to minimize talent hold cost. *Optimization Theory and Application*, page 3, 1993.

[2] Anna-Lena Nordström and Suleyman Tufekci. A genetic algorithm for the talent scheduling problem. *Computers and Operations Research*, 21(8):927 – 940, 1994. Heuristic, Genetic and Tabu Search.

[3] Maria Garcia De La Banda, Peter J Stuckey, and Geoffrey Chu. Solving talent scheduling with dynamic programming. *INFORMS Journal on Computing*, 23(1):120–137, 2011.

[4] Yury Kochetov. Iterative local search methods for the talent scheduling problem. In *Proceedings of 1st international symposium and 10th Balkan conference on operational research, September*, volume 22, pages 282–288, 2011.

[5] Felix Bomsdorf and Ulrich Derigs. A model, heuristic procedure and decision support system for solving the movie shoot scheduling problem. *Or Spectrum*, 30(4):751–772, 2008.

[6] Barbara M Smith. Constraint programming in practice: Scheduling a rehearsal. *Re-search Report APES-67-2003, APES group*, 2003.

[7] Peter Gregory, Alice Miller, and Patrick Prosser. Solving the rehearsal problem with planning and with model checking. In *Proceedings of the Workshop on modelling and solving problems with constraints. Held in conjunction with the 16th European Conference on Artificial Intelligence (ECAI 2004)*, pages 157–171, 2004.

[8] Noppadon Sakulsom and Wipawee Tharmmaphornphilas. A multi-objective music rehearsal scheduling problem. In *Proceedings of 12th Asia Pacific Industrial Engineering & Management Systems Conference (APIEMS 2011)*, 2011.