Solving The Rehearsal Scheduling Problem

Camilo Valenzuela, Martín Villanueva, Germán 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

***Resumen*—"Rehearsal Problem". El fin del problema es diseñar un orden específico de las piezas musicales para minimizar el tiempo de espera de los músicos que están presentes, pero no están tocando. El problema será abordado mediante programación lineal entera.**

***Palabras Claves*—Rehearsal scheduling problem, programación lineal entera, optimización.**

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 están presentes pero no están 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 restricciones. 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 encontraban secuenciadas desde la primera hasta la última. Sakulson and Tharmmaphornphilas (2011)[8], consideraron la organización del ensayo en varios días, para así, poder reducir el número de días que los músicos deben tocar. Además, en cada día, tratan de minimizar el tiempo de espera de cada músico

A diferencia del anterior, se considerarán piezas musicales de distinta duración, con limitación tanto en los slots de ensayo como en los días de ensayo. Por lo tanto, malas

asignaciones de los días de ensayo, pueden conducir a una solución no factible. Distinto del modelo presentado por Cheng et al. (1993)[1], quien considera costos por músicos, ahora se presentará el problema con costos iguales para los músicos, es decir que la paga por día es igual para cada músico. Así, para minimizar el tiempo de espera, la meta es reducir al mínimo la rotación entre los músicos.

Es importante señalar la contribución que realizaron Sakulson and Tharmmaphornphilas (2014)[**?**], la cual sirve de base para la resolución al problema que fue realizado en el presente documento.

III. Modelo Matemático o LP

III-A. Formulación Estándar

A continuación se presenta una solución general al *Rehearsal Scheduling Problem* ocupando programación lineal entera basado en el modelo planteado por Sakulson and Tharmmaphornphilas (2014)[**?**]. 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

Para la extensión del "The Rehearsal Problem", se observa que lo único que se toma en cuenta es el tiempo de espera de los músicos. Una versión más compleja del problema tratado, corresponde al "Talent Scheduling Problem"; En este se tiene que cada actor (que corresponde a un músico en el problema de la orquesta) tiene un costo asignado, pues a cada uno se le paga una cantidad distinta de dinero por unidad de tiempo, en donde generalmente mientras más experiencia más dinero debería ganar. Pero si se tiene a las personas de mayor costo esperando, el gasto va a ser necesariamente más alto. Por lo que el objetivo en este problema, es buscar la secuencia en que deben desarrollarse los ensayos, minimizando el gasto total de tener músicos experimentados (o que sean más costosos) esperando.

Este problema se extiende de manera natural desde el "Rehearsal Problem", agregando un costo asociado a cada músico. Por lo tanto es necesario agregar un nuevo parámetro al modelo, que es precisamente el costo por unidad de tiempo de tener al músico k presente en la orquesta:

c_k : Costo unitario de tener al músico k presente

El modelo es exactamente el mismo, salvo por la función objetivo (dada la similitud de los problemas). La nueva función objetivo buscará minimizar el costo de tener a los músicos esperando (no simplemente su tiempo de espera). Para esto, por cada slot en donde el músico k se encuentre presente pero sin tocar (esperando), la función objetivo suma una cantidad igual al costo del músico (c_k)

$$\text{Min } z = \sum_{k=1}^{p} \sum_{j=1}^{n} c_k \cdot w_{kj} \tag{16}$$

por lo que dependiendo de los costos c_k que se definen en los músicos, se esperaran resultados diferentes al del modelo estándar.

IV. Experimentación

IV-A. Entorno (Hardware y Software)

Para de la implementación del modelo se utilizaron los siguientes elementos de hardware y software:

- **Hardware:**
  - Procesador: AMD Phenom II x4 840 Processor 3.20 GHZ.
  - Memoria RAM: 4 GB.
  - HDD: 150 GB SATA2.
  - Vídeo: AMD RADEON HD 7850.
- **Software:**
  - Sistema Operativo: Microsoft Windows 7 x64.
  - Solver: LINGO 10.0

IV-B. Resultados modelo estándar

Los resultados de importancia para la experimentación, son el orden en que se deben ensayar las piezas, el valor de la función objetivo y el tiempo de ejecución del programa.

$s_{9,1}$	1,000000
$s_{9,2}$	1,000000
$s_{9,3}$	1,000000
$s_{9,4}$	1,000000
$s_{9,5}$	1,000000
$s_{9,6}$	1,000000
$s_{4,7}$	1,000000
$s_{4,8}$	1,000000
$s_{4,9}$	1,000000
$s_{6,10}$	1,000000
$s_{6,11}$	1,000000
$s_{5,12}$	1,000000
$s_{5,13}$	1,000000
$s_{5,14}$	1,000000
$s_{1,15}$	1,000000
$s_{1,16}$	1,000000
$s_{2,17}$	1,000000
$s_{2,18}$	1,000000
$s_{2,19}$	1,000000
$s_{2,20}$	1,000000
$s_{7,21}$	1,000000
$s_{7,22}$	1,000000
$s_{7,23}$	1,000000
$s_{7,24}$	1,000000
$s_{7,25}$	1,000000
$s_{8,26}$	1,000000
$s_{8,27}$	1,000000
$s_{8,28}$	1,000000
$s_{8,29}$	1,000000
$s_{8,30}$	1,000000
$s_{8,31}$	1,000000
$s_{8,32}$	1,000000
$s_{3,33}$	1,000000

Cuadro I
RESULTADOS MODELO ESTÁNDAR

$s_{9,1}$	1,000000
$s_{9,2}$	1,000000
$s_{9,3}$	1,000000
$s_{9,4}$	1,000000
$s_{9,5}$	1,000000
$s_{9,6}$	1,000000
$s_{4,7}$	1,000000
$s_{4,8}$	1,000000
$s_{4,9}$	1,000000
$s_{2,10}$	1,000000
$s_{2,11}$	1,000000
$s_{2,12}$	1,000000
$s_{2,13}$	1,000000
$s_{1,14}$	1,000000
$s_{1,15}$	1,000000
$s_{5,16}$	1,000000
$s_{5,17}$	1,000000
$s_{5,18}$	1,000000
$s_{6,19}$	1,000000
$s_{6,20}$	1,000000
$s_{8,21}$	1,000000
$s_{8,22}$	1,000000
$s_{8,23}$	1,000000
$s_{8,24}$	1,000000
$s_{8,25}$	1,000000
$s_{8,26}$	1,000000
$s_{8,27}$	1,000000
$s_{7,28}$	1,000000
$s_{7,29}$	1,000000
$s_{7,30}$	1,000000
$s_{7,31}$	1,000000
$s_{7,32}$	1,000000
$s_{3,33}$	1,000000

Cuadro II
RESULTADOS MODELO EXTENDIDO

Como input se utilizó la tabla que se muestra a continuación (instancia más conocida del problema), donde se muestran los ensayos en que deben tocar cada músico

Músico \Ensayo	1	2	3	4	5	6	7	8	9
1	1	1	0	1	0	1	1	0	1
2	1	1	0	1	1	1	0	1	0
3	1	1	0	0	0	0	1	1	0
4	1	0	0	0	1	1	0	0	1
5	0	0	1	0	1	1	1	1	0
Duración	2	4	1	3	3	2	5	7	6

En el Cuadro I se muestran los valores que no son nulos de las variables de decisión definidas al inicio, se considera que aportan al análisis.

El mejor resultado que se obtuvo para la función objetivo fue de **17**, y el tiempo de ejecución del programa fue de **1hr : 39min : 33seg**.

IV-C. Resultados modelo extendido

Para el caso del modelo extendido se utilizó la misma instancia del problema, resuelta con el modelo estándar, pero ahora se asignan los costos correspondientes a cada músico

Músico	Costo
1	1
2	5
3	2
4	2
5	3

El modelo extendido demoró casi el doble que el modelo estándar, con un tiempo de ejecución total de **3 hr : 10 min : 02 seg** y un costo mínimo de la función objetivo de **32**.

En el Cuadro II se muestran las mismas variables que en el caso estándar, pero con los resultados obtenidos del modelo extendido.

V. Análisis de Resultados

V-A. Modelo estándar

Se puede observar que el tiempo necesario para resolver el problema usando el modelo estándar es muy grande. Por otro lado, el resultado es bueno teniendo en cuenta que el problema sin optimizar (tal y como venían los ensayos) tiene un tiempo de espera de 49 [u. de tiempo] y el resultado optimizado es de 17 [u. de tiempo], casi un $65\,\%$ de mejora en los tiempos de espera. La solución que optimiza el problema (tiempo de espera total de los músicos) es poner los ensayos en el orden que se muestra a continuación

$$9 \to 4 \to 6 \to 5 \to 1 \to 7 \to 8 \to 3$$

V-B. Modelo extendido

Se observa que el tiempo de ejecución es muy grande, esto debido a la gran cantidad de variables que considera el modelo y que al agregar los costos, existe una mayor cantidad de combinaciones por probar. El resultado obtenido para la función objetivo es el óptimo, pero la mejora (respecto a tal y como venían los ensayos) no es tanto , ya que el valor inicial (sin realizar ningún orden) es 34 [u. de costo] y el óptimo es 32 [u. de costo].

El orden en que se deben realizar los ensayos para minimizar los costos por tiempos de espera, es el siguiente

$$9 \to 4 \to 2 \to 1 \to 5 \to 6 \to 8 \to 3$$

V-C. Análisis General

El tiempo de ejecución de los dos modelos se debe a la gran cantidad de variables que se usaron en el modelo, y al método que utiliza el software *LINGO* para resolver el problema, que en este caso es *Branch and Bound*, el cual depende mucho del orden en que se va a recorriendo el árbol de resultados.

En relación al orden de las piezas, se observan diferencias en cómo los distintos modelos organizan las piezas, pero esto se debe a que cada modelo apunta a cosas distintas. uno minimiza el tiempo de espera total y el segundo debe minimizar el costo a pagar por los músicos cuando estos están esperando.

VI. Conclusiones y trabajo futuro

El presente proyecto nos ayudo y enseño a realizar un paper formal basado en los estándares que se utilizan hoy en día, además, tuvimos la experiencia de aprender de otras publicaciones para utilizarlas en el desarrollo de nuestro proyecto. Esto se debe, a que es nuestra primera vez que desarrollamos un proyecto de esta índole.

Respecto a la problemática tratada en este documento, se entendió como utilizar el método de programación lineal entera en un problema de asignación, buscando reducir el tiempo de espera y la correcta distribución de los músicos en las distintas piezas para tocar; es decir, ya no se piensa solamente en quién se va a asignar, sino que también, en la distribución de las asignaciones.

Nos damos cuenta, que este tipo de problema es mapeable con y sin modificaciones a distintas situaciones donde se necesite algún tipo de asignación, por ejemplo, en otros ámbitos en donde se necesite realizar la asignación de personal o minimizar los tiempos de espera de quiénes deban realizar estas tareas. La idea de mapear el problema es que muchas veces es más fácil resolver otro problema de características similares con pequeñas variaciones y luego acomodar el procedimiento a la situación actual.

Pensamos que a futuro se pueden encontrar nuevos métodos más eficientes que los que existen hoy en día, y obtengan el resultado optimo en mucho menos tiempo y menos iteraciones, como son los problemas NP.

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.

[9] Noppadon Sakulsom and Wipawee Tharmmaphornphilas. Scheduling a music rehearsal problem with unequal music piece length. *Computers & Industrial Engineering*, 70:20–30, 2014.