

Universidad Nacional Autónoma de México
Facultad de Ciencias
Complejidad Computacional

Tarea

Problema de asignación óptima de salones resuelto con Búsqueda Tabú

Ángel Iván Gladín García
No. cuenta: 313112470
angelgladin@ciencias.unam.mx

4 de diciembre 2019

1. Descripción del problema

El artículo habla sobre el problema de la asignación de salones como un problema de optimización matemática.

Al inicio de cada periodo académico la administración conoce las asignaturas a las que los estudiantes van a ir, y también deben de distribuirlas en su planta física de tal forma que todos los interesados reciban las materias que pidieron. Pero el modelo tiene un conjunto de restricciones llamadas **duras** (su cumplimiento es obligatorio) y **blandas** (su cumplimiento es optativo). Si las restricciones anteriores son satisfechas se dice que la propuesta dada es **factible**. El tipo de restricción llamada blanda dice que su cumplimiento no es obligatorio pero miden el grado de satisfacción de los estudiante.

Este problema combinatorio es clasificado como del tipo no lineal entero mixto y es de difícil solución. Clasificado como NP completo

Se han descrito varios métodos para resolver este problema, tales como: métodos secuenciales, métodos de clusterización, métodos basados en restricciones, y los métodos metaheurísticos.

Se aborda este problema para la solución del problema de asignación óptima de salones usando la técnica de **Búsqueda Tabú**.

Se va a plantear algoritmos constructivos para generar una configuración inicial de calidad y también se va a definir la estructura de vecindad para la búsqueda local con *Búsqueda Tabú*.

2. Planteamiento del problema

La naturaleza de este tipo de problemas son de organizar una secuencia de eventos, en un periodo de tiempo determinado satisfaciendo un conjunto de restricciones. En este caso como es la asignación de salones las restricciones son; asignación de recursos, asignación de tiempo, restricciones de tiempo entre sesiones, dispersión de las sesiones, coherencia de las reuniones, capacidad de las salas, continuidad.

En este artículo como ya se mencionó antes, se busca el cumplimiento de restricciones duras y las restricciones blandas (que nos son obligatorias que se cumplan).

Las restricciones duras en este caso son; que el salón asignado cumpla con los requisitos apropiados para el evento que se hará, los horarios (que no haya conflictos), y que un salón no tenga dos eventos a la misma hora.

Si todas esas restricciones se cumplen se dice que la propuesta es **factible** pero como queremos que los estudiantes hagan sus actividades de forma cómoda, así que queremos que se cumpla el mayor número de restricciones blandas. Si una restricción blanda no se cumple será penalizada en **la función objetivo** incrementando su valor en 1 por cada restricción violada,

Analizando todo eso, se empieza a formalizar el problema de programación de horarios de clase, de tal forma que si se cumplen las restricciones duras, se **minimizan** las restricciones blandas.

3. Búsqueda Tabú

La *Búsqueda Tabú* es una técnica de optimización combinatorial que proviene de la inteligencia artificial y usa conceptos de memoria adaptativa y exploración sensible. Un algoritmo de Búsqueda Tabú completo utiliza técnicas de exploración y de memoria avanzadas, como son: memoria de corto y largo plazo, estrategias de intensificación, diversificación, oscilación estratégica, *path relinking* y lista de configuraciones de élite, entre otras.

3.1. Función objetivo

$$\min f(x) \quad x \in X$$

Donde f es una función y X es un conjunto de restricciones.

4. Modelado matemático y codificación

La información del problema se maneja por medio de vectores y matrices en los cuales se indica:

- Matriz de booleanos que indica si un evento (fila) requiere cierta característica (columna).
- Matriz de booleanos que indica si un salón (fila) requiere cierta característica.
- Un vector que indica la capacidad del salón i .
- Matriz de booleanos que indica si un estudiante está matriculado en un evento.
- Matriz de booleanos que indica los eventos de los salones, indicando si el evento i se puede programar en el evento j .

Lo que se busca minimizar es:

$$z = hf + hu + hc$$

Donde la variable hf representa el número de eventos programados al final de cada día para todos los estudiantes, hu el número de eventos únicos en cada día para todos los estudiantes, y hc representa los casos en que se programan más de dos eventos consecutivos en un día para todos los estudiantes.

5. Aplicación de Búsqueda Tabú a la programación óptima de horarios de clase

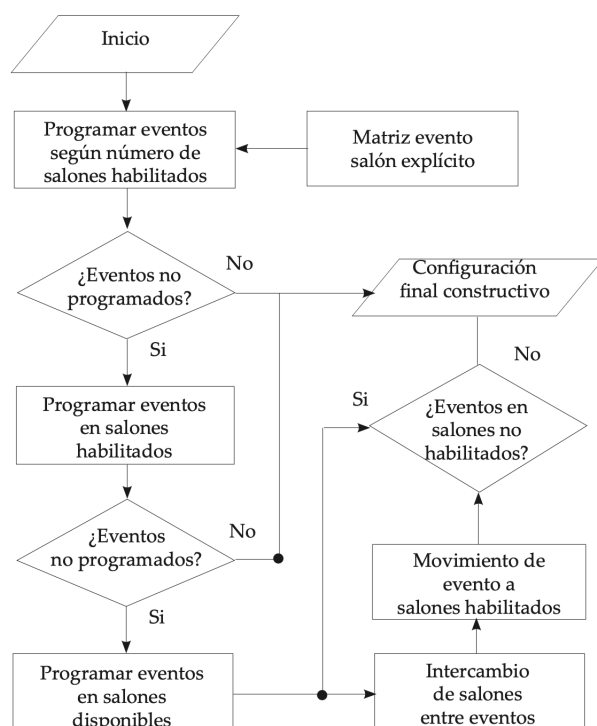
Tendrá cuatro fases para la solución de problema.

1. Tener una programación inicial para que los eventos sean programados con pocas restricciones duras violadas.
2. Búsqueda local para disminuir restricciones duras violadas.
3. Alcanzar una solución factible, en la que se cumplen las restricciones duras usando la **Búsqueda Tabú** para minimizar restricciones duras.
4. Usar **Búsqueda Tabú** para minimizar restricciones blandas, manteniendo la factibilidad de la configuración.

5.1. Generación de configuración inicial

Se hace de forma prioritaria teniendo los eventos que tiene menor número de salones habilitados. Esto se hace ordenando los eventos de menor a mayor número de salones habilitados. Al principio de programan los eventos que solo se pueden hacer en un salón en específico. Se van *emperajendo* así salones con los eventos de forma creciente.

Se puede ver en este diagrama la forma de hacerse:



5.2. Evaluación de la función objetivo

Lo que se hace es contar el número de restricciones duras y blandas que son violadas. Se revisa las restricciones duras de la configuración, checando si el salón en el que se hará el evento este habilitado, ósea que tiene las capacidades y características que se necesitan. Después de checa si hay crece en los

horarios para los estudiantes, se hace esto checando las combinaciones de de dos salones a la misma hora.

5.3. Índice de sensibilidad

Se crea un índice para cada evento, usando un contador que tiene el número de problemas asociados a este. Esto sirve para el proceso de búsqueda.

5.4. Estructura de vecindad

El vecindario de una configuración consiste en configuraciones obtenidas haciendo pequeños cambios a la configuración actual. Un pequeño cambio en la configuración puede ser la modificación de hora o de salón para un evento. Con el fin de alterar la hora de un evento, sin que se afecte su factibilidad por el salón, se puede intercambiar el bloque de tiempo de dicho evento con el de otro suceso en el mismo lugar.

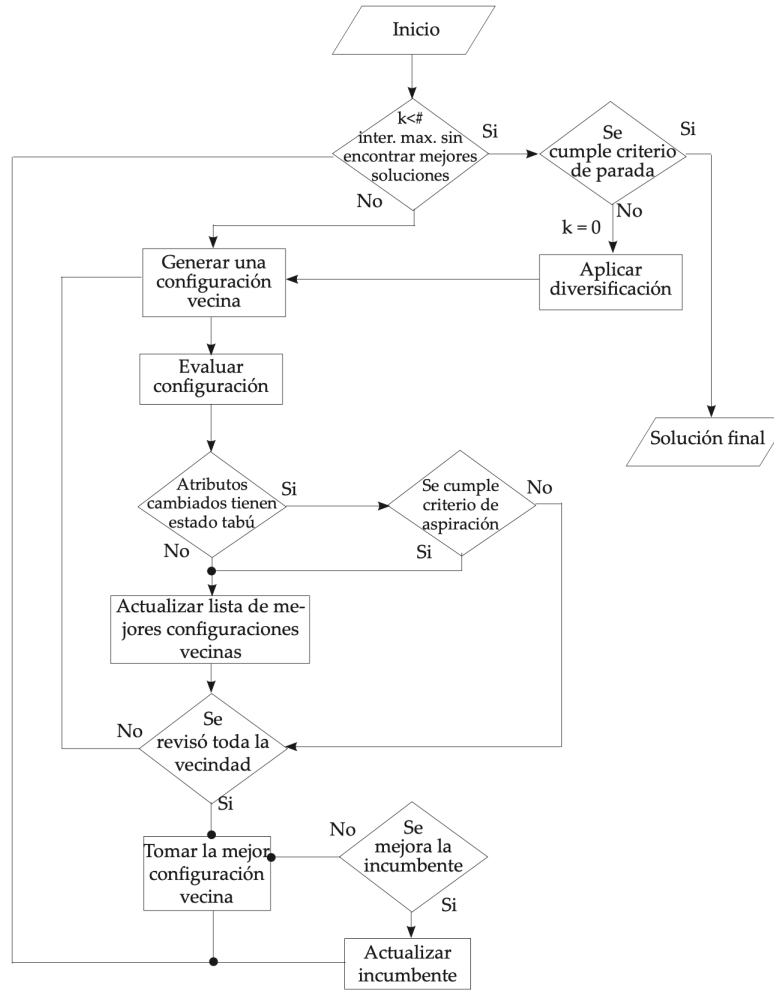
Se generan configuraciones vecinas a partir de un intercambio del bloque de tiempo del evento con otro suceso en el mismo salón y un movimiento del evento a un bloque de tiempo libre, en un salón habilitado para la actividad.

5.5. Manejo de la memoria de corto plazo

En Búsqueda Tabú se utiliza una estructura de memoria de corto plazo que evita regresar a configuraciones ya visitadas, con lo que se puede escapar de óptimos locales en el proceso de búsqueda.

5.6. Criterio de aspiración

« Si se da el caso en que la mejor configuración vecina se encuentre prohibida, y su función objetivo mejora la mejor función objetivo encontrada hasta el momento en la búsqueda (valor de la incumbente), el criterio de aspiración permite seleccionar esta configuración a pesar de estar excluida. »



6. Conclusiones

El problema es *complejo* por las variables relacionadas y condiciones involucradas.

Como la forma en la que se generan la configuración inicial es *eficiente* por lo que el número de iteraciones necesarias en la etapa de la Búsqueda Tabú para las restricciones duras es menor porque desde un principio se hacen movimientos que lo que buscan es evitar restricciones duras y que se reduzcan.

Los movimientos propuestos para generar vecindarios tienen la capacidad junto con las estrategias de Búsqueda Tabú de escapar de ótimos locales y minimizar ambas restricciones. Esto se hace cuando el proceso de búsqueda no mejora después de un número determinado de iteraciones.

Al final este artículo aborda este problema a su manera presentando la forma en como lo modelaron y lo resolvieron.

Referencias

- [1] Fredy, John & Franco, John & Toro, Eliana & Ocampo, Toro & Alfonso, Ramón & Rendón, Gallego. (2008). Problema de asignación óptima de salones resuelto con Búsqueda Tabú.