

Problema de asignación óptima de salones resuelto con Búsqueda Tabú

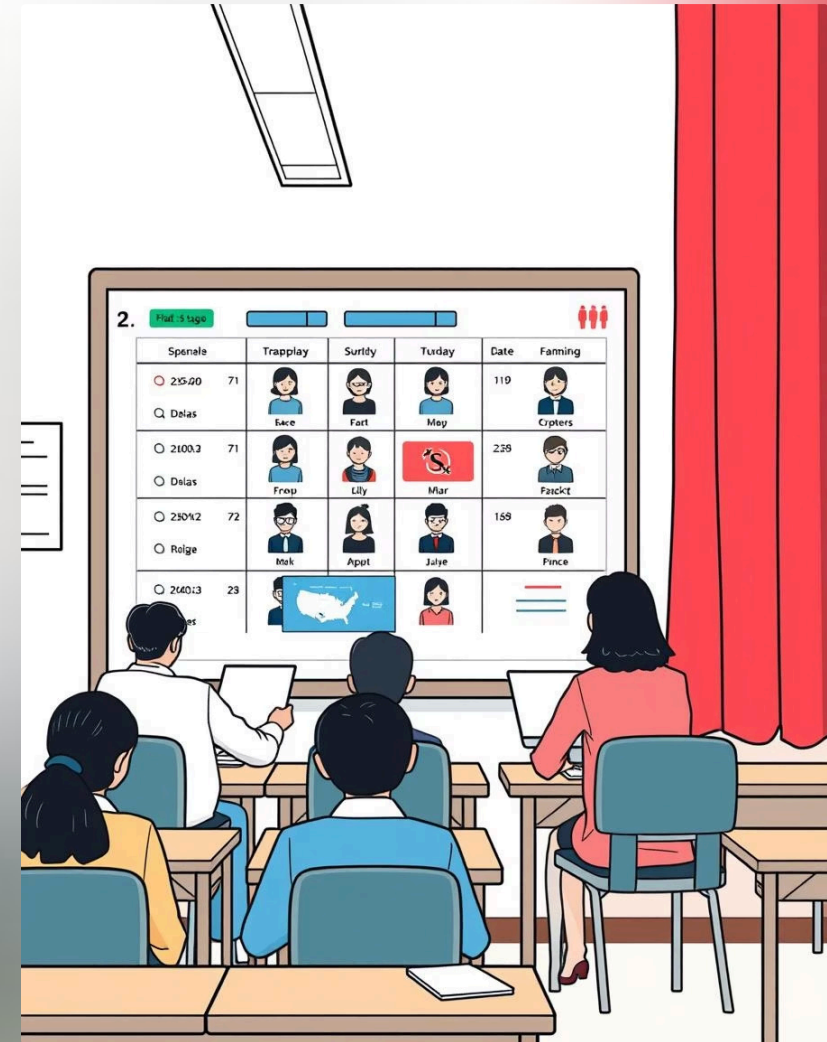
Ingeniería & Desarrollo. Universidad del Norte. 24: 159-185, 2008

John Fredy Franco Baquero, Eliana Mirledy Toro Ocampo, Ramón Alfonso Gallego Rendón

Explorando soluciones eficientes para un problema NP-completo típico.



Luciana Puentes, Julia Kroepfl y Adriano Fabris



Índice

01

El Problema de Asignación de Salones

Definición y complejidad.

02

Búsqueda Tabú: Una Metaheurística

Principios de la técnica.

03

Restricciones: Duras y Blandas

Criterios de factibilidad y calidad.
Requisitos funcionales vs no.

04

Fases de la Metodología

Estrategia.

05

Resultados y Comparativas

Evaluación del rendimiento y futuras direcciones.

Comparación de enfoques

Dada la alta complejidad combinatoria, los métodos exactos son prohibitivos. Aquí es donde las metaheurísticas demuestran su valor, ofreciendo soluciones de buena calidad en tiempos razonables.



Métodos Exactos

- Rigurosos pero computacionalmente costosos.
- Complejos de formular y ejecutar.
- Ejemplos: Graph Traversals



Metaheurísticas

- Más fáciles de implementar y adaptar.
- Hallan soluciones viables y de buena calidad.
- **No garantizan encontrar el óptimo global.**
- Ejemplos: Búsqueda Tabú, Simulated Annealing

El Desafío de la Asignación de Salones

La asignación de salones, un problema recurrente en instituciones educativas como escuelas y universidades, busca **distribuir eventos (clases, exámenes) en horarios y aulas idóneos**. Es un problema de optimización matemática que se clasifica como **NP-completo**.

El modelo empleado en este trabajo se basa en el utilizado por la comunidad científica del Reino Unido que fue desarrollado para un concurso denominado International Timetabling Competition con el cual se busca resolver el problema University TT.

Tabla 1
Comparación entre la asignación de horarios escolares y universitarios

Características	Escolar	Universitario
Programación	Pocas elecciones Mallas bien estructuradas	Muchas elecciones Mallas débilmente estructuradas
Disponibilidad profesor	Ajustado (Poseen gran carga)	Flexible (Posee carga liviana)
Salas	Pocas salas Mismo tamaño Centralizadas	Muchas salas Variedad de tamaños Descentralizadas
Carga estudiantes	Muy saturado Una sola jornada	Medianamente holgado Utiliza mañanas y tardes
Criterio de optimización	Satisfacción de restricciones	Minimización de restricciones Transgredidas

Búsqueda Tabú: Navegando el Espacio de Soluciones

"Una estrategia de optimización que incorpora **memoria adaptativa y exploración sensible** para guiar la búsqueda."

Esta técnica resuelve problemas del tipo (1):

$$\begin{array}{ll} \min & f(x) \\ \text{s.a.} & x \in X \end{array}$$

f: función general, lineal o no

X: conjunto de restricciones, lineales o no

x: pueden ser continua, discreta o mixta

La **exploración sensible** de Búsqueda Tabú se basa en la idea de que una *"mala decisión tomada mediante una estrategia produce más información que una buena selección hecha de forma aleatoria"*.

Restricciones

Restricciones Duras (3)


Son mandatorias y no pueden ser violadas. Su incumplimiento hace que el horario sea **inviabile**:

- Capacidad y equipamiento adecuados del aula.
- No solapamiento de eventos para estudiantes.
- Un solo evento por salón a la misma hora.

Restricciones Blandas

Reflejan criterios de calidad y comodidad. Cada violación se penaliza en la función objetivo:

- Evitar muchas clases seguidas.
- Evitar una sola clase en el día.
- No asignar cursos a última hora del día.

 El objetivo principal es lograr una solución **factible** (todas las restricciones duras satisfechas) y luego **minimizar** las violaciones de las restricciones blandas.

La función objetivo se incrementa (+1) por cada Restricción Blanda cometida.

Detalles de la implementación

Información características requeridas por evento	Información características de los salones	Información estudiantes	Información horarios
<p>Matriz (evento x características): 1 si el evento la requiere, 0 en caso contrario</p> <p>Matriz (evento x salón): 1 si el evento (i) se puede realizar en el salón (j), sino 0</p> <p>Matriz (evento_hora_salón): indica la hora y el salón asignados a cada evento</p>	<p>Matriz (salón x características): 1 si el salón cuenta con ella, 0 en caso contrario</p> <p>Vector (capacidad) : en la posición "i" se almacena la capacidad del salón "i"</p>	<p>Matriz (evento x estudiantes): 1 si el estudiante (j) asiste al evento (i), sino 0</p> <p>Matriz (horario_estudiante): indica la hora y salón para cada estudiante</p>	<p>Matriz (evento x horario) : 1 si el evento se programa en esa hora, sino 0</p>

Mecanismos



Memoria a Corto Plazo: Lista Tabú

Prohíbe temporalmente movimientos recientes para evitar caer ciclos.



Memoria a Largo Plazo

Registra la frecuencia de movimiento de cada evento para identificar áreas menos exploradas.



Reubicación Forzosa

Si la solución no mejora, se fuerzan movimientos en eventos poco modificados para explorar nuevas regiones.



Criterio de Aspiración

Permite violar la lista Tabú si se encuentra una solución mejor.

Metodología en Cuatro Fases



Fase 4: Búsqueda Tabú para optimalidad



Fase 3: Búsqueda Tabú para factibilidad

Generar nuevas configuraciones con el objetivo de mejorar el índice de sensibilidad.



Fase 2: Búsqueda local para factibilidad

Se cuenta las restricciones duras y blandas que son violadas (índice de sensibilidad)



Fase 1: Construcción

Algoritmo constructivo (greedy) que asigna eventos a los salones con las características requeridas

Fase 1: construcción inicial

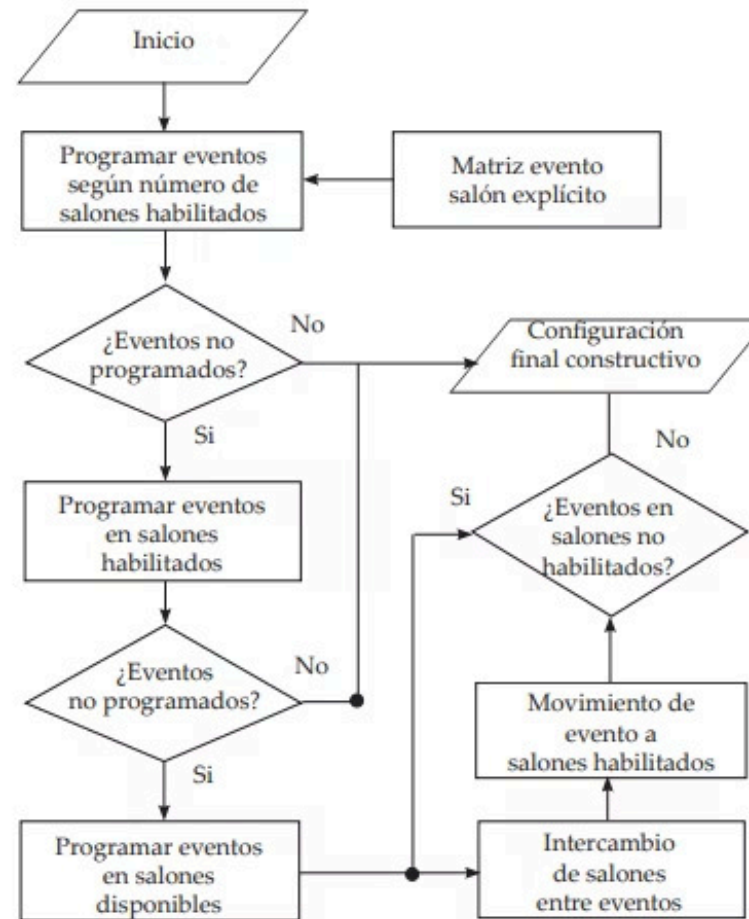


Figura 2. Diagrama del algoritmo constructivo

Cálculo de función objetivo

Para evaluar la función objetivo de una programación de horarios, se cuentan las restricciones duras y blandas que son violadas.

Para la revisión de las restricciones blandas, se debe generar el horario de los estudiantes; conocida la programación de los eventos y los estudiantes que los asisten, se usa la matriz evento_estudiante_explicito, que guarda la información de la matriz eventos_estudiantes de manera eficiente, aprovechando su tendencia esparcidora, dado que no todos los estudiantes asisten a un evento en particular. De esta manera, la generación de los horarios se hace de forma eficiente: se revisa el de cada estudiante y día por día; si se presentan más de dos eventos consecutivos o eventos únicos, constituyen violaciones de restricciones blandas

Para conocer el número de estudiantes que tienen programados eventos en una hora final de un día, se revisa en cada salón, a la hora final de cada día, si existe un evento programado; en dicho caso, el número de restricciones blandas violadas corresponde al número de estudiantes que asisten al evento referido.

Índice de sensibilidad

De forma simultánea a la evaluación de la función objetivo de una configuración, se crea un índice para cada evento; mediante un contador que almacena el número de problemas asociados a este, teniendo índices diferentes para las restricciones duras y las blandas, según sea el tipo de problema. Con estos índices se orienta el proceso de búsqueda.

Fase 3: Estructura de la vecindad

Seleccionando un evento, se pueden generar configuraciones vecinas a partir de:

- Intercambio del bloque de tiempo del evento con otro suceso en el mismo salón
- Movimiento del evento a un bloque de tiempo libre en un salón habilitado

Para seleccionar los eventos que sirven para generar la vecindad se usa el índice de sensibilidad que se forma al calcular la función objetivo, los eventos que cambian son los que presentan un índice mayor.

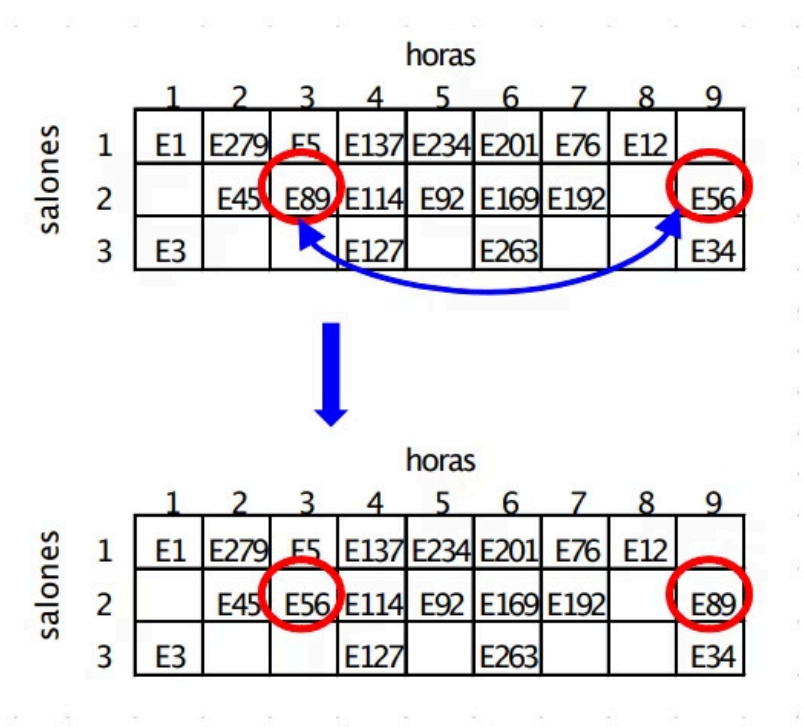


Figura 4. Intercambio de eventos

Búsqueda Tabú y memoria a corto plazo

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. La memoria de corto plazo consiste en un vector (estado tabú) que almacena el estado de prohibición de un evento. Cuando se ejecuta un movimiento en el que participa un evento, la posición del vector estado tabú asociado a ese evento cambia a un valor_tabú, el cual corresponde al número de iteraciones en que se prohíben movimientos que involucren al suceso que cambió. Los elementos del vector estado_tabú diferentes de cero se disminuyen en una unidad en cada iteración hasta que lleguen a cero, estado en el que se permiten movimientos que cambien la programación del evento relacionado.

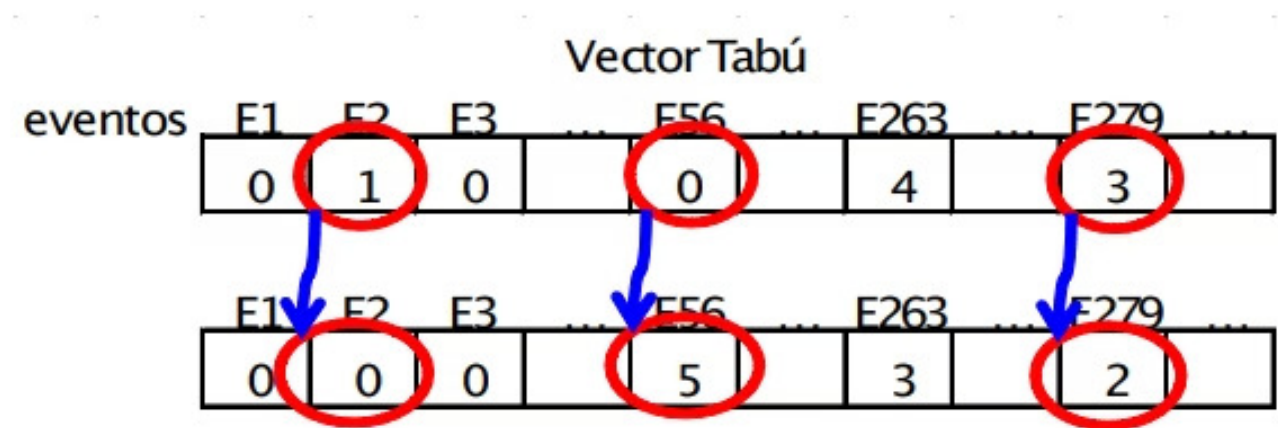
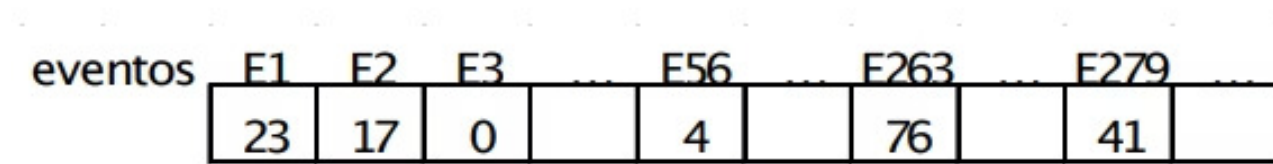


Figura 6. Manejo del vector estado_tabú

Fase 4: Diversificación usando memoria basada en frecuencia

La memoria propuesta basada en frecuencia almacena el número de veces en que la programación de un evento ha sido modificada. Esta memoria se implementa en un vector con dimensión igual al número de eventos.



eventos	E1	E2	E3		E56		E263		E279	
	23	17	0		4		76		41	

Figura 7. Memoria de largo plazo

La estrategia de diversificación emplea la memoria de largo plazo, usando un vector ordenado de forma ascendente según el número de cambios de un evento. De ese vector ordenado se toman los primeros elementos (que corresponden a aquellos sucesos que menos han cambiado) donde el número de eventos y el número de iteraciones con los que se hace diversificación, son parámetros adicionales de Búsqueda Tabú. La programación de los eventos seleccionados es cambiada de manera forzada, para que se dirija la búsqueda a regiones no visitadas (o poco frecuentadas).

Resultados y Comparativas

Se evaluó el rendimiento del método propuesto usando instancias estándar, con 400 segundos de tiempo de búsqueda. La siguiente tabla presenta las violaciones de **restricciones blandas** obtenidas por diferentes algoritmos en distintas instancias:

Tabla 3
Condiciones casos de prueba

	I	II	III	IV
Eventos	400	400	400	350
Salones	10	10	10	10
Características	10	10	10	5
Estudiantes	200	200	200	350

Tabla 4
Resultados de la optimización para los casos de prueba

	I	II	III	IV
Restricciones blandas	99	66	113	101
Horas únicas en un día para un estudiante	4	7	4	20
Más de dos horas continuas para un estudiante	64	39	46	81
Evento en hora final de día para un estudiante	31	20	63	0

Tabla 5
Comparación de resultados

	I	II	III	IV
Búsqueda Tabú propuesta	99	66	113	101
Ganador competencia con Simulated Annealing	45	25	65	44
Algoritmo memético [X1] (prom. 10 corridas)	104	91	126	127
Algoritmo híbrido [X2] (mejores resultados)	57	31	61	5

¿Preguntas? Muchas Gracias

Fuentes:

http://www.scielo.org.co/scielo.php?script=sci_arttext&pid=S0122-34612008000200011&lng=en&nrm=iso&tlng=es

