UNIVERSIDAD DE SONORA

DIVISIÓN DE CIENCIAS EXACTAS Y NATURALES

DEPARTAMENTO DE MATEMÁTICAS

TRABAJO DE TESIS INTITULADO

"EL MÉTODO DE BÚSQUEDA TABÚ PARA LA PROGRAMACIÓN DE HORARIOS"

QUE PARA OBTENER EL TÍTULO DE LICENCIADA EN MATEMÁTICAS
PRESENTA:

PAULINA DANAE LÓPEZ CEBALLOS

HERMOSILLO, SONORA. JUNIO DEL 2000

T E M A R I O

INTRODUCCIÓN

Desde finales de la década de los 60's se han desarrollado diferentes métodos matemáticos para el diseño de modelos que resuelvan el problema de la asignación de horarios escolares sin haber llegado a una solución definitiva, ya que, la programación de horarios depende de varios factores, que se van complicando a medida que se tratan de organizar los niveles educativos más altos; esto es por lo siguiente: en la educación básica el número de grupos coincide con el número de maestros, y las asignaturas (materias) están fijas en cada ciclo escolar. En el nivel medio el número de grupos está fijo en cada escuela y por lo general se mantienen homogéneos hasta el tercer año escolar, es decir el número de grupos del primer año escolar es el mismo que en el segundo y tercer año. Las materias están fijas y lo único que varía es el maestro que imparte la asignatura, así que el problema solo se reduce a evitar que un maestro esté asignado a dos grupos en la misma hora. La solución para estos dos niveles educativos consiste en asignar a los maestros y no a los grupos, ya que estos últimos están fijos. Este problema es relativamente sencillo.

En el nivel medio superior las asignaturas cambian de escuela en escuela, y el número de grupos no coincide con el número de maestros disponibles ni se mantiene uniforme en cada ciclo escolar. El problema se empieza a complicar ya que un maestro puede impartir clases en un grupo ó más y una ó varias asignaturas relacionadas con su área, de manera que la organización del horario debe cuidar que un mismo maestro no esté asignado a la misma hora en dos grupos diferentes o que le corresponda una asignatura en la que no este calificado; aquí todavía tenemos una homogeneidad en el número de alumnos por grupo y el número de materias asignadas a cada grupo, pero puede suceder que dos grupos del mismo ciclo escolar no lleven las mismas asignaturas, ya que se tienen áreas diferentes dependiendo del perfil que se desee estudiar para ingresar al nivel superior; por ejemplo se tienen áreas de fisico-matemáticas (para las ciencias naturales y exactas ó ingenierias) se tiene el área de humanidades, el área de medicina y seguridad social, el área de ciencias sociales (para las carreras de Derecho, Psicología, Sociología) y el área de comunicación (para carreras como periodismo) por mencionar algunas áreas. Así que, aunque el número de asignaturas por grupo permanece igual, las asignaturas en sí son diferentes, por esto, el número de grupos por ciclo escolar va cambiando. En esta parte hay que cuidar que las materias del área no se traslapen con las materias del tronco común (es decir las que no son del área, y que se imparten a todos los grupos), y que los alumnos estén bien asignados al área de estudio.

Cuando la organización de un horario se realiza en la educación superior (universitaria), se tiene un problema muy complicado, aquí el número de maestros por carrera es diferente, el número de grupos por carrera es diferente, el número de horas clase de cada materia también es distinto, los horarios no son iguales, y los alumnos pueden llevar el número de asignaturas que deseen; además se tienen restricciones del número mínimo y máximo de asignaturas para ser asignadas a determinado maestro (esto tiene que ver con su tipo de contratación), además el número de salones disponibles no está fijo, cambia cada semestre ya sea porque se construye un edificio nuevo, se abre otra carrera, o porque se reacomoda para laboratorio. Aquí el problema no solo consiste en cuidar que no se traslapen las asignaturas en los grupos, si no que además no se traslapen los grupos en las aulas disponibles y que los maestros no tengan que impartir dos asignaturas en el mismo horario. Aquí también hay que cuidar que a determinado maestro le corresponda el número de grupos que tiene asignados según su contrato, y además que no exceda de un tope establecido como máximo.

CAPITULO I

ANTECEDENTES Y ALGUNOS MODELOS DESARROLLADOS

1.1 ANTECEDENTES

Se han realizado diversos trabajos al rededor del mundo para tratar de resolver el problema de la asignación de horarios, cada uno de ellos trata un problema particular, y en muchas ocasiones se han obtenido resultados muy buenos cuando son aplicados en una escuela específica, con pocos alumnos y de educación elemental, pero conforme se utilizan en otras escuelas o se llevan a cabo en escuelas de educación superior, el método utilizado empieza a tener fallas de manera directamente proporcional al nivel educativo utilizado o al número de grupos que se tratan de acomodar en el horario. Ninguno de estos métodos ha llegado a tener una solución cien por ciento efectiva y que sirva para todas las escuelas y todos los niveles educativos.

Sin embargo los resultados obtenidos han mejorado de manera muy efectiva el proceso empleado con anterioridad, que en los casos estudiados, es el proceso manual. Uno de los principales avances que se han tenido es en el tiempo empleado para hacer la programación, el cuál se reduce significativamente en todos los casos (de días a minutos). Esto se debe a la utilización de técnicas computacionales, lo que es una ventaja, ya que permite hacer tantas modificaciones a la programación como se necesite, hasta obtener el mejor resultado posible en poco tiempo. Otra ventaja es que se pueden involucrar en la programación tantas especificaciones y limitaciones como se quiera, ya que las técnicas computacionales permiten manejar un número grande de variables sin que afecte significativamente el tiempo de programación.

A continuación se describen algunos de los métodos que han sido utilizados para la programación de horarios. Se eligieron éstos, ya que han sido de los más comúnmente utilizados para la programación de horarios, no solo en escuelas sino también en procesos de producción, programación de vuelos, programación de robots, etc. También se incluye un ejemplo sencillo, en cada caso.

1.2 ALGUNOS MODELOS DESARROLLADOS

En esta segunda parte se van a presentar tres modelos matemáticos que fueron desarrollados para resolver problemas de programación de horarios, las tres técnicas utilizadas son diferentes entre si y también en relación a la desarrollada en esta tesis.

La idea es dar un panorama general (utilizando otros métodos) de lo que significa resolver un problema de gran magnitud, como lo es el de la programación de horarios.

1.2.1 Modelo de programación lineal entera.

Este modelo se empezó a desarrollar en la década de los 60's. Para resolver problemas de programación de horarios, en 1969, N.L. Lawrie [1] retoma la última versión que se tiene de este modelo y la modifica para resolver un problema de programación de horarios de una escuela elemental (primaria) en Inglaterra, con 1060 alumnos y 6 años escolares.

Este modelo se basa en una lista de objetos, los cuales representan los datos de entrada para la computadora; estos objetos pueden ser maestros, grupos de alumnos, salones de clase o piezas de equipo como: proyectores, videocaseteras, etc. Los cuales deben estar disponibles simultáneamente para que puedan coincidir en un número específico de horarios en la semana. Es decir, un determinado maestro y un determinado grupo de alumnos, coincidiendo simultáneamente en un salón asignado en un horario específico, quizás utilizando un proyector.

En este ejemplo el autor no considera los objetos como unidades aisladas, sino como unidades conjuntas: "maestro - horario - grupo de alumnos - salón de clase", que coinciden al mismo tiempo. El autor considera que se deben ver a los grupos de estudiantes con sus materias, sus salones y sus maestros como una unidad, cada conjunto de unidades forman un *plan* y un conjunto de *planes* forman lo que será la programación total de esta escuela.

Definiremos un plan como un informe de la currícula escolar y su organización para un grupo de alumnos dentro de un período de tiempo, en este caso un año escolar.

Aunque el autor no especifica detalladamente como se utilizan las expresiones de los requerimientos en los *planes*, nos da una descripción breve con un ejemplo del tercer año escolar para esta escuela. También nos señala las ventajas y desventajas de éste método.

La tabla 1.1.1 nos representa un *plan* para el tercer año escolar, la columna A del *plan* especifica los requerimientos para 5 clases de Inglés, 2 de latín y 4 de comercio, para ser programados durante 6 períodos a la semana. La columna B especifica que se requieren 4 clases de Inglés, 1 de manualidad, 5 de materias técnicas, 1 de corte y confección y 1 de un conjunto " X ", el cuál es un acuerdo para agrupar a las materias misceláneas que utilizan el menor tiempo semanal y que no se evalúan, como son: arte o apreciación artística, apreciación musical ó clase de canto, educación física y educación religiosa. Las demás columnas pueden ser interpretadas en forma similar.

Aunque esta manera de formar los *planes*, no deja claro qué alumno va con qué clase en qué horario; ni tampoco qué maestros de Inglés, Latín o Comercio serán requeridos en la columna A o cuáles salones serán ocupados por qué grupo; el autor comenta que esta información estará disponible para realizar la programación que será redactada para cada *plan*, y puede estar incluida al pie de página de cada *plan*, como se ilustra en la tabla 1.1.1.

Esta forma de ver el problema simplifica la planeación de la currícula y su organización para cada grupo de alumnos, ya que provee una notación compacta con énfasis en los requerimientos del personal para cada departamento y omite referirse explícitamente a las clases o a como los alumnos son reagrupados dentro de los conjuntos (o grupos), en un período de tiempo. Este tipo de notación hace al método relativamente fuerte ya que, al tratar a un año escolar como una unidad para propósitos de planeación curricular aún en escuelas grandes, evita que los alumnos terminen dispersos en varios salones para tomar sus materias o que tengan horas "libres" (sin clase) en un horario intermedio, lo que ocurre en algunas escuelas, simplemente por la dificultad que se presenta a la hora de realizar la programación para años escolares grandes.

La desventaja que se podría pensar en esta forma de ver a un año escolar como un todo, es que le transfiere al *plan* la carga de determinar no solo la currícula y su ambiente (la cuál aparentemente es elaborada por la computadora), sino también los traslapes en los conjuntos o clases a lo largo de la semana. Por ejemplo, si un año escolar se divide en "áreas" o "especialidades", las cuales no se deben mezclar a lo largo de la semana, este sistema de organización por medio de *planes* podría generar que: se programen dos materias simultáneamente; que se programen dos materias que no están relacionadas (no son de área o especialidad) o que queden espacios en la programación indeterminados. El autor menciona que para salvar este problema se analice "qué es mejor" antes de tomar una decisión.

La tabla 1.1.2 ilustra cinco *planes*, los cuales describen la curricula y su organización para los seis años escolares de una escuela primaria en Inglaterra que denotaremos como "B". La aparente discrepancia entre los planes y los años escolares se debe a que 5to y 6to año se agruparon en un solo plan porque coinciden en su curricula.

Muchos detalles de los años escolares han sido suprimidos, ya que muchos maestros pueden impartir varias materias (asignaturas) las cuales pueden especificarse de muchas maneras diferentes en el *plan*. Lo que si se especifica en la tabla 1.1.2 es por ejemplo, la forma en que las materias prácticas son organizadas, en particular, cómo dos períodos de la materia "técnica / manualidad" son usados por dos materias de lenguaje para el latín. También se especifica en la tabla 1.1.2, la forma en que las materias de inglés y matemáticas son organizadas. Existe suficiente personal académico para hacer posible los arreglos simples, es decir la ocurrencia simultanea de todos los grupos de inglés y matemáticas en cada año escolar. Lo que no se especifica es la organización del resto de la curricula la cuál consiste de francés, historia, geografía, latín (solo dos de los cinco periodos son especificados), así como las clases denotadas como "X" y otra gran cantidad de materias denotadas por "Y" en los planes.

La tabla 1.1.3 muestra a los datos listos para ser introducidos a la computadora. Cada columna de cada *plan* nos da una línea de la tabla 3, los cuales especifican el número de períodos y el personal docente requerido por departamento para ese período. Al pie de la tabla 1.1.3 se localiza el número de maestros requeridos de cada departamento. Note que existe una aparente discrepancia entre las columnas "X" y "Y" de la tabla 1.1.3 y los grupos "X" y "Y" especificados en el *plan*. Por ejemplo, la columna 1 del 3er. año escolar en la tabla 1.1.2, especifica 3 clases de los conjuntos "X" y la línea 20 de la tabla 1.1.3 (que es su correspondiente) indica 5 requeridos por el departamento "X". La explicación consiste en que el número de maestros disponibles para impartir las materias es deducido del número de maestros disponibles en los departamentos de Música, Arte, Educación Física y Educación Religiosa, lo que genera un número mayor.

Note también que los *planes* en la tabla 1.1.1 y 1.1.2 son expresados en términos de materias impartidas, mientras que la tabla 1.1.3 es expresada en términos de requerimientos departamentales.

Los datos que se requieren para elaborar los *planes* consisten en: 1) personal académico por departamento, 2) materias a impartir por los departamentos y 3) la codificación que se usará en los datos de entrada y salida.

Para la escuela B, se tienen los siguientes datos:

DEPARTAMENTO	PERSONAL ACADÉMICO DISP.	MATERIAS HA IMPARTIR
Inglés	9	Inglés
Matemáticas	9	Matemáticas
Ciencias	9	Ciencia General, Física, Química, Biología
Técnica	6	Técnica
Manualidad	4	Manualidad, Corte y Confección
Comercio	4	Comercio, Organización Económica
Clásicos	3	Latín, Griego, Y
Lenguaje Moderno	9	Francés, Alemán, Español, Y
Estudios Modernos	8	Estudios Modernos, Historia, Geografía, Y
Música	3	Música, X, Y
Arte	4	Arte, X, Y
Educación Física	4	E.F. X, Y
Educación Religiosa	1	E.R. X, Y

Note que el número de maestros disponibles para impartir la materia X es 12 y para la materia Y 32. Las siguientes letras nos muestran la codificación usada en la tabla 1.1.3 y 1.1.4.

M: Matemáticas	H: Manualidad	E: Inglés
Z: Estudios Modernos	L: Clásicos	S: Ciencia
F: Lenguajes Modernos	A: Arte	U: Música
X: Departamento X	T: Técnica	Y: Departamento Y
C: Comercio		

FORMULACIÓN MATEMATICA.

Dos cosas hay que tomar en cuenta en este problema para poder resolverlo:

1. Evitar que los *planes* se traslapen con otro, de tal forma que las restricciones del personal no se violen durante un periodo semanal. Este resultado es el que denotaremos como "línea de salida" en lo sucesivo.
2. Las permutaciones en las "líneas de salida" para satisfacer los requerimientos en la distribución de las clases a través de la semana.

Solo el primer paso ha sido programado, por lo que la formulación matemática se da solo para éste, comenta el autor.

Se define para cada grupo (de 30 alumnos) una columna, estos grupos son vistos como un salón donde se realiza cierta actividad educativa, ya sea laboratorio, salón de música, de artes manuales o académicas. Además cada grupo tiene un número específico de horas que debe cubrir a la semana y un conjunto de materias (o asignaturas) que se deberán asignar.

Definimos entonces, un *arreglo* como un conjunto de columnas, uno para cada *plan*, las cuales no violan ninguna restricción del personal. Por ejemplo, para los cinco *planes* de la tabla 1.1.2, la columna B del primer plan, la columna F del segundo plan, la columna G del tercer plan, la columna B del cuarto plan y la columna E del quinto y sexto forman un arreglo para el total de personal docente por departamento para enseñar las clases especificadas en estas columnas, y este total es menor que el número de maestros disponibles.

Entonces el problema puede ser visto de la siguiente manera: encontrar tantos arreglos (no necesariamente distintos) como períodos se requieran en la semana. Esto tiene una formulación de programación lineal. Las restricciones deben ser satisfechas por cualquier solución de cada columna para cada *plan*.

Si suponemos que todos los arreglos posibles pueden ser numerados, entonces se define para cada j (un valor entero), una variable no negativa x_j, como el número de veces que ocurre en la *línea de salida* del j - ésimo arreglo. Entonces existen tantas variables como arreglos (N digamos) y tantas constantes (m digamos) como número de columnas en los *planes* del problema (37 para este ejemplo).

Si se numeran las columnas desde 1 hasta m, comenzando en la columna A del primer *plan*, se puede definir entonces a:

b_i = como el número de veces que ocurre la columna i en el *plan*.

a_{ij} = 1 si la columna i aparece en el arreglo j

= 0 en otro caso.

Con $i = 1,2, ..., m.$ y $j = 1,2, ..., N.$

Las constantes en el problema pueden ser escritas entonces como:

$$\sum_{j=1}^{N} a_{ij} x_j = b_i$$

con $i = 1,2, ..., m.$

Tabla 1.2.1

Plan para el tercer año escolar con 260 alumnos										
Columna:	A	B	C	D	E	F	G	H	I	J
Períodos:	6	6	4	2	7	3	4	4	2	2
Conjuntos:	5E	4E	4F	6F	6M	4M	3Hi	3Ch	2Bi	2Bi
	2Lt	1H	1H	5X	4X	6X	2G	3PH	2Ph	2Ph
	4C	5T	5T				2EO	2C	2C	2C
		1DD	2X				4GSC	1A	1A	1A
		1X	1C					1Mu	1Mu	1Mu
								1Ge	1Ge	1Ge
								1sP	1Sp	1Sp
								3X	3X	1H
										4T
										2X

Ejemplos de una curricula estudiantil										
No académicas:										
Construcción	E	T	T	X	M	X	Gsc	Ms	X	T
Comercio	C	E	C	F	X	M	Gsc	Ms	X	X
Académicas:										
Sc/Ge/Sp	Lt	E	F	F	M/X	X/M	Hi/G	Sc/Ge/Sp		
Comercio	C	E	F	F	M/X	X/M	Hi/G	Comercio		
Manualidad/Técnico	E	H/T	H/T	F	M/X	X/M	Hi/G	Ph/Ch	X	H/T

ABREVIACIONES					
E	Inglés	M	Matemáticas	Bi	Biología
Lt	Latín	Hi	Historia	A	Arte
C	Comercio	G	Geografía	Mu	Música
H	Manualidad	EO	Organización Económica	Ge	Alemán
T	Técnica	GSc	Ciencia General	Sp	Español
DD	Corte y confección	Ch	Química	MS	Estudios Modernos
F	Francés	Ph	Física		
X	Materias misceláneas (de menor tiempo) consistentes en arte y Apreciación musical, educación física y religiosa.				

La matriz de coeficientes a_{ij} que es por definición una matriz cero-uno que contiene entre el 85% y el 90 % de ceros en sus entradas, se puede notar que:

$$\sum_{i \in \text{plan } k} a_{ij} = 1$$

TABLA 1.2.2

CURRÍCULA PARA LA ESCUELA B (DE 6 AÑOS ESCOLARES Y 1060 ALUMNOS)										
PRIMER AÑO ESCOLAR (150 ALUMNOS)										
COLUMNA:	A	B	C	D	E					
PERÍODO:	6	6	6	2	20					
MATERIAS:	5E	5M	4Sc	1H	5Y					
			2H	1T						
			2T	4Sc						
				2Lt						
SEGUNDO AÑO ESCOLAR (180 ALUMNOS)										
COLUMNA:	A	B	C	D	E	F				
PERIODO:	6	6	4	2	2	20				
MATERIAS:	6E	8M	6Sc	3T	2T	5Y				
			2H	3H	1H					
			2T	3Sc	3Sc					
					2Lt					
TERCER AÑO ESCOLAR (260 ALUMNOS)										
COLUMNA:	A	B	C	D	E	F	G	H	I	J
PERIODO:	6	6	4	2	7	3	4	4	2	2
MATERIAS:	5E	4E	4F	6F	6M	4M	3Hi	3Ch	2Bi	2Bi
	2Lt	1H	1H	5X	4X	6X	2G	3Ph	2Ph	2Ph
	4C	5T	5T				2EO	2C	2C	2C
		1DD	2X				4GSc	1A	1A	1A
		1X	1C					1Mu	1Mu	1Mu
								1Ge	1Ge	1Ge
								1Sp	1Sp	1Sp
								3MS	3X	1H
								1Hi		4T
										2X
CUARTO AÑO ESCOLAR (260 ALUMNOS)										
COLUMNA:	A	B	C	D	E	F	G	H		
PERIODO:	6	6	6	7	3	4	4	4		
MATERIAS:	4E	5E	6F	7M	2M	4Hi	4Ph	6Ch		
	3Lt	1H	1H	2X	7X	4G	2Bi	1X		
	3C	2T	2T			1X	2Ge	2Ge		
		3X	2X				1Gk	1Gk		
							1Mu	1Mu		
							1A	1A		
							3C	3C		
QUINTO Y SEXTO AÑO ESCOLAR (210 ALUMNOS)										
COLUMNA:	A	B	C	D	E	F	G	H		
PERIODO:	6	4	7	4	4	6	6	3		
MATERIAS:	7E	2Hi	7M	4Ph	4Ch	2Lt	6F	7X		
		2G	1X	2Bi	2Ge	3T	3T			
		1MS		2Ge	2A	2H	2H			
		3X		2ª	2Mu	1C	1X			
				2Mu	1C	3X				
				1C	3X					
				1X						

TABLA 1.2.3.

DATOS DE ENTRADA														
No. línea	Período	E	M	S	T	H	C	L	F	Z	U	A	X	Y
1	6	5												
2	6		5											
3	6			4	2	2								
4	2			4	1	1		2						2
5	20													5
6	6	6												
7	6		8											
8	4			6	2	2								
9	2			3	3	3								
10	2			3	2	1		2						2
11	20													5
12	6	5					4	2						2
13	6	4			5	2							1	1
14	4				5	1	1		4				2	6
15	2								6				5	11
16	7		6										4	4
17	3		4										6	6
18	4			4			2			5				5
19	4			6			2		2	4	1	1	2	8
20	2			4			2		2		1	1	5	7
21	2			4	4	1	2		2		1	1	4	6
22	6	4					3	3						3
23	6	5			2	1							3	3
24	6				2	1			6				2	8
25	7		7										2	2
26	3		2										7	7
27	4									8			1	9
28	4			6			3	1	2		1	1	2	5
29	4			6			3	1	2		1	1	3	6
30	6	7												
31	4									5			3	8
32	7		7										1	1
33	4			6			1		2		2	2	5	7
34	4			4			1		2		2	2	5	7
35	6				3	2	1	2					3	5
36	6				3	2			6				1	7
37	3												7	7
Profesores por Depto.		9	9	9	6	4	4	3	9	8	3	4	12	32

Y como cada arreglo contiene una y solo una columna para cada *plan*, y puesto que:

$$\sum_{i \in \text{plan } k} b_i = N_p$$

(El número de períodos en la semana) y como el *plan* especifica una actividad educativa (columna) para cada periodo en la semana, entonces:

$$\sum_{i \in \text{plan } k} \sum_{j=1}^{N} a_{ij}\, x_j = \sum_{i \in \text{plan } k} b_i$$

Lo que se reduce a:

$$\sum_{j=1}^{N} x_j = N_p$$

Así las m ecuaciones constantes no son independientes. Y como existen k *planes*, tenemos entonces k-1 ecuaciones, una para cada *plan*.

Por lo que la función objetivo quedaría descrita de la siguiente forma:

$$\text{maximizar} \quad \sum_{j=1}^{N} c_j x_j$$

Con c_j definido como el mínimo de los valores bi de las columnas en los arreglos j, que también se usa como límite superior para x_j, ya que los arreglos en la programación pueden tener alta multiplicidad, comenta el autor. Así se reduce el número de arreglos diferentes que pueden ocurrir en una solución. Sin embargo, puesto que la función objetivo tiene una pequeña relación con el criterio de normalidad, los conocedores se inclinan a favor de producir un número de soluciones enteras alternas.

METODO COMPUTACIONAL.

Todas las variables fueron generadas para un método de programación lineal escrito en ALGOL, incorporando el método de formas enteras de Gomory, utilizando la función objetivo descrita anteriormente.

RESULTADOS.

Los resultados presentados a continuación incluyen la escuela B (que describimos en este método) y las escuelas D, G y W.

Escuelas	B	D	G	W
No. de alumnos	1060	700	980	490
No. de profesores	74	45	62	31
No. de años escolares	5	5	3	4
No. de líneas	37	45	28	48
No. total de arreglos	1454	1459	210	2363
Períodos disponibles para profesores	2960	1890	2480	1395
Periodos específicos para profesores en cada año escolar	1789	1282	1208	905
Porcentaje de periodos disponibles para profesores	60.4	67.8	48.7	64.9
Tiempo de Computación en las corridas (minutos)	2.7	11.9	1.3	7

1.2.2 Técnica de coloración

En este segundo ejemplo definiremos la técnica de coloración, primero y después veremos cómo es utilizada para la programación de exámenes Universitarios.

Como los problemas de horarios se basan en el arreglo de eventos que tienen la misma duración y que están sujetos a la condición de que ciertos eventos pueden o no llevarse a cabo al mismo tiempo, estos problemas pueden ser expresados en términos de gráficas. Los eventos son representados por los vértices de la gráfica y se estipula que un par de vértices están conectados por un eje unidireccional si y solo si los eventos correspondientes no toman lugar al mismo tiempo. Organizar los eventos en un horario sujetos a la restricción mencionada es equivalente a colorear la gráfica correspondiente de tal forma que dos vértices no adyacentes tengan el mismo color.

La determinación del mínimo número de intervalos de tiempo que son necesarios para formar el horario es de esta manera, lo mismo que encontrar el mínimo número de colores requeridos para colorear la gráfica. Es bien conocido que encontrar el número de colores requeridos para colorear una gráfica arbitraria es un problema no resuelto aún.

En el caso de la programación de exámenes, los eventos que se van a programar en el horario son los exámenes en si, los cuales requieren de un período de tiempo. La condición será representada por una matriz de conflicto (matriz cero - uno) $C = \{c_{ij}\}$, donde $c_{ij} = 1$ si los exámenes de las materias i y j no toman lugar al mismo tiempo, para que algún candidato pueda realizar ambos exámenes, y $c_{ij} = 0$ en otro caso. El objetivo es determinar el mínimo número de períodos que se requieren para acomodar 500 exámenes [2].

Este problema es resuelto de dos formas diferentes: por ordenación de vértices y por medio de una matriz similar, ambos métodos se comparan al final.

1.2.2.1 Coloreando por ordenación de vértices

Este método consiste en acomodar los vértices en orden decreciente según su grado, esto es, el número de ejes que salen del vértice. El primer vértice comienza con el primer color en todos sus vértices (grupo). Los vértices son inspeccionados en orden: cualquier vértice que no esta conectado a un miembro del primer grupo es agregado al grupo, coloreándolo del mismo color. El segundo grupo comienza con el primer vértice que no ha sido coloreado, y otra vez todos los vértices son inspeccionados en orden: cualquier vértice que no esta conectado a un miembro del segundo grupo es agregado al grupo. El proceso continúa hasta que todos los vértices están coloreados. Este procedimiento es bien conocido en el uso de la técnica de coloración.

Esta claro que cualquier vértice de grado menor que el número de grupos siempre se puede agregar a uno de los grupos existentes, siempre que no este conectado a alguno de ellos. De igual forma una vez que el número de grupos exceda el grado del primer vértice no coloreado de la lista, los vértices restantes pueden ser coloreados sin introducirlos en alguno de los grupos.

Aunque éste método es simple y usualmente produce resultados aceptables, no puede en general garantizar que generará el número mínimo de grupos (colores). Sin embargo, si por ejemplo un subconjunto de n vértices puede ser localizado, y cada par en éste es conectado, se puede probar que n es el valor mínimo. Por ejemplo, con conjuntos pequeños de datos para problemas de horarios, involucrando 20 vértices, se puede determinar el número mínimo y máximo de colores, y mediante un proceso de prueba y error el número exacto de colores es determinado.

Sin embargo cuando se tienen tantos como 500 vértices, los valores mínimo y máximo pueden diferir considerablemente y solo se puede tener una idea general de la gráfica coloreada.

EJEMPLO 1.2.2.1

Considere el ejemplo de la figura 1.2.1, donde se ilustra cómo el número de grupos puede ser innecesariamente incrementado por el camino de la numeración de vértices.

Como todos los vértices son de grado dos, se pueden ordenar de la siguiente manera:

1, 2, 3, 4, 5, 6.

Colorear los vértices de ésta manera produce los siguientes grupos:

(1, 2), (3, 4), (5, 6).

Sin embargo, los vértices pueden estar numerados de forma diferente y por consecuencia ordenados en otra forma:

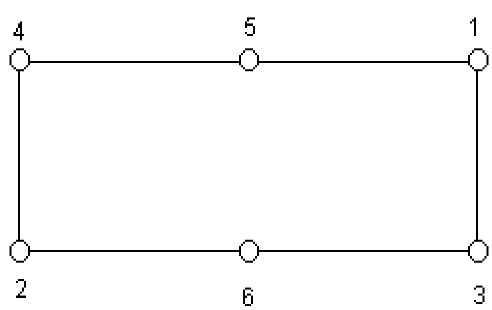
1, 5, 4, 2, 6, 3.

En cuyo caso solo dos grupos son generados:

(1, 4, 6), (5, 2, 3).

Se puede ver que en el primer caso los vértices 1 y 2 están agrupados juntos, y por lo tanto el número de grupos es incrementado.

FIGURA 1.2.2.1



Para evitar esto, analizaremos otra forma de colorear por medio de la formación de una matriz similar.

1.2.2.2 Coloreando por medio de la formación de una matriz similar

El problema es análogo al de la clasificación de objetos por atributos, donde una colección de objetos tiene que ser dividida dentro de compartimentos. Esta técnica es usada en Taxonomía donde se va formando la matriz similar, que señala a aquellos pares de objetos de gran similitud que deben ser puestos en el mismo grupo.

De la misma forma una matriz similar S ={sij} es formada para determinar cuales vértices deben tener el mismo color. La matriz similar encontrada más efectiva es definida como:

$$Sij = 0 \text{ si } cij = 1$$

$$Sij = \sum_{k} (c_{ik} \,\&\, c_{jk}) \text{ si } c_{ij} = 0$$

Donde & es la operación lógica "Y", como se muestra en la Tabla 1.2.1 y k se asume sobre todos los vértices i y j, es decir si i y j no están conectados, la similaridad es la del número de otros vértices k donde i y j están conectados, la similaridad es la del número de otros vértices k donde *i* y *j* están conectados. Cada par de vértices con esta similitud es coloreado de acuerdo al siguiente algoritmo, donde el nivel de similaridad es reducido a uno, y de nuevo cada par de vértices con esta similaridad es coloreado. La matriz similar es revisada repetidamente, reduciendo el nivel de similaridad en cada paso, hasta que todos los vértices están coloreados. Los vértices de grado menor que el número de grupos no se colorean hasta que puedan ser agregados a uno de los grupos existentes.

TABLA 1.2.2.2.1

	0	1
0	0	0
1	0	1

ALGORITMO DE COLORACIÓN

El siguiente procedimiento es adoptado para colorear un par de vértices *i* y *j*, donde ambos, uno o ninguno de los vértices está ya coloreado.

(a) Los vértices *i* y *j* ya están coloreados

1. Continuar con el siguiente par.

(b) Uno de los vértices (digamos *i*) está coloreado como el grupo G y el otro vértice *j* no está coloreado aún.

1. Si el grado de *j* es menor que el número de grupos, entonces *j* puede ser coloreado y es ignorado; continuar con el siguiente par.
2. Tratar de agregar el vértice *j* al grupo G, i.e. si $c_{jk} = 0$ para cada vértice k en el grupo G, entonces agregar *j* a G; continuar con el siguiente par.
3. Continuar con el siguiente par si *j* no puede ser agregado a G.

(c) Ninguno de los vértices *i* y *j* está coloreado.

1. Si el grado de ambos vértices es menor que el número de grupos, entonces son ignorados.
2. Encuentra el primer grupo G donde i y j pueden ser agregados, i.e. si $c_{ik} = 0$ y $c_{jk} = 0$ para cada vértice k en el grupo G.
3. Si *i* y *j* no pueden ser agregados a algún grupo existente, entonces forma un nuevo grupo con estos vértices.

EJEMPLO 1.2.2.2.1

La matriz de conflicto y la matriz similar, obtenidas de la Tabla 1 se muestran en la Tabla 2. De la matriz similar se puede ver que (1, 4, 6) y (2, 3, 5) tienen la mayoría de similaridades. El algoritmo anterior combina los vértices en este sentido, formando dos grupos, donde la coloración es independiente de la numeración de los vértices.

TABLA 1.2.2.2.2

MATRIZ CONFLICTO

C_{ij}	1	2	3	4	5	6
1	0	0	1	0	1	0
2	0	0	0	1	0	1
3	1	0	0	0	0	1
4	0	1	0	0	1	0
5	1	0	0	1	0	0
6	0	1	1	0	0	0

MATRIZ SIMILAR

S_{ij}	1	2	3	4	5	6
1	0	0	0	1	0	1
2	0	0	1	0	1	0
3	0	1	0	0	1	0
4	1	0	0	0	0	1
5	0	1	1	0	0	0
6	1	0	0	1	0	0

IMPLEMENTACIÓN

Ya que los requerimientos de almacenaje se prohiben para la matriz de conflicto cuando los conjuntos son muy grandes y además como la matriz consiste de ceros mayormente, se puede almacenar una lista para cada materia del conjuntos de materias de cada clase que se va ha examinar. También puede haber puntos de almacenamiento para la matriz similar completa donde los elementos son requeridos en orden descendiente a su magnitud, y donde la mayoría de ellos sean cero. Cada nivel de similitud es de esta manera almacenada como una lista de pares de cada materia con similitud, evitando así que se tenga que revisar toda la matriz repetidamente.

Cuando los ejemplos son grandes, la lista de pequeñas similaridades (menos de cinco), puede ser enorme y de valor pequeño, y puede no ser recordada. Cualquier materia que permanezca, puede ser fácilmente coloreada cuando la similitud es reducida a este nivel.

APLICACIÓN Y CONCLUSIÓN

Los autores Welsh & Powell propusieron en 1967, la siguiente formula para obtener una frontera superior para el número de colores de una gráfica:

$$\max_{i} \min (i, d_i + 1)$$ donde d_i es el grado para los vértices i.

La Tabla 1.2.3 muestra tres estimaciones para el número de colores que fue desarrollado por computadora usando varios ejemplos prácticos.

Si bien, los ejemplos numéricos indican que el método de matriz similar es mejor que el método por ordenación de vértices en solo dos casos, una búsqueda exhaustiva ha verificado que el número de colores se obtiene en todos los casos. En todos los ejemplos, la frontera superior del número de colores utilizando la formula de Welsh & Powell es la de menor valor.

Los dos primeros ejemplos en la Tabla 1.2.3 se refieren a los exámenes de una escuela secundaria y los últimos cuatro se refieren a los exámenes universitarios.

Una comparación más detallada de los dos métodos se hizo usando un generador de datos aleatorios. El generador de números aleatorio es usado para producir la matriz de conflicto con una probabilidad dada para dos materias conflictivas. Cada matriz es coloreada por ambos métodos y el porcentaje de matrices para los cuales el método de similitud es mejor (i.e. que requiera menor número de grupos), es igual o es peor, está dado en la Tabla 1.2.4. Las matrices usadas son de orden: 20, 50 y 100 y las probabilidades de conflicto son: 0.25, 0.5, y 0.75. También se incluye en la tabla el rango de estimaciones para determinar el número de colores usado para cada tipo de matriz

TABLA 1.2.2.2.3

Comparación en la estimación del número de colores.

NÚMERO DE MATERIAS A EXAMINAR	α	β	δ
71	10	10	30
116	20	20	50
204	10	10	21
380	9	9	15
498	15	13	35
569	19	18	48

1. α = número de colores obtenidos por ordenación de vértices.
2. β = número de colores usando el método de matriz similar.
3. δ = frontera superior propuesta por Welsh & Powell.

TABLA 1.2.2.2.4

Comparación de estimaciones del número de colores en matrices aleatorias obtenido por los métodos de: matriz similar y ordenación de vértices.

Número de vértices	Probabilidad de conflicto	% por el que el método de matriz similar es:			Rango para el número de colores
		Mejor	Igual	Peor	
20	0.25	21	61	18	3 - 5
20	0.50	26	54	20	5 - 8
20	0.75	21	67	12	8 - 10
50	0.25	8	66	26	6 - 8
50	0.50	30	52	18	10 - 13
50	0.75	48	31	21	16 - 20
100	0.25	7	20	73	10 - 13
100	0.50	13	50	37	18 - 22
100	0.75	72	20	8	28 - 33

Se obtienen dos conclusiones de la Tabla 4.

1. El método por ordenación de vértices usado no proporciona el mejor resultado en aproximadamente el 25% de los casos, ya que el método de matriz similar produce mejores soluciones.
2. Con matrices pequeñas de orden 20, existe una pequeña similitud en los métodos, la diferencia entre la estimación del número de colores raramente excede a uno. Con matrices grandes, de orden 100, los métodos difieren considerablemente. El método de ordenación de los vértices es mejor cuando la probabilidad de conflicto es menor (i.e. la matriz de conflicto es poco densa), el método de matriz similar es mejor cuando la probabilidad de conflicto es alta.

Por último analizaremos otra técnica (de programación entera) usada en la programación de cursos, este método se basa en la división del problema en dos subgrupos, uno relacionado con los grupos exclusivamente y el otro relacionado con los horarios exclusivamente, los cuales se relacionan a su vez, entre sí.

Éste método es el que mayor parecido tiene con el método de búsqueda tabú, que es el utilizado en la programación de los horarios de este trabajo, ya que utiliza las permutas en los elementos de cada subgrupo para mejorar la solución obtenida previamente, hasta obtener la mejor solución encontrada, que conocemos como ÓPTIMA.

1.3 MODELO DE SUBGRUPOS

Este modelo [3] incluye dos componentes principales (el subproblema del horario y el subproblema de la agrupación) que interactúan vía la función objetivo. Primero, se van a especificar algunos conceptos que caracterizan al problema.

Cada estudiante s es registrado en un conjunto de cursos R_s. Los cursos donde los estudiantes son registrados se repiten cierto tiempo durante una semana. Así que se deberá determinar los grupos de estudiantes registrados en cada curso w (subproblema de la agrupación) y una sección de un curso específico $a \in Q_w$ es asociada con cada grupo. Note que un curso w que no se repita incluye solo una sección (i.e. $|Q_w| = 1$).

Cada sección del curso incluye un número específico de materias en clase cada semana, la duración de esas materias no es uniforme, algunas duran una hora, otras dos, etc., por lo que en el subproblema de horario, tenemos que asignar estas materias.

El concepto de conflicto entre las materias se introduce a continuación: Dos materias están en conflicto si se van a realizar (o al menos en parte) simultáneamente y

1. Ambas materias se imparten por el mismo profesor, o
2. Requieren el mismo salón de clase (un laboratorio por ejemplo), o
3. Al menos un estudiante tiene que cursar ambas materias.

La duración del conflicto se define como el número de horas durante las cuales dos materias se llevan a cabo simultáneamente. Ya que la duración de las materias no es la misma, se sigue que dos materias que comienzan a distinta hora pueden estar en conflicto.

Por ejemplo, si una materia de dos horas i comienza a la hora j que corresponde a las 8:00 a.m. del lunes y otra materia de una hora k comienza a la hora l que corresponde a las 9:00 a.m. del lunes, entonces ambas materias se llevan a cabo al mismo tiempo, de las 9:00 a las 10:00 a.m. y la duración del conflicto es de una hora. Por otro lado, si l es la hora de comienzo de la materia i y j es la hora de comienzo del trabajo k, entonces ambas materias no se traslapan. (Ver Figura 1.2.3.1).

El conjunto de salones disponibles es seccionado en varios subconjuntos, cada uno de los cuales incluyen salones del mismo tipo (note que ciertos subconjuntos son usados para especificar los conflictos del tipo 2). Para cada materia, uno y solo un tipo de salón de clase se requerirá especificar. El modelo incluye dos conjuntos de variables:

(1) variables de horario: para cada materia i y un horario de inicio j, $1 \leq i \leq n$, $1 \leq j \leq m$,

$x_{ij} = 1$ si la materia i comienza en el horario j *y* $x_{ij} = 0$ en otro caso

(2) variables de grupo: para cada estudiante s y cada curso a, $1 \leq s \leq S$, $1 \leq a \leq A$,

$y_{sa} = 1$ si el estudiante s es asignado al curso a *y* $y_{sa} = 0$ en otro caso

Una vez definidas estas variables, el problema puede presentarse entonces como un problema de programación entera, de la siguiente manera:

$$\text{Min } f(x,y) = f_1(x) + \mu f_2(x,y) \qquad \text{(P)}$$

(P) sujeta a

$$\sum_{j=1}^{m} x_{ij} = 1 \quad \text{y} \quad x_{ij} = 0 \text{ ó } 1 \quad \text{con } 1 \leq i \leq n, \; 1 \leq j \leq m$$

$$\sum_{a \in Q_w} y_{sa} = 1 \quad \text{y} \quad y_{sa} = 0 \text{ ó } 1 \quad \text{con } 1 \leq s \leq S, \; 1 \leq a \leq A$$

El modelo de programación entera (P) está definido en términos de constantes independientes para las variables x *y* y con interacciones en la función objetivo. El procedimiento para resolver esta función se reduce en el siguiente algoritmo:

Algoritmo principal

Paso 0 (Inicialización)

Sea x^* *y* y^* una solución viable para (P).

Sea $f^* = f(x^*, y^*)$ y

posible = verdadero

Mientras sea posible hacer

Paso 1(Subproblema de la agrupación)

Dado $x = x^*$ determinar una solución óptima (y^{**}) del subproblema de la agrupación

si $f^* \leq f(x^*, y^{**})$ entonces

posible = falso

de otra forma

$f^* = f(x^*, y^{**})$

$y^* = y^{**}$

posible = verdadero

Paso 2 (Subproblema de horarios)

Dado $y = y^*$ determinar una solución óptima (x^{**}) del subproblema de horarios

si $f^* \leq f(x^{**}, y^*)$ entonces

posible = falso

de otra forma

$f^* = f(x^{**}, y^*)$

$x^* = x^{**}$

posible = verdadero

termina el mientras.

En el paso 1, la proyección es llevado en el espacio de las variables "x", mientras que el paso 2 la proyección es llevado al espacio de las variables "y". Además, este procedimiento es heurístico en el sentido de que no es necesario que converja a una solución óptima. Por otro lado, este problema (P) de programación entera no es probable que se pueda resolver en un tiempo de computo aceptable, por lo que, si el procedimiento heurístico trabaja, entonces será el aceptado.

En la próxima sección, los subproblemas son analizados y se usarán algunos problemas como fuente para analizar el método, donde el objetivo será reducir el número de conflictos. En este contexto, la aproximación que se hará tiene más sentido ya que puede ser vista como un procedimiento completo para reducir el número de conflictos modificando sucesivamente el horario - maestro y la agrupación de los estudiantes. Este tipo de aproximación puede ser usado siempre que las variables del modelo puedan ser seccionadas dentro de subconjuntos (con sus constantes específicas) interactuando solo a través de la función objetivo. De aquí que, el modelo puede ser optimizado sucesivamente para cada subconjunto de variables manteniendo los arreglos hasta que el equilibrio es alcanzado, y la función objetivo no puede ser mejorada.

El problema del control de tráfico donde la función objetivo esta especificada en términos de los costos del congestionamiento, es un ejemplo donde esta metodología puede ser usada.

1.3.1 El subproblema de horarios

Las variables del subproblema de la agrupación toman los valores y^* que representa a los estudiantes que toman ambas materias i y k dada en la agrupación y^{**}. El modelo utiliza un procedimiento de intercambio que modifica los puntos de inicio en el horario para una o dos materias (n en cada iteración). La selección de esas materias que van a ser reasignadas va de acuerdo con la matriz $E = [e_{ij}]$ ($1\leq i \leq n$, $1\leq j \leq m$) donde e_{ij} mide la reducción en el valor de la función objetivo si la materia i comienza en el lugar de la materia j. Mas aún, las entradas en la matriz E también son usadas para evaluar los cambios de reasignar dos materias simultáneamente.

En cada iteración de éste procedimiento, se mide el valor de la función objetivo cuando se reasigna a la materia i en el inicio de horario de la materia j. Cuando la función objetivo no puede ser mejorada con un simple cambio, entonces el procedimiento puede proceder a la doble reasignación donde dos materias son reasignadas simultáneamente. Aunque en la practica la doble reasignación es raramente usada porque el tiempo de computo requerido es muy grande.

Como la matriz es fácilmente actualizada después de la reasignación de las materias, se explica la eficiencia de la aproximación para problemas de tamaño pequeño; desafortunadamente, el tamaño de la matriz E crece rápidamente cuando el tamaño del problema se incrementa, y más importante es el hecho de que el tiempo requerido para inicializar por la computadora se vuelve prohibitivo. Los resultados indican que más del 25% del tiempo para inicializar la matriz E en todas las soluciones de tamaño pequeño crece directamente proporcional al tamaño del problema, por lo que se propone un nuevo procedimiento de intercambio que no requiera el uso de los ahorros de la matriz E, comenta el autor.

Este nuevo procedimiento se basa en las medidas del costo c_{ij}, de iniciar con la materia en *i* en vez de *j*, especificada en términos de la disponibilidad y de los conflictos que sean creados por esta asignación en la función objetivo, los cambios deben incluir un término relacionado con la frontera superior en el número de salones disponibles.

El nuevo procedimiento revisa primero la tabla de horario principal para reducir el número de conflictos y después reduce los costos de disponibilidad de datos y el exceso de uso de salones de clase. Resumiendo se tiene que:

- Durante la primera mitad del procedimiento, todo las fuentes (hora maestro) son revisadas. Si algunas entradas están en conflicto, todos los cambios sencillos en las horas de inicio se llevan a cabo hasta eliminar el conflicto.
- Durante la segunda mitad, los nuevos cambios son aplicados para reducir los costos en los datos de entrada disponibles y el excesivo uso de salones de clase.

1.3.2 El subproblema de la agrupación

Asumimos ahora que la tabla de horario principal x* está disponible, por lo que redefiniendo los grupos de estudiantes en secciones se puede reducir el número de conflictos del tipo 3, debido a los estudiantes.

El algoritmo trata ahora de distribuir a los estudiantes tan homogéneamente como sea posible dentro de las secciones, esto se lleva a cabo mediante el procedimiento de intercambio, tal como se hizo en el subproblema de horario.

EJEMPLO 1.3.2.1

Se utilizó una institución con 3300 estudiantes inscritos, los cuales llevan en promedio 7 cursos, con más de 200 materias disponibles a elegir. El plan de estudios de la Institución incluye 250 cursos que generan 825 secciones, en la mayoría de las cuales se imparten una o dos materias por semana, sin embargo, algunos de estos cursos imparten más de cinco materias. Esto genera un problema con más de 100 materias para ser organizadas en el horario, donde cada materia tiene una duración de una a cinco horas diarias.

La institución tiene 85 salones disponibles, los cuales son seccionados en 5 tipos, además de 30 salones específicos (subconjunto simple) usados para los conflictos específicos del tipo 2. Cada semana incluye 45 horas de clase que generan 45 horas de inicio.

Para inicializar el procedimiento, se requiere de un horario inicial y una agrupación dada, por lo que se tomó la programación usada el año anterior. Esta programación inicial se dividió en dos conjuntos de acuerdo a dicha agrupación . El primer conjunto, *agrupación inicial balanceada* es donde los estudiantes están bien distribuidos a lo largo de las secciones; en el segundo conjunto todos los estudiantes son asignados a la misma sección de cada curso, por lo que se tiene una *agrupación inicial desbalanceada*.

El problema se resolvió para diferentes valores en los parámetros definidos a continuación:

μ = factor de penalización para el conflicto tipo 2 debido a los estudiantes.

ν = factor de penalización para el conflicto tipo 1 debido a las materias que requieran un salón específico.

ρ = factor de penalización para el uso excesivo de salones de clase.

La calidad de las soluciones se midió con respecto a cuatro características diferentes:

1. Número de conflictos del tipo 3, debido a los estudiantes.

NCS = total de horas durante las cuales los estudiantes son programados para llevar dos materias simultáneamente (donde, si 3 estudiantes son programados para llevar al mismo tiempo dos materias entonces el valor de NCS incrementa a 3).

2. Número de conflictos de tipo 1 y 2 debido a las materias en un salón específico.

NCPS = total de horas durante los cuales existen conflictos del tipo 1 y 2.

3. Distribución de los estudiantes en las secciones.

DIS = suma de estudiantes en cada sección, obtenida como una relación entre el número de estudiantes registrados en cada curso y el número de secciones de cada curso.

4. Falta de salones.

NP = total de horas durante las cuales existe una falta de salones.

La eficiencia del procedimiento puede ser inferido por la siguiente información:

1. El CPU = tiempo (en segundos) que se requiere para cada algoritmo (el de agrupación y el de horario).

2. LECDEP = al número de reasignaciones sencillas ejecutadas por el algoritmo de horario.

La primera serie de pruebas (cuatro pruebas) se llevó acabo evaluando el efecto de incrementar el parámetro ν manteniendo μ y ρ fijo. El número de conflictos del tipo 1 y 2 (NCPS) decreció conforme ν aumentó, sin embargo, para $\rho = 0.0001\ \mu$ (i.e. la penalización para el excesivo uso de salones de clase no tuvo relevancia) la solución generada en estas cuatro pruebas indicó una falta de salones de algún tipo durante NP horas a la semana.

Pero si se mantiene a $\nu = 30\mu$ e incrementamos ρ a 15μ, el número NP se reduce a cero. Más aún, los resultados en la quinta prueba muestran una solución de relativa calidad para NCS, NCPS y DIS.

En la sexta prueba el problema se resolvió usando los mismos valores para los parámetros como en la prueba cinco, pero el subproblema de la agrupación se resolvió con la variante de que el paso 4 del algoritmo se repitió para cada estudiante hasta el final, con el propósito de reducir el valor de DIS más rápidamente. El resultado indicó que este propósito se cumplió, pero el valor de DIS en la solución final empeoró. Por otro lado, el número de iteraciones se redujo y así el tiempo CPU.

Finalmente, en las pruebas 7 y 8 la agrupación inicial está muy desbalanceada y esto se refleja en el gran valor para DIS en la solución final. Por otro lado, es interesante ver la gran mejoría en el valor de NCS cuando es comparada con su valor en las pruebas 5 y 6, en este sentido, es importante notar que se puede mejorar significativamente el número de conflictos si permitimos una menor satisfacción en la distribución de los estudiantes.

El tiempo de cómputo requerido para cada iteración fue directamente proporcional al número de conflictos que aún no han sido eliminados, es decir, al avanzar el algoritmo el número de conflictos decrece y de igual forma el tiempo requerido para cada iteración.

Los resultados indicaron que 29 de los conflictos NCS del tipo 3 fueron inevitables, ya que por alguna razón dos materias teniendo 29 alumnos en común estaban fijas desde el principio. La mala noticia fue que las reasignaciones dobles nunca fueron usadas en el algoritmo de horario, pero analizando la solución generada después, se vió la posibilidad de eliminar un conflicto (NCPS = 1) de tipo 2 con una doble reasignación.

El número de conflictos finales parece representar un avance significativo sobre los resultados en la programación del horario para esta Institución, en comparación con los que existían antes.

Otras dos versiones de este procedimiento se llevaron a cabo en dos departamentos diferentes de una Universidad y los resultados fueron consistentes con los mencionados anteriormente.

CAPITULO II

EL MÉTODO DE BÚSQUEDA TABÚ

En este segundo Capítulo se describen brevemente los orígenes del método de búsqueda tabú (*introducción*), luego se introduce el problema de optimización combinatoria como antecedente al modelo de búsqueda tabú (*descripción*), por último se describe el modelo de búsqueda tabú aplicado al problema de optimización combinatoria (*modelo de búsqueda tabú*) y se aplica en un ejemplo conocido como "el problema de las n - reinas".

Esto con la finalidad de entender el modelo de búsqueda tabú aplicado a un problema más sencillo como lo es el problema de la n - reinas, para finalmente aplicarlo en la programación de horarios (CAPITULO III).

2.1 INTRODUCCIÓN

El método de búsqueda tabú tiene su origen en procedimientos combinatorios aplicados a problemas de curvatura no lineal desarrollados al final de la década de los 70's, aunque muchos autores le dan el crédito inicial a Fred Glover y Pierre Hansen [4] ya que son los primeros autores en desarrollar la técnica en todo su potencial hacia el segundo quinquenio de la década de los 80's, ellos utilizaron ésta técnica para resolver problemas de optimización a gran escala. Este método se ha aplicado para resolver problemas de programación de grupos de trabajo [4], donde se requiere de una secuencia de pasos para obtener un producto determinado, y el objetivo principal es obtener el producto en el menor tiempo posible, así que se desea "descubrir" cual es la distribución del trabajo en los grupos de tal manera que el tiempo sea el menor, cumpliendo las normas de calidad del producto.

Otro ejemplo de optimización es la planeación aérea, donde el objetivo principal es minimizar las rutas de los aviones de tal forma que cumplan con el programa de vuelo, pero que el costo de operación de los aviones sea el menor posible. También se utiliza para elaborar el calendario de vuelo para los pilotos y las aeromozas de modo que todos los vuelos cuenten con el personal requerido y que los horarios sean convenientes para el personal, es decir, que no tengan vuelos largos consecutivos o que no se traslapen sus vuelos.

Los últimos desarrollos involucran el problema del vendedor, donde se desea descubrir la ruta más corta, para que el vendedor pueda visitar todas las ciudades que tiene programadas y regrese al punto de partida, eligiendo el camino más corto; el diseño de circuitos integrados y el problema de programación de horarios, que es el que desarrollamos más adelante.

El problema principal de algunos algoritmos heurísticos es la dificultad de escapar de la optimalidad local, es decir que al encontrar una solución óptima (o aparentemente óptima) el método no se detenga, si no que siga buscando otras posibles soluciones óptimas. Esto a propiciado que los métodos más recientes involucren búsquedas heurísticas utilizando el enfoque de la inteligencia artificial para evitar el problema de la optimalidad local, tales como: algoritmos genéticos, redes neuronales, recocido simulado, análisis de objetivos, búsqueda dispersa y búsqueda tabú, entre otros.

Este método heurístico (búsqueda tabú) está diseñado para escapar de la optimalidad local, basado en el manejo y uso de una colección de principios que sirven para resolver el problema de manera "inteligente", esto es, haciendo uso de memoria flexible para involucrar dos procesos, el de la adquisición y el de mejoramiento de la información; así, al tener cierta "historia" de los caminos ya recorridos y de los óptimos encontrados, se puede evitar permanecer en las mismas regiones, y recorrer regiones nuevas para encontrar otras mejores soluciones.

El método de búsqueda tabú se basa en tres puntos principales:

1. El uso de memorias diseñadas para permitir evaluar la información de búsqueda histórica.
2. Un mecanismo de memoria que restringe y libera el proceso de búsqueda.
3. La utilización de memorias de diferentes lapsos de tiempo: la de término corto, la de término intermedio y la de término largo, para guardar (por un tiempo) aquellas características que lograron una buena solución, y olvidando otras (ya

transcurrido el tiempo de memoria) para permitir al método diversificarse dentro de nuevas regiones.

Además de esto, el método de búsqueda tabú tiene la habilidad de obtener resultados de alta calidad con equipo computacional modesto, lo que es una ventaja, ya que en muchas ocasiones no se cuenta con equipo computacional de alta calidad para implementar "nuevos métodos".

2.2 DESCRIPCIÓN

Definamos un problema de optimización combinatoria de la siguiente forma:

$$\text{Minimizar } C(x): x \in X \qquad \textbf{(1)}.$$

La función objetivo C(x) puede ser lineal o no lineal y la condición $x \in X$ se interpreta como ciertas restricciones en la componente x para valores discretos, y X es el conjunto de vectores que normalmente se califican como viables.

Muchos procedimientos (heurísticos y óptimos) que sirven para resolver el problema de optimización combinatoria y que pueden ser escritos en la forma (1), se pueden definir de manera que involucren una secuencia de movimientos que conduzcan a una solución de prueba y error, es decir que de una solución inicial (o anterior) se pueda llegar a otra seleccionada de $x \in X$, lo que definimos entonces como el movimiento *s* que consiste en un mapeo definido de un subconjunto de X(*s*) en X..

$$s : X(s) \rightarrow X$$

Asociado con $x \in X$ esta el conjunto $S(x)$ el cual consiste de todos los movimientos $s \in S$ que "pueden" ser aplicados a x, es decir:

$$S(x) = \{ s \in S : x \in X(s)\}$$

y también

$$X(s) = \{ x \in X : s \in S(x)\},$$

El conjunto S (x) puede ser visto como una “función de vecindad”, la cual verifica todos los movimientos alrededor de una solución dada. Es importante hacer notar la importancia de los movimientos, es decir si x’ y x’’ son distintos elementos del conjunto X(s), entonces $s(x') \neq s(x'')$, es decir hay que distinguir a los movimientos que transforman una solución de prueba en otra solución de prueba diferente. Llamaremos actualización cuando se pasa de f(x) a f(x’), es decir:

$$f(x) \rightarrow f(x').$$

Una vez definido el problema de optimización combinatoria definamos el método que vamos a utilizar.

2.3 MÉTODO DE BÚSQUEDA TABÚ

La búsqueda tabú se basa en dos elementos claves: primero se restringe la búsqueda al clasificar ciertos movimientos como prohibidos (tabú), posteriormente se permite una búsqueda libre por un período corto, utilizando una función de memoria corta que provee “estrategias de olvido”, lo que permite utilizar movimientos tabú por un período corto, si este movimiento mejora cualquier otra solución, esto se conoce como "el criterio de aspiración". Esto crea el procedimiento básico conocido como “relación dual” entre las restricciones tabú y el criterio de aspiración para construir y guiar el proceso de búsqueda.

Aunado a esto se introducen funciones de memoria corta, intermedia y memoria larga, las dos últimas operan en contraposición a la función de memoria corta, estas funciones de memoria permiten utilizar los movimientos tabú más de una vez, si fuera necesario, proporcionando estrategias de olvido, lo que permite una exploración más extensa en la vecindad, en busca del mejor movimiento posible. El término corto, intermedio y largo se refieren al tiempo (iteraciones) en que un movimiento se mantiene como tabú y no puede ser utilizado.

Es conveniente para entender el método de búsqueda tabú empezar con una programación sencilla conocida como “escalando la colina” [5] la cual utiliza un proceso unidireccional desde su punto de partida hasta su óptimo local (en este ejemplo utilizaremos la aproximación en dirección descendente).

"ESCALANDO LA COLINA"

Desarrollo heurístico para:

$$\text{Minimizar } C(x): x \in X \qquad \textbf{(1)}$$

1. Seleccionamos una $x \in X$ inicial.
2. Seleccionamos algunos $s \in S(x)$ tales que

   $c(s(x)) < c(x)$.

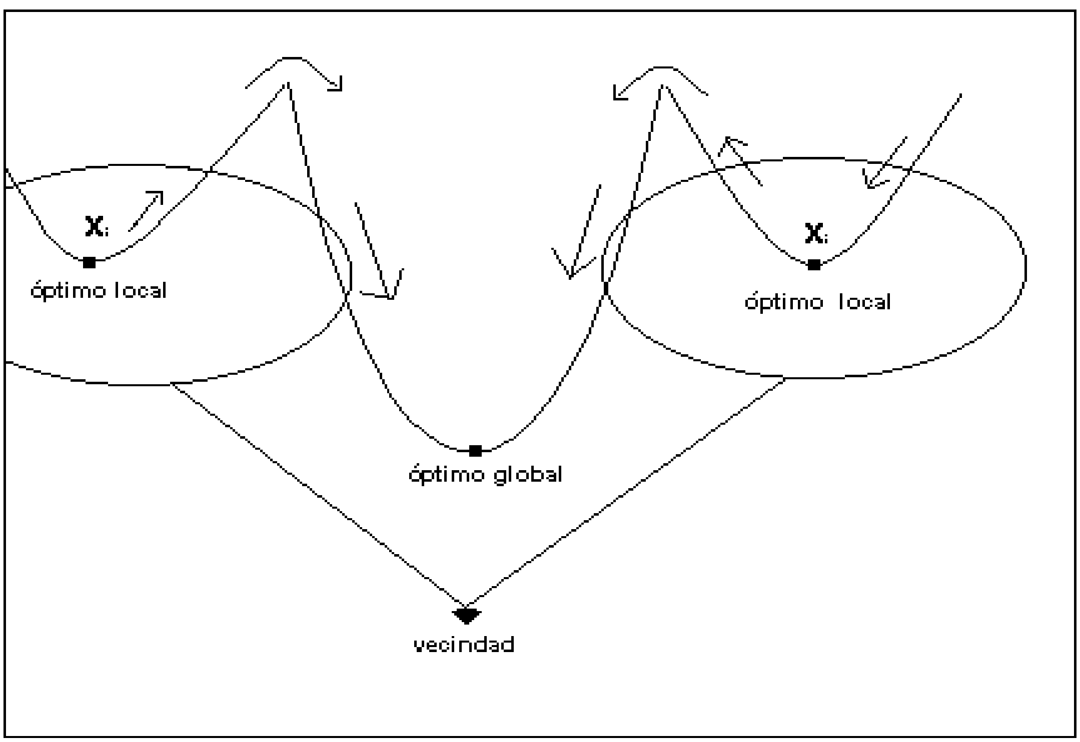
   Si s no existe, entonces x es el óptimo local y el método se detiene.

De otra forma,

3. Sea $x := s(x)$

   y regresa al paso 2.

GRAFICA 2.3.1



La principal limitación del procedimiento de "escalar la colina" es el problema combinatorio, el cuál pone al optimo local encontrado como un punto detenido (el proceso se detiene), cuando no es posible encontrar mejores movimientos, lo que no significa que, el óptimo local encontrado sea el óptimo global (ver GRAFICA 2.3.1).

La búsqueda tabú guía al procedimiento a continuar explorando sin que se confunda la ausencia de mejores movimientos, y no permite caer otra vez en el óptimo local del cual previamente emergió. De esta habilidad de incorporar y guiar el método en otros movimientos, la búsqueda tabú puede ser vista como una solución al problema combinatorio (1).

Para resolver el problema combinatorio (1), se crea un subconjunto T de S, cuyos elementos son llamados *movimientos tabúes*. Los elementos de T son determinados utilizando información histórica del proceso de búsqueda o pueden ser elegidos como movimientos prohibidos cuando se desea que una característica específica dentro del problema se cumpla, es decir los elementos de T son elegidos de una lista específica o por referencia al conjunto de condiciones tabú, entonces definimos:

$$T(\mathrm{x})=\{s \in S(\mathrm{x}): \mathrm{s(x)} \text{ viola una condición tabú}\}.$$

El procedimiento ("escalando la colina") se reescribe nuevamente como sigue:

"BUSQUEDA TABÚ SIMPLE"

1. Seleccionamos una $x \in X$ inicial y sea $x^*= x$. El conjunto de iteraciones contadas a partir de k=0. Comienza con T vacío.
2. Si $S(x)$ -T es vacío irse al paso 4.

De otra forma hacemos $k := k+1$ y seleccionamos $s_k \in S(x) - T$ de tal manera que

$s_k(x)=$ ÓPTIMO$(s(x) : s \in S(x) - T)$.

3. Sea $x := s_k(x)$. Si $c(x) < c(x^*)$, donde x^* denota la mejor solución encontrada hasta el momento, sea $x^* := x$.

4. Si un número determinado de iteraciones ha transcurrido en su totalidad o desde que x* fue la ultima mejoría obtenida; o si $S(x) - T = \varnothing$ se obtuvo directamente del paso 2, el proceso se detiene.
   De otra forma, se actualiza *T* y se regresa al paso 2.

Tres aspectos de este método merecen énfasis:

I. El uso de *T* provee de una "búsqueda restringida" y por consiguiente la solución generada depende específicamente de la composición de T y del paso 4.
II. El método no hace referencia a la condición del local óptimo, excepto implícitamente donde el local óptimo mejora a la mejor solución previamente encontrada.
III. El mejor movimiento es elegido en cada paso empleando el criterio de quedarse con la función ÓPTIMA.

La manera indicada para formar el conjunto tabú es basarse en la probabilidad de que la búsqueda se haga cíclica, es decir de seguir una secuencia de movimientos que regresen a la solución anteriormente visitada, la cual está inversamente relacionada con la distancia de la solución de prueba disponible x, de la solución previa. Si la distancia es medida en términos del número de movimientos que toma desde que la solución previa fue visitada, se conviene que no intervengan movimientos permitidos que regresen a su predecesor, entonces *T* es designado como contador cíclico de acuerdo a lo que se asumió.

La meta máxima es evitar regresar a la solución previa, para prevenir la elección de movimientos que representen la reversa de cualquier movimiento tomado durante una secuencia de t-iteraciones, el procedimiento se mueve progresivamente hacia afuera de la previa t-iteración (ver GRAFICA 2.3.1). Un descubrimiento empírico de la aplicación del método de búsqueda tabú es que *T* tiene un rango altamente estable de valores que previenen el reciclaje y permiten notablemente buenas soluciones [6].

Un elemento importante del método de búsqueda tabú es la incorporación de la función de aspiración de equilibrio $A(s, \text{x})$, cuyos valores dependen de los movimientos específicos de s y/o del vector x. Se dice que la función de aspiración es alcanzada sí:

$$c(s(\text{x})) < A(s, \text{x}).$$

El papel de $A(s, \text{x})$ es el de agregar flexibilidad, al poder utilizar movimientos que son tabú si el criterio de aspiración es alcanzado. La meta es hacer esto de manera que se conserve la habilidad para evitar el reciclaje.

Para explicar como lograr esta meta, nos referiremos a un movimiento en un segundo sentido: como un ejemplo particular del mapeo (es decir, de x en $s(\text{x})$) cuya identidad depende tanto de x como de s, que llamaremos “movimiento de solución específica”.

Existen 3 estrategias de equilibrio para evitar el reciclaje, las cuales implican prevenir el movimiento de la solución específica de x a $s(\text{x})$ si:

1) $s(\text{x})$ ha sido visitada anteriormente,
2) Si el movimiento s ha sido aplicado a x anteriormente y
3) Si el movimiento s^{-1} ha sido aplicado a $s(\text{x})$ anteriormente.

Aunque 1) es el único criterio para evitar que un movimiento se cicle, cuando el proceso revisa si el estado tabú de un movimiento puede ser anulado basándose en 1), generalmente requiere más memoria y un gran esfuerzo.

Si un movimiento tabú es permitido de manera que solo la condición 2) falte, entonces es posible que un movimiento regrese tan pronto como fue realizado, regresando a la solución anteriormente visitada (algunos experimentos han demostrado que la lista tabú designada de ésta manera, para prevenir un rango de repeticiones y movimientos en reversa no trabaja bien [4]).

La condición 3) sin embargo, es compatible con la estructura de lista tabú, usando la parte 3) para evitar el reciclaje y permitiendo movimientos tabú que vayan de x a *s*(x), aunque la solución específica que proviene del movimiento *s*(x) a x, ya haya ocurrido anteriormente.

El intento para prevenir el regreso a una solución generada anteriormente será más efectivo si agregamos la condición 3), que confiando solamente en la condición 2). Así, agregando las condiciones 2) y 3) fortalecemos la condición 1).

Las restricciones tabú y el criterio de aspiración (función de aspiración) juegan un papel doble, restringiendo y guiando el proceso de búsqueda, las restricciones tabú no permiten a un movimiento ser considerado admisible si éste no se aplica, mientras que el criterio de aspiración permite a un movimiento ser considerado admisible si éste se aplica (es decir sí satisface la condición de ÓPTIMO). Esta complementariedad de las restricciones tabú y del criterio de aspiración facilita la integración de una estructura común.

En suma, el criterio de aspiración y las restricciones tabú pueden sujetarse a una estructura organizacional común de trabajo y pueden ser vistas como aspectos diferentes del mismo principio conceptual. En un extremo como aspiración *A*(x) que es un conjunto pequeño que corresponde a un posible valor *c*(x) de un estatus tabú previo, y por otro lado *T*(x) como el conjunto de restricciones tabú.

El motivo por el que remarcamos esta forma de integración del criterio de aspiración y las restricciones tabú es la hipótesis [6] de que diferentes atributos de movimientos pueden ocasionar diferencias significativas en la calidad de las soluciones generadas y esto puede estar sujeto a diferentes movilidades del estado tabú, es decir, pueden ser recordados en la lista tabú y estar gobernados por diferentes niveles de aspiración, los cuales pueden ser archivados por el mismo mecanismo general utilizado.

Por último tenemos las memorias de término intermedio y largo que permiten al método de búsqueda tabú integrar las estrategias de intensificación y diversificación de manera efectiva. La memoria de termino intermedio opera para registrar y comparar las características de las mejores soluciones generadas durante un periodo particular de la búsqueda, éstas características se toman como un distintivo regional, de tal forma que el método busque nuevas soluciones que cumplan con ese distintivo (Ver GRAFICA 2.3.1). La función de memoria de término largo en cambio, emplea principios que son inversos a los de la función de término intermedio, esta memoria se utiliza para producir un punto inicial de la búsqueda en una nueva región mediante la penalización de las características que la memoria de término intermedio tiene como distintivo, es decir las convierte en tabú, lo que permite al método ir en busca de otras características distintas de las que prevalecen en la región actual de búsqueda, esto da una mayor diversificación en la búsqueda (Ver GRAFICA 2.3.1).

2.4 EJEMPLO

Se tomó como ejemplo el problema de las N - Reinas [4] del juego de ajedrez para ejemplificar el método de búsqueda tabú.

El problema de las N - Reinas consiste en colocar n- reinas en un tablero de ajedrez de n x n, de tal forma que, dos reinas no puedan "comerse" una a la otra. Para que esto no suceda dos reinas no pueden ocupar la misma línea: horizontal, vertical o diagonal. Sí esto ocurre entonces se dice que existe una *colisión*.

El problema de optimización consiste entonces, en encontrar un arreglo que minimice el número total de colisiones en el tablero de ajedrez de n x n. La solución ÓPTIMA será aquella que produzca cero colisiones.

Sí solo se utilizan dos reinas en el tablero de ajedrez, entonces se pueden encontrar varios arreglos que generen la solución ÓPTIMA., por lo que el problema será entonces: "maximizar el número de reinas que pueden ser colocadas en el tablero de ajedrez de n x n", sujetas a la restricción anterior (que no ocupen la misma línea horizontal, vertical o diagonal en el tablero de ajedrez).

Este problema puede ser modelado como un problema de programación entera de tipo 0-1, sin embargo se planteó como un problema de permutación.

2.4.1 Definición

Sean las n - reinas indexadas por las letras (i = 1,2,..., n), de tal forma que la reina i tomará el i *-esimo* renglón. Y se define a π (i) como la columna donde la reina i está colocada, entonces se tiene la siguiente permutación:

$$\Pi = \{\pi (1), \pi (2),..., \pi (n)\}$$

La cual especifica la posición de las n- reinas en el tablero de ajedrez (Ver FIGURA 2.4.1 Y 2.4.2), esta presentación garantiza que dos reinas no se puedan "atacar" por estar en el mismo renglón o columna, de ésta manera, el problema se reduce a minimizar el número de colisiones en las diagonales.

En la FIGURA 2.4.1, se presenta un tablero de ajedrez de 4 x 4 que corresponde a la permutación Π = {3,4,2,1}, donde se presentan dos colisiones: las reinas 1 y 2 y las reinas 3 y 4, ya que éstas se pueden atacar.

FIGURA 2.4.1.

	1	2	3	4
1			Q	
2				Q
3		Q		
4	Q			

Una información importante en la solución de este problema, es el número de reinas que se encuentran en la misma diagonal en un arreglo. Ya que el tablero de ajedrez de tamaño n x n, tiene 2n -1 diagonales negativos y 2n -1 diagonales positivos, dos arreglos (denotados por d_1 y d_2) de tamaño 2n -1 serán suficientes para almacenar la información.

Para identificar cada diagonal se verá que: la suma de un renglón indexado y una columna indexada es constante en cualquier diagonal negativa, y la diferencia en esos índices es constante en cualquier diagonal positiva. Por ejemplo, las reinas 1 y 2 están colocadas en la diagonal positiva indexado como -2 ya que $1- \pi(1) = 2 - \pi(2) = -2$. Es decir, si a la reina indexada como 1 se le resta la columna donde esta colocada (indexada como $\pi(1)=3$) produce el mismo resultado que si, a la reina indexada como 2 se le resta la columna donde esta colocada.

De igual forma, se puede ver que las reinas 3 y 4 están colocadas en la diagonal negativa indexado como 5 (i.e. $3+ \pi(3)= 4 + \pi(4) =5$). Otra información importante es que los índices de las diagonales positivas tienen un rango que va de [1-n, n-1], a diferencia del rango para las diagonales negativas que es el intervalo cerrado [2, 2n].

2.4.2 LA BÚSQUEDA TABÚ.

Para este ejemplo se utilizó un tablero 7 x 7, con 7 reinas. Se da una solución inicial aleatoria, como lo muestra la FIGURA 2.4.1, para comenzar.

FIGURA 2.4.2.1: Permutación inicial.

INDICES DE LAS COLUMNAS

4	5	3	6	7	1	2

Este orden especifica que existe una reina en el renglón 1 columna 4, otra en el renglón 2 columna 5, renglón 3 columna 3, renglón 4 columna 6, etc. El resultado de esta configuración muestra cuatro colisiones, las reinas (1,2), (4,5), (6,7) y (2,6), ya que se encuentran en la misma diagonal (ver FIGURA 2.4.2).

FIGURA 2.4.2.2: Posición de las reinas en el tablero.

		Q				
			Q			
	Q					
					Q	
						Q
Q						
	Q					

El método de búsqueda tabú, funciona bajo el precepto de una vecindad, la cuál se construye para identificar soluciones alternas que pudieran ser rechazadas por estar en alguna solución obtenida.

Cada intercambio en el orden de las reinas se identificara como un *movimiento*, los cuales son usados frecuentemente en los problemas de permutación, para definir a la vecindad.

De ésta manera un movimiento llevará de una solución a otra. En este problema (con 7 reinas) la vecindad está compuesta de 21 soluciones alternas (combinaciones de 7 en 2) que representan a todos los movimientos posibles.

Asociado a cada movimiento se tiene un valor que representa el cambio en la función objetivo (i.e. el total de colisiones) como resultado del movimiento ejecutado. El valor del movimiento provee bases fundamentales para evaluar la calidad del movimiento (i.e., sí mejoró, empeoró o dejó sin cambio a la función objetivo), sin embargo existen otros criterios que también son importantes.

Un punto importante en el Método de Búsqueda Tabú es la obtención del subconjunto de movimientos de una vecindad que se consideran como prohibidos (tabú), esta clasificación dependerá de la historia de los movimientos, es decir aquella permuta (cambio) entre dos reinas que generó soluciones pasadas.

FIGURA 2.4.2.3: Movimiento de las reinas 2 y 6.

Permutación inicial	4	5	3	6	7	1	2
Movimiento	4	1	3	6	7	5	2

Para evitar que este movimiento se ejecute en forma inversa, es decir, que se regrese a la posición anterior y por consiguiente no se avance en la búsqueda; o que al estar ejecutando estos dos movimientos indefinidamente se caiga en un ciclo, se debe clasificar este movimiento como **tabú**, lo que significa que este par de reinas (2,6) = (6,2) permanecerá sin movimiento un tiempo determinado. Se eligió 3 iteraciones, para este ejemplo.

FIGURA 2.4.2.4: Estructura tabú, para el movimiento de las reinas (2,6)

	2	3	4	5	6	7
1						
3					3 iteraciones	
3						
4						
5						
6						
7						

Cada celda en la FIGURA 2.4.2.4 contiene el número de iteraciones que permanecerán sin movimiento las reinas que están involucradas, lo que se describió como *función de memoria*, al principio de este Capítulo (sección 2.3).

Como se mencionó anteriormente, las restricciones tabú pueden ser utilizadas bajo ciertas circunstancias, como por ejemplo: cuando un movimiento tabú resulta ser la mejor solución, comparada con los demás movimientos efectuados. En este caso, la clasificación tabú deberá ser eliminada. La condición que permite que un movimiento tabú deje de serlo se conoce como el *criterio de aspiración.*

A continuación se presentan 5 iteraciones en las cuales se utilizó, tanto la *función de memoria* como el *criterio de aspiración*.

La solución inicial tiene un total de cuatro colisiones, la estructura tabú se inicializa en cero. Después de evaluar algunos movimientos, se consideraron los 5 mejores en términos de las colisiones que se presentaron.

Iteración 0 (Punto de partida).

Solución inicial

4	5	3	6	7	1	2

Estructura tabú

	2	3	4	5	6	7
1						
2						
3						
4						
5						
6						

Movimiento		Valor
1	7	-2
2	4	-2
2	6	-2
5	6	-2
1	5	-1

De los 5 movimientos, cuatro reducen en dos las colisiones, por lo que se puede elegir cualquiera de ellos, se selecciona el primer movimiento (1,7).

Iteración 1

Solución inicial

2	5	3	6	7	1	4

Estructura tabú

	2	3	4	5	6	7
1						3
2						
3						
4						
5						
6						

Movimiento		Valor
2	4	-1
1	6	0
2	5	0
1	2	+1
1	3	+1

Esta nueva solución contiene dos colisiones (se redujo en dos el número de colisiones con el movimiento efectuado). La estructura tabú muestra que el movimiento en el par de reinas (1,7) esta prohibido por las próximas 3 iteraciones.

De los siguientes 5 movimientos uno disminuye en una unidad el número de colisiones, dos movimientos no generan cambio en el número de colisiones y los otros dos aumentan en una unidad el número de colisiones. Así que, se selecciona el movimiento que mejora el paso anterior, las reinas (2,4).

Iteración 2

Solución inicial

2	6	3	5	7	1	4

Estructura tabú

	2	3	4	5	6	7
1						2
2			3			
3						
4						
5						
6						

	Movimiento		Valor
	1	3	0
T	1	7	1
T	2	4	1
	4	5	1
	6	7	1

Ahora se tiene una colisión con las reinas 1 y 4.

Se puede ver que la condición tabú de las reinas (1,7) bajó en una unidad, puesto que ya efectuamos una iteración. Ahora las reinas (2,4) son las que tienen prohibido modificarse por las próximas tres iteraciones, entonces estos dos pares de reinas tienen condiciones tabú, por lo que no podemos modificarlas.

De los movimientos que quedan, uno no genera cambio y los otros dos aumentan en una unidad el número de colisiones, por lo que se selecciona el movimiento que tiene menor valor (1,3).

Iteración 3

Solución inicial

3	6	2	5	7	1	4

Estructura tabú

	2	3	4	5	6	7
1		3				1
2			2			
3						
4						
5						
6						

	Movimiento		Valor
T	1	3	0
T	1	7	0
	5	7	1
	6	7	1
	1	2	2

En este paso se continúa con una colisión ya que el movimiento anterior (cuyo valor era cero) no mejoró el número de colisiones (la función objetivo). Esta iteración tiene 3 movimientos tabú con diferente valor en la *función de memoria* (estructura tabú) la pareja (1,3) con valor 3, la pareja (2,4) con valor 2 y la pareja (1,7) con valor 1.

De los 5 movimientos posibles dos no modifican el número de colisiones, dos aumentan en uno el número de colisiones y un último movimiento aumenta en dos el número de colisiones. Los dos primeros movimientos tienen clasificación tabú, y como no mejoran el número de colisiones (aplicando el *criterio de aspiración*), no se pueden seleccionar, por lo que se selecciona el primer movimiento entre los dos que tienen valor uno, que es (5, 7).

Iteración 4

Solución inicial

3	6	2	5	4	1	7

Estructura tabú

	2	3	4	5	6	7
1		2				
2			1			
3						
4						
5						3
6						

	Movimiento		Valor
	4	7	-1
T	5	7	-1
	1	5	0
	2	5	0
T	2	4	2

En esta iteración se tienen dos colisiones, tres movimientos tabú y dos movimientos para ser seleccionados que mejoran el número de colisiones, pero como uno de ellos es denotado como tabú se selecciona el otro (4,7).

Iteración 5

Solución inicial

3	6	2	7	4	1	5

Estructura tabú

	2	3	4	5	6	7
1		1				
2						
3						
4						3
5						2
6						

	Movimiento		Valor
T	1	3	-1
	5	6	-1
T	5	7	0
	1	6	0
	1	7	2

En esta iteración se tiene una colisión, tres movimientos tabú y cinco posibles movimientos, dos de los cuales reducen el número de colisiones. Si se selecciona el primer movimiento de los dos posibles que mejoran el número de colisiones, se estaría seleccionando un movimiento tabú, lo que significa regresar a un movimiento anterior, sin embargo, como este movimiento mejora la función objetivo (i.e. el número de colisiones), y más aún proporciona la mejor solución posible (función objetivo) con cero colisiones, entonces hacemos uso del criterio de aspiración y se selecciona este movimiento (1,3).

Este movimiento produce la mejor solución posible (ÓPTIMO GLOBAL) por lo que, el procedimiento de búsqueda tabú termina.

2.4.3 Solución final

Posición de las reinas						
2	6	3	7	4	1	5

	Q					
					Q	
		Q				
						Q
			Q			
Q						
				Q		

Aquí se tienen una solución al problema de las 7 reinas en el tablero de ajedrez de 7x7, en forma tal que se tienen cero colisiones; es decir se alcanzó la función objetivo al obtenerse la solución ÓPTIMA.

CAPITULO III

MODELACIÓN CON EL MÉTODO DE BÚSQUEDA TABÚ PARA EL PROBLEMA DE PROGRAMACIÓN DE HORARIOS

En este capítulo se describe el modelo de búsqueda tabú aplicado a la programación de horarios (*el problema de la programación de horarios*), después se describen los requerimientos y limitaciones que se necesitan para poder obtener un resultado óptimo (*requerimientos y limitaciones*) y por último se da la formulación matemática descrita en pseudocódigo del programa que se realizó (*formulación matemática*).

3.1 EL PROBLEMA DE LA PROGRAMACIÓN DE HORARIOS

Ahora describiremos el método de búsqueda tabú para el problema específico de la programación de horarios, para lo cuál definiremos nuestro problema de la siguiente manera:

Sea X_{ijk} una variable entera la cual toma el valor uno si se programa al maestro i en el horario j y se le asigna la materia k y toma el valor cero en cualquier otro caso.

Definiremos S como el conjunto de materias que se le programaron al maestro i en los diferentes horarios que tenga disponibles, entonces tendríamos que:

$$S \leq \sum X_{irk} \quad \text{para } j = 1,2,\ldots\ n.$$

Definimos C(S(x)) como la función que genera la información de a qué maestros se le asigna qué materia y cuantas veces, por lo que:

Mínima carga académica del maestro $\leq$ C(S(x)) $\leq$ *Máxima carga académica del maestro*.

Dado que la variable X_{ijk} toma el valor *uno* si se programa al maestro i en el horario j y se le asigna la materia k, entonces tendríamos que:

$$\sum X_{rjk} = 1 \text{ ó } \sum X_{rjk} = 0$$

Y dado que S representa el número de cursos programados para el maestro i, en el horario j, tendríamos:

$$S \leq \sum X_{irj} \qquad \text{para } r=0, 1,2,... \text{ máxima carga académica.}$$

3.1.1 Función objetivo

El cálculo de la función objetivo esta formado por varias componentes, las cuales se desean minimizar, estas son:

- *El componente académico*: se tiene un costo C_{ik} de asignar al maestro i en la materia k, en este caso mientras más deseable sea asignar a un maestro una materia, menor debe ser C_{ik}.
- *La componente de horarios*: una vez establecida una programación se puede medir lo compacto de ésta, midiendo el total de horas libres entre cursos para cada maestro.
- *La componente de cursos distintos*: se puede medir para cada maestro las materias distintas asignadas.

3.1.2 Movimientos posibles

Los movimientos posibles asignados en este problema son:

1. <u>Subir o bajar el valor de una variable (cero o uno):</u> Para subir el valor de una variable se requiere que la suma de variables en ese grupo sea *cero* y todas las variables del grupo pueden subir.
2. <u>Subir una variable y bajar otra</u>. Para bajar el valor de una variable se requiere que la suma del grupo sea *uno* y en ese grupo solo una variable pueda bajar.

Para los movimientos de las variables donde se utilizan dos variables, una que sube y otra baja se requiere que:

1. Si ambas variables están en el mismo grupo, entonces la variable que baja deberá ser la que tiene asignado el número uno en el grupo.

2. Si ambas variables están en grupos distintos se requiere entonces que la variable que baja esté localizada en el grupo cuya suma sea uno y la variable que sube puede localizarse en cualquier grupo cuya suma sea cero, esto asegura que la suma de un grupo sea cero y la suma en el otro sea uno.

El procedimiento se llevaría a cabo de la siguiente manera: Sobre todos los grupos se revisa la suma de las variables. Si ésta vale uno se busca en las variables del grupo, cual variable vale 1 y se prueba la disminución de su valor, haciendo la variable 0. Si la suma es cero se prueba el incremento de todas las variables (una por una) en el grupo, eligiendo por último la que genera "la mejor solución". Cada uno de los movimientos realizados es almacenado en la lista T(x) tabú de tal manera que el movimiento anterior no podrá realizarse otra vez a menos que cumpla con el criterio de aspiración (es decir que genera la mejor solución encontrada en el grupo).

Para los movimientos de dos variables se procede en forma similar: Se comparan las sumas de dos grupos: si sus sumas son distintas, se busca en el grupo cuya suma sea uno la variable que vale uno, para disminuir su valor a cero. Y de forma similar a la anterior, se prueba aumentando el valor de cada una de las variables del otro grupo. Si el grupo es el mismo y la suma es cero, no se hace nada. Si la suma es 1 se procede en forma similar. Cada vez que se genere un movimiento la lista tabú se ira incrementando.

La lista tabú se implementa guardando dos valores, el primero indica las variables que suben su valor y el segundo las que bajan, cuando el movimiento es de una sola variable se puede representar como (0,1) si su valor baja ó (1,0) si su valor sube.

3.2 REQUERIMIENTOS Y LIMITACIONES

Los aspectos que se consideraron en la programación son:

1. Número de cursos a programar
2. Total de salones disponibles
3. Total de cursos para asignar a cada maestro
4. Cursos preestablecidos a ciertos maestros

5. Los horarios preestablecidos a ciertas materias
6. Programación de un número determinado de materias en el horario matutino y otro número determinado en el horario vespertino, y
7. La compactación de horarios

Como se puede ver todos estos aspectos generan un modelo grande respecto al número de variables y de restricciones que se manejan, sin embargo un acomodo adecuado de las estructuras de datos nos simplificó su implementación. Además el número de movimientos posibles se disminuye en gran medida por las restricciones que se establecen para los grupos que obligan a que cada maestro que no se le pueda asignar dos materias en el mismo horario es decir que, $\sum X_{rjk} = 1$, es el máximo valor que puede tomar.

3.3 FORMULACIÓN MATEMÁTICA (pseudocódigo)

INTRODUCIR DATOS ← BASE DE DATOS INICIAL

INICIA {

PASO 1. Formar los grupos de cada maestro

PASO 2. Seleccionar un X_{ijk} en el primer grupo
Si $X_{ijk}=0$ entonces sube, de otra forma baja

PASO 3. Guardar el movimiento en la lista tabú.

PASO 4. Verificar si $\sum X_{rjk} = 1$ ó $\sum X_{rjk} = 0$

Si $\sum X_{rjk} = 1$ continuar con el siguiente grupo

Si $\sum X_{rjk} = 0$ regresar al PASO 2

PASO 5. Verificar si los cursos asignados al maestro i se encuentran en el rango

$$L_1(i) \le \sum_{i,k} X_{rjk} \le L_1(i)$$

Si se cumple, continuar con el siguiente PASO
Si no, regresar al PASO 2

PASO 6. Medir la compactación de horarios del maestro i en el horario j

$$\sum_{j} X_{rjk} \text{ para los j's donde } X_{rjk} = 0$$

Si $\sum X_{rjk}$ es mínima:
i) catalogar este arreglo como ÓPTIMO
ii) hacer los demás movimientos tabú
iii) continuar con el siguiente PASO

PASO 7. Medir el costo académico de los ÓPTIMOS del paso anterior

$$\sum_{i,j,k} C_{ijk} X_{rjk}$$

Hacer ÓPTIMO el arreglo que tenga el menor costo.

PASO 8. Verificar si el número de salones disponibles en el horario j anda en el rango

$$L_2(j) \leq \sum_{j,k} X_{rjk} \leq L_2(j)$$

Si no se cumple regresar al PASO 2
Si se cumple continuar con el siguiente PASO

PASO 9. Verificar si en el horario j se programaron $C_2(j)$ cursos de la materia k

$$\sum_{i} X_{rjk} \geq C_2(j)$$

Si no se cumple regresar al PASO 2
Si se cumple continuar con el siguiente PASO

PASO 10. Verificar si para la materia k se programaron los cursos $C_3(k)$ en la mañana y $C_4(k)$ en la tarde, tomando en cuenta que L es el último horario de la mañana.

$$\sum_{i} X_{rjk} = C_3(j) \qquad \text{con } j \leq L$$

$$\sum_{i} X_{rjk} = C_4(j) \qquad \text{con } j \geq L$$

Si no se cumple regresar al PASO 2.

Si se cumple catalogar este arreglo como la MEJOR SOLUCIÓN

TERMINA.

3.4 ALGORITMO DEL MÉTODO DE BÚSQUEDA TABÚ PARA LA PROGRAMACIÓN DE HORARIOS (PSEUDOCÓDIGO)

```
INICIO

Leer ← (salones_disponibles, maestros_disponibles, cursos_a_programar, horario_mat,
horario_ves, min_carga_maestro, max_carga_maestro, costo_académico, long_tabu,
solucion_inicial );
```

```
/* Se calculan los vectores de principio y fin de cada grupo */

SUBVOID principio_final (cuantos_en_grupo, principio_grupo, final_grupo, numero_grupos)
INICIO
i = entero;
principio_grupo ← 0;
final_grupo = cuantos_en_grupo -1;
FOR     i = 1, numero_grupos -1;
        principio_grupo(i) = final_grupo (i-1) +1;
        final_grupo(i) = principio_grupo(i) + cuantos_en_grupo(i) -1;
FIN
```

```
/* Se crea el vector de horarios para el maestro */

SUBVOID crea_horario (maestro, vector, ini_grupo_maestro, fin_grupo_maestro, suma_grupos,
principio_grupo, solucion, tipo_horario)
INICIO
i = entero;
FOR     i = 0, n;
        Vector ← 0;
FOR     i = ini_grupo_maestro, fin_grupo_maestro;
        IF      suma_grupos (i) = 1;
                Vector (i) ← 1;
        ENDIF
FIN
```

Nota: **n** *toma los siguientes valores 14 para el área de Contabilidad y Administración, 13 para el área de Economía y 12 para las demás áreas.*

```
/* Se mide lo compacto de los horarios */

SUBINT mide_compacto (vector)
INICIO
inicio, final, horas = entero;
i, flag, suma = entero;

        /* horario matutino */

inicio ← 0;
final ← 0;
flag ← 0;
suma ← 0;
FOR     i = 0,5
        IF    vector (i) =1 AND flag =1;
              final = i;
              horas = horas + 1;
        ENDIF
CONTINUE;
        IF    vector (i) =1 AND flag =0
              inicio =i;
              final = i;
              flag =1;
              horas =1;
        ENDIF
        IF    flag <> 0
              Suma = final- inicio +1 -horas;
        ENDIF

        /*horario vespertino*/
```

```
incio,final ← 6;
flag ← 0;
FOR     i = 6,11
        IF vector (i) =1  AND flag =1
            final = i,
            horas = horas +1;
        ENDIF
CONTINUE;
        IF    vector (i) =1 AND flag =0
            inicio = i;
            final = i;
            flag = 1;
            horas = 1;
        ENDIF
        IF    flag < > 0
            Suma = suma +final -inicio +1 -horas;
        ENDIF
RETURN suma;
FIN
```

/* Se mide lo compacto del horario para el maestro */

```
SUBINT compacto_maestro (maestro, ini_grupo_maestro, fin_grupo_maestro, suma_grupos,
principio_grupo, solucion, tipo_horario)
INICIO
i, vector (n), resultado = entero;
FOR     i= 0, n-1;
        vector ← 0;
        resultado = mide_compacto (vector);
RETURN resultado;
FIN
```

/* Se calcula el vector para las materias que imparten los maestros */

```
SUBINT cursos_programados (maestro, ini_grupo_maestro, fin_grupo_maestro, suma_grupos,
principio_grupo, solucion)
INICIO
i = entero;
FOR     i = 0, m;
        vector (i) =0;
FOR     i = ini_grupo_maestro, fin_grupo_maestro;
        IF    suma_grupos (i) =1;
              Vector (i) ← 1;
        ENDIF
FIN
```

***m** toma los valores 5 para Sociología y Adm. Pública, 2 para Derecho, 5 para Químico Biólogo, 2 para Psic. Y C. de la Com. 4 para Economía, 24 para Matemáticas, 6 para Ing. Ind. Y de Sistemas, 11 para Ciencias e Ingeniería y 4 para Contabilidad y Administración.*

```
/* Se calcula el costo académico de la programación */

SUBINT costo(numero-grupos, principio_grupo, suma_grupo, solucion, costo)
INICIO
i, suma = entero;
FOR     i = 0, numero_grupos -1;
        IF    suma_grupos (i) = 1
              suma = suma + costo;
        ENDIF
RETURN suma;
FIN
```

```
/* Se calcula el vector de error y el valor de la función objetivo */

SUBVOID actualiza(sube, baja, error, nuevo_error, parametros, tipo_horario, tipo_curso,
tipo_maestro, salones utilizados, dept_salones_utilizados, cursos_programados,
dept_cursos_programados, min_carga_maestro, max_carga_maestro, carga_maestro, costo,
nueva_funcion, ini_grupo_maestro, fin_grupo_maestro, suma_grupos, principio_grupo, solucion)
```

```
INICIO
i, e0, e1, e2, e3, e4, e5, e6, vector[n] = entero;
n11, n12, n13, n14;
n11 ← -1;
n12 ← -1;
n13 ← -1;
n14 ← -1;
FOR    i=0, 5
       nuevo_error (i) = error (i);
       IF    sube > -1
             IF    parametros (0) <> 0
                   n11 = tipo_horario (sube);
                   e0 = 0;
             ENDIF

                   IF      dept_salones_utilizados (n11) < salones_utilizados (n11)
                           e0 = salones_utilizados(n11) -dept_salones_utilizados (n11);
                           e1 = 0;
                   ENDIF

                   IF      dept_salones_utilizados (n11) < salones_utilizados (n11) +1
                           e1 = salones_utilizados(n11) +1 -dept_salones_utilizados (n11);
                   nuevo_error (0) = nuevo_error (0) + e1 -e0;
                   ENDIF

             IF    parametros (1) <> 0
                   n12 = tipo_curso (sube);
                   e0=0;
             ENDIF

                   IF      ABS [cursos_programados (n12) - dept_cursos_programados (n12)] >0;
                           e0 = ABS [cursos_programados (n12) - dept_cursos_programados (n12)];
                   e1 = 0;
                   ENDIF
```

```
        IF      ABS [cursos_programados (n12) - dept_cursos_programados (n12)] <0;
                e1= ABS [cursos_programados (n12) - dept_cursos_programados (n12)];
        nuevo_error (1) = nuevo_error (1) + e1 - e0;
        ENDIF

IF      parametros (2) <> 0
        n13 = tipo_maestro (sube);
        e0=0;
ENDIF

        IF      carga_maestro (n13) > max_carga_maestro (n13)
                e0 = carga_maestro (n13) - max_carga_maestro (n13);
        ENDIF

        IF      carga_maestro (n13) < min_carga_maestro (n13)
                e0 = min_carga_maestro (n13)  -carga_maestro (n13);
        e1= 0;
        ENDIF

        IF      carga_maestro (n13) +1 > max_carga_maestro (n13)
                e1=carga_maestro (n13) +1 -max_carga_maestro (n13);
        ENDIF

        IF      carga_maestro (n13) +1 < min_carga_maestro (n13)
                e1=min_carga_maestro (n13)  -carga_maestro (n13);

        nuevo_error (2) = nuevo_error (2) +e1 -e0;
        ENDIF

IF      parametros (3) <> 0
        nuevo_error (3) = nuevo_error (3) + costo (sube);
ENDIF
```

```
        IF    parametros (4) <> 0
              nuevo_error (4) =nuevo_error (4) + mide_compacto (sube);
        ENDIF

        IF    parametros (5) <> 0
              nuevo_error (5) =nuevo_error (5) + compacto_maestro (sube);
        ENDIF

solucion= mejor_solucion (vector);

return ( solucion);

FIN
```

CAPITULO IV

IMPLEMENTACIÓN DEL PROGRAMA

En este capítulo se aplica el método descrito en el capítulo anterior con datos reales obtenidos del Departamento de Matemáticas de la Universidad de Sonora, donde se programan alrededor de 250 maestros cada semestre en varias áreas de la Universidad (4.1), después se describen los aspectos generales de la codificación que dan lugar a los datos de entrada del programa (4.2), se describen los resultados obtenidos (4.3) y por último se comparan estos resultados con programaciones anteriores que ha realizado el Departamento de Matemáticas (4.4).

4.1 UN EJEMPLO ESPECÍFICO

Para probar la modelación del método de búsqueda tabú, se eligió el departamento de Matemáticas de la Universidad de Sonora, por varias razones. Primero, porque es el departamento que más grupos atiende (alrededor de 460 por semestre), con un total de 150 profesores disponibles, lo que permite probar el programa con un conjunto grande de variables. Segundo, por la facilidad que se tiene para obtener la información y tercero, para poder comparar la programación realizada con el método de "búsqueda Tabú", con la programación manual que hace el departamento.

Para iniciar la programación, se pidió al departamento la información que ellos recopilan de los profesores al inicio de cada semestre, esta información es vaciada en una hoja (ver FIGURA 4.1.1) por los propios profesores, la cual se conoce como "*hoja personal*" (ver FIGURA 4.1.1). Esta hoja personal contempla lo siguiente:

- Nombre
- Licenciatura que estudiaron
- Maestría (sí tiene)
- Dirección
- Teléfono
- Información de las materias que imparte (cuáles son)

- Horarios disponibles (cada maestro elige a su discreción)
- Observaciones (si desean sobre carga o disminución, etc.).

Con estas hojas, se tiene la información requerida por el programa. Cada una de éstas hojas se codificó y se capturó (como se verá en la sección 4.2).

Para elaborar la programación se codificaron un total de 171 profesores adscritos al departamento de Matemáticas. Cada profesor tiene para elegir 35 materias diferentes, de las cuales 11 se imparten en la Lic. de Matemáticas y las demás en las carreras a las cuales el departamento de Matemáticas les da servicio (ver TABLA 4.1.1), como son: Ingeniería, Sociología, Economía, Psicología, Arquitectura, Ciencias de la Comunicación, Derecho, Químico Biológicas, etc.

Estas carreras están agrupadas por áreas de Servicio, mismas que seguimos en la programación, éstas son:

TABLA 4.1.1

AREA	No. PROFESORES	MATERIAS	No. CURSOS
CIENCIAS- INGENIERÍA	50	11	103
SOCIOLOGIA Y ADM.	11	5	14
CS. DE LA COMUNICACIÓN	7	2	10
CONTABILIDAD Y ADM.	41	3	75
ING. IND. Y DE SISTEMAS	19	5	32
DERECHO	10	2	18
QUÍMICO BIÓLOGO	10	5	15
ECONOMÍA	5	4	10
MATEMATICAS	18	24	24

La programación se formulo por áreas ya que de esta manera lo lleva a cabo el departamento de Matemáticas, esto es porque los horarios y salones disponibles se envían al departamento de Matemáticas por áreas; el encargado de la programación elabora los horarios y los envía a los departamentos, una vez que esta terminada la programación. Además, así se pudo comparar la programación del "modelo de búsqueda tabú" con el hecho por el departamento de Matemáticas.

Es conveniente agregar que muchos profesores imparten clase en dos o más áreas, pero para efectos de la programación, cada profesor contó una vez en cada área, así que, el total de profesores programados (171) no corresponde al total de profesores del departamento.

Por último, dentro de la programación se omitió al departamento de Arquitectura, donde se imparten dos materias (álgebra y geometría), ya que por ser carrera de reciente creación solo cuenta con dos grupos.

FIGURA 4.1.1. Hoja Personal

DEPARTAMENTO DE MATEMATICAS

NOMBRE: ____________________

LICENCIATURA:__________ PASANTE:__________ TITULADO:__________

MAESTRIA:__________ PASANTE__________ TITULADO:__________

DIRECCION:__________TELEFONO__________

INFORMACION PARA EL SEMESTRE 96-2 ESPECIFICACIONES DE MATERIAS Y HORIARIOS

☐	MATEMATICAS Y ESTADISTICA APLICADA A LAS CIENCIAS SOCIALES.	☐	ANALISIS NUMERICO Y PROGRAMACION
___	Matemáticas I	___	Análisis Numérico I
___	Estadística I	___	Programación
___	Estadística II	☐	CALCULO Y ECUACIONES DIFERENCIALES
___	Estadística III	___	Cálculo Diferencial
☐	MATEMATICAS Y ESTADISTICA APLICADA A LAS CS. EC. ADMINISTRATIVAS	___	Cálculo Integral
___	Matemáticas Aplicadas I	___	Cálculo de Varias Variables
___	Matemáticas Aplicadas II	___	Ecuaciones Diferenciales
___	Matemáticas Aplicadas III	☐	LICENCIATURA EN MATEMATICAS
___	Estadística Aplicada I	___	Cálculo Diferencial e Integral IV
___	Estadística Aplicada II	___	Ecuaciones Diferenciales II
___	Estadística Aplicada III	___	Geometría Diferencial I
☐	PROBABILIDAD Y ESTADISTICA	___	Algebra Moderna I
___	Probabilidad	___	Análisis Matemático I
___	Estadística	___	Fun. de Variable Compleja I
___	Probabilidad y Estadística	___	Algebra Moderna II
☐	ALGEBRA Y GEOMETRIA	___	Análisis Matemático II
___	Algebra Superior I	___	Fun. de Variable Compleja II
___	Geometría Analítica	___	Análisis Matemático III
___	Algebra Lineal I	___	Análisis Matemático IV
		☐	OTROS

PROFESOR LLENE ESTA TABLA :

HORARIO DISPONIBLE						Materias y Horarios Asignados Semestre 96-1				
HOR.	LUN.	MAR.	MIER.	JUE.	VIER.	Clave	Grupo	Materia	Hora	L M M J V AULA
7-8										
8-9										
9-10										
10-11										
11-12										
12-13										
13-14						Materias y Horarios Asignados Semestre 96-2				
14-15						Clave	Grupo	Materia	Hora	L M M J V AULA
15-16										
16-17										
17-18										
18-19										
19-20										
20-21						OBSERVACIONES				
21-22										

4.2 ASPECTOS GENERALES DE LA CODIFICACIÓN Y PROGRAMACIÓN

La codificación se llevó a cabo separando a las áreas por tamaño, (según el número de grupos que se programan) en pequeños, medianos y grandes, de la siguiente manera:

PEQUEÑOS

Economía

Sociología y Administración Pública

Ciencias de la Comunicación

Químico Biológicas

MEDIANOS

Ingeniería Industrial y de Sistemas

Derecho

Matemáticas

GRANDES

Contabilidad y Administración

Ciencias e Ingeniería

Una vez agrupados de esta manera, se comenzó con la codificación para las áreas pequeñas, después las medianas y por último las grandes.

La codificación consistió en asignar a los maestros, horarios y cursos un número, comenzando por los maestros en el orden en que aparecen en las listas que nos facilitó el departamento de Matemáticas y comenzando siempre con el número cero, después se siguió con las materias de la misma manera y por último los horarios. Esto se resume en la FIGURA 4.2.1.

FIGURA 4.2.1 Programación de un semestre anterior para la licenciatura en Economía.

DEPARTAMENTO DE MATEMATICAS

CATALOGO DE GRUPOS

PERIODO 96-2 AREA: LIC. EN ECONOMIA

Clave Gpo. Nombre de la Materia -------------------- -------------- Maestro -------------------- Aula ---------------- Lun Mar Mie Jue Vie

0223	M01	MATEMATICAS Y ESTADISTICA	GABRIEL GONZALEZ ESQUER	10H	201	15-16	15-16	15-16	15-16
0223	M02	MATEMATICAS Y ESTADISTICA	DÓRAME BUERAS LUCIA G.	10H	305	12-13	12-13	12-13	12-13
0223	M03	MATEMATICAS Y ESTADISTICA	DÓRAME BUERAS LUCIA G.	10H	201	13-14	13-14	13-14	13-14
0228	M01	MATEM. APLI. A LA ECONOMIA I	CHACARA MONTES MARIA M.	10H	306	15-16	15-16	15-16	15-16
0228	M01	MATEM. APLI. A LA ECONOMIA I	CHACARA MONTES MARIA M.	10H	306	12-13	12-13	12-13	12-13
0238	M01	MATEM. APLI. A LA ECONOMIA II	VALENZUELA SASTRE MARIA	10H	305	15-16	15-16	15-16	15-16
0238	M01	MATEM. APLI. A LA ECONOMIA II	ZEPEDA MILANEZ CLAUDIA A.	10H	201	11-12	11-12	11-12	11-12
0238	M01	MATEM. APLI. A LA ECONOMIA II	VALENZUELA SASTRE MARIA	10H	203	8-9	8-9	8- 9	8- 9
0238	M01	INFERENCIA ESTADISTICA	DÓRAME BUERAS LUCIA G.	10H	202	15-16	15-16	15-16	15-16
0238	M01	INFERENCIA ESTADISTICA	DÓRAME BUERAS LUCIA G.	10H	202	11-12	11-12	11-12	11-12

TABLA 4.2.2 Codificación de las materias

CODIFICACION	NOMBRE DE LA MATERIA
0	Matemáticas y Estadística
1	Matemáticas aplicadas a la Economía I
2	Matemáticas aplicadas a la Economía II
3	Inferencia Estadística

TABLA 4.2.3 Codificación de los maestros

CODIFICACION	NOMBRE DEL MAESTRO
0	Zepeda Millanes Claudia A.
1	Chacara Montes María Mercedes
2	Dórame Bueras Lucía
3	González Esquer Gabriel
4	Valenzuela Sastré María T.

La codificación de los horarios fue igual para casi todas las áreas, exceptuando Contabilidad y Economía. La codificación quedó de la siguiente manera:

TABLA 4.2.4 Codificación de los horarios

CONTABILIDAD		ECONOMIA		OTRAS ÁREAS	
HORARIO	CODIFICACIÓN	HORARIO	CODIFICACIÓN	HORARIO	CODIFICACIÓN
07-08	0	07-08	0	07-08	0
08-09	1	08-09	1	08-09	1
09-10	2	09-10	2	09-10	2
10-11	3	10-11	3	10-11	3
11-12	4	11-12	4	11-12	4
12-13	5	12-13	5	12-13	5
13-14	6	13-14	6	13-14	6
14-15	7	15-16	7	15-16	7
15-16	8	16-17	8	16-17	8
16-17	9	17-18	9	17-18	9
17-18	10	18-19	10	18-19	10
18-19	11	19-20	11	19-20	11
19-20	12	20-21	12	20-21	12
20-21	13	21-22	13		
21-22	14				

Una vez asignada la numeración (codificación) para los maestros, las materias y los horarios, pasamos a la hoja personal del maestro (FIGURA 4.2.1) donde se codificaron los horarios y materias que solicitaban de acuerdo a lo establecido en las tablas anteriores (TABLAS 4.2.2, 4.2.3 Y 4.2.4). Solo se consideraron las materias del área, es decir si un maestro solicitó dar dos materias, una en Contabilidad y otra en Ciencias Sociales, se consideró solo la materia de Contabilidad para la codificación del área Contable y la materia de Ciencias Sociales en su respectiva codificación. De esta manera si un maestro imparte cursos en varias áreas, aparecerá en todas ellas con diferente número en su codificación. Esto no afecta ya que la programación se hizo por separado para cada área.

Cuando se terminó de codificar la hoja de programación de los maestros, se pasó a construir una matriz para cada maestro (ver FIGURA 4.2.3), esta matriz consistió en 3 renglones, en el primero representa el número que le tocó al maestro en la codificación de maestros, el segundo renglón representa los horarios que tiene el maestro disponible para impartir las materias de acuerdo a su hoja personal, y el tercer renglón representa las materias que desea impartir.

El número de columnas es diferente para cada maestro ya que depende del número de horarios disponibles y del número de materias que da en el área. Así, si un maestro tiene 11 horarios disponibles e imparte dos materias diferentes del área, su matriz tendrá (11 x 2) 22 columnas, (ver primera codificación de la FIGURA 4.2.3). Si otro maestro en la misma área tiene 10 horarios disponibles e imparte tres materias, su matriz consistirá de (10 x 3) 30 columnas (ver segunda codificación de la FIGURA 4.2.3).

Le llamaremos un *bloque* al que se forma con el recorrido de cada horario con todas las materias disponibles, así en el primer ejemplo anterior donde se tiene un total de 22 columnas, éstas son agrupadas en 11 bloques (de 2 columnas cada uno). El número de renglones es el mismo para todos los maestros.

Una vez terminadas las matrices de todos los maestros de cada área, se le asignó un número negativo (el costo académico), este número significa la prioridad en la programación, mientras más negativo es el número asignado tiene mayor prioridad en la programación (ya que la función objetivo es minimizar). Con estos valores negativos se formó un vector para que el programa priorizara a qué maestro programar primero, segundo etc.

La asignación del número dependió de su categoría en el tabulador de la Universidad y su antigüedad, de tal forma que, el maestro cuya categoría es T.C. (Tiempo Completo) y tiene mayor antigüedad, tiene asignado el número más chico (más negativo). Así se continuó para todos los maestros cuya categoría es T.C. hasta llegar al de menor antigüedad.

FIGURA 4.2.3 Codificación de los maestros, materias y horarios.

Zepeda Milanés Claudia A.

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	3	4	4	5	5	6	6	7	7	8	8	9	9	10	10	11	11	12	12	13	13
2	3	2	3	2	3	2	3	2	3	2	3	2	3	2	3	2	3	2	3	2	3

Chácara Montes María Mercedes

1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	1	1	1	2	2	2	3	3	3	4	4	4	5	5	5	7	7	7	8	8	8	9	9	9	1	1	1
																											0	0	0
0	1	2	0	1	2	0	1	2	0	1	2	0	1	2	0	1	2	0	1	2	0	1	2	0	1	2	0	1	2

Dórame Bueras Lucia G.

2	2	2	2	2	2	2	2	2	2	2	2
3	3	4	4	5	5	6	6	7	7	8	8
0	3	0	3	0	3	0	3	0	3	0	3

González Esquer Gabriel

3
7
0

Valenzuela Sastre María T.

4	4
1	7
2	2

Una vez agotados los maestros de Tiempo Completo, se continuó con los meseros cuya categoría es H.S.I. (Horas Sueltas Indeterminados), los que a su vez se priorizaron por antigüedad, y por último los maestros H.S.D. (Horas Sueltas Determinados) que son los que se contratan por tiempo determinado, por lo general por un semestre, de la misma manera se priorizaron por antigüedad. Así el maestro cuya categoría es H.S. determinado y tiene la antigüedad más pequeña se le asignó el número -1 (menos uno), que es el mayor de todos.

Por último se determinó el rango [l_1, L_2], mínimo y máximo de grupos que se le deben programar a cada maestro. Esto depende de su categoría de la siguiente manera:

- Tiempo Completo, se le programan mínimo dos grupos y máximo tres, pero puede reducirse a un solo grupo si el maestro tiene algún puesto administrativo.

- Horas Sueltas Indeterminado, se le programan mínimo el número de horas que tiene indeterminadas y máximo 5 grupos, si éstos son de 5 horas a la semana, pero pueden ser 6 si tiene algún grupo de 4 horas a la semana.
- Horas Sueltas Determinado, se le programan mínimo un grupo y máximo 5 o 6 de la misma forma que el anterior, de tal manera que el número máximo de horas que imparta a la semana no exceda de 25 horas.

Una vez terminada esta codificación (FIGURA 4.2.3), se formaron los vectores que se utilizaron como información inicial para alimentar el programa.

Los 19 vectores que se utilizaron se describen a continuación:

- Los vectores del maestro:

1. **Cuántos en grupo:** En este vector se colocó el tamaño de cada bloque en la matriz de cada maestro, para que el programa pudiera distinguir entre los datos de uno y otro maestro. Su tamaño coincide con el número de bloques de todas las matrices.
2. **Grupos maestro:** Este vector indica el número de veces que aparece la codificación de cada maestro. Su tamaño coincide con el número de maestros que se programaron.
3. **Tipo maestro:** Aquí se colocó la codificación de cada maestro tantas veces como columnas tenga su matriz. Su tamaño coincide con el número total de columnas que se formaron con todas las matrices.
4. **Tipo horario:** En este vector se coloca la codificación de los horarios, es decir el segundo renglón de todas las matrices. Su tamaño coincide con el anterior.
5. **Tipo curso:** Al igual que las dos anteriores su tamaño coincide con el total de columnas de todas las materias, ya que este vector se forma con la codificación de las materias, es decir el tercer renglón de todas las matrices.
6. **Costo académico:** Aquí se coloca el número que se asignó a cada maestro dependiendo de su antigüedad y categoría.
7. **Mínima carga:** Se captura el número mínimo de horas que se le deben programar a cada maestro. Su tamaño coincide con el número de maestros que se programaron.
8. **Máxima carga:** Al igual que el anterior se captura el número máximo de grupos que se le deben programar a cada maestro. Su tamaño es el mismo que el anterior

9. **Vector maestro:** En este vector se coloca el número del maestro en la codificación, solo de aquellos maestros en los que se deben programar ciertas materias varias veces, esto es debido a su categoría

- El vector de espacio físico.

10. **Dep. salones utilizados:** Este vector nos indica cuantos de cuantos salones se dispusieron en cada horario que se codificó. Su tamaño coincide con el número de horarios que se codificaron.

- Los vectores de materias:

11. **Dep. cursos prog. :** Este vector nos indica cuantos cursos programó el departamento para cada materia. Su tamaño depende del número de materias programadas.
12. **Vector materia:** Aquí se coloca la materia (su codificación).
13. **Vector veces:** Aquí se coloca el número de veces que se le debe programar la materia. El tamaño de los tres vectores es el mismo, y depende del número de maestros que tengan esta característica; aunque puede ser mayor, pues un maestro que deba ser programado en dos o tres materias, aumentará el tamaño, ya que se capturará en el vector maestro tantas veces como materias se le deban de programar.
14. **Vector uno materia:** Aquí se coloca la codificación de la materia que se va a programar varias veces.
15. **Vector uno veces:** Aquí va el número de veces que se va a programar la materia requerida.

    El tamaño de estos vectores es el mismo y depende de la solicitud de los diferentes departamentos en la programación.
16. **Vector dos materia:** Aquí se captura la materia codificada
17. **Vector dos veces:** Y aquí el número de veces que se desea programar. El tamaño de estos tres últimos vectores depende directamente de la solicitud de los departamentos, es decir es diferente para cada área de las que se programaron.

- Los vectores de horario.

18. **Vector uno horario:** Estos tres vectores están relacionados entre sí. En éste primer vector se coloca el horario, en el que se quiera que cierta materia se programe varias veces.
19. **Vector dos mañana-tarde:** Estos tres últimos vectores se utilizaron para programar al igual que los anteriores, una materia varias veces, pero cuando no se especifica el horario, sino solo se pide que se programe en la mañana o en la tarde y cuantas veces

Estos 19 vectores se construyeron para cada una de las áreas que se programaron, una vez terminados, se introdujeron en el programa para su "corrida".

4.3 RESULTADOS OBTENIDOS

El programa se elaboró en lenguaje C++, versión 3.0, la corrida se llevó a cabo en una computadora marca LANIX 486, con 32 MHz de velocidad.

El tiempo que tardó el programa en obtener la mejor solución (SOLUCIÓN ÓPTIMA, se encontró en el rango de 45 a 90 segundos, esto dependió del área que se programó, como se muestra a continuación.

TABLA 4.3.1. Tiempos obtenidos (en segundos).

TIEMPOS OBTENIDOS HASTA ENCONTRAR LA SOLUCIÓN ÓPTIMA	
	TIEMPO EN SEGUNDOS
PEQUEÑOS	
Economía	45
Sociología y Administración Pública	50
Ciencias de la Comunicación	56
Químico Biológicas	64
MEDIANOS	
Ingeniería Industrial y de Sistemas	70
Derecho	73
Matemáticas	80
GRANDES	
Contabilidad y Administración	85
Ciencias e Ingeniería	90

4.4 COMPARACIÓN CON ANTERIORES PROGRAMACIONES (TIEMPOS Y RESULTADOS)

Como ya comentamos anteriormente el tiempo de programación es verdaderamente pequeño, esto ahorra mucho tiempo en la programación y permite hacer cambios cuantas veces se quiera, ya que el tiempo en la programación nunca superó los 2 minutos.

El mayor tiempo empleado fue en la codificación, ya que el programa no contó en ese momento con un programa de codificación y captura que formaran los vectores. Sin embargo una vez terminados estos programas el tiempo en la captura será de igual forma mínimo, ya que solo se requerirá capturar las *hojas personales* de cada maestro una primera vez, y modificarlas si es necesario, ya que un maestro de tipo H.S.I. o H.S.D. puede pasar a ser M.T.C, por un periodo de tiempo o de manera indefinida, si gana una plaza.

La codificación y captura a mano se llevó una semana y media trabajando de tres a cinco horas diarias, lo que representa el 25% del tiempo que le toma al departamento de Matemáticas hacer la primera programación, sobre la cual hacen modificaciones,empleando la prueba y error.

Se comparó la programación hecha por el departamento (que le llamaremos manual) y se comparó con los datos de salida del programa que se elaboró con el método de búsqueda tabú, y se obtuvo lo siguiente:

- Todos los maestros fueron programados en las materias que pedían en el horario que marcaron como disponible, coincidiendo con la programación manual en este punto.
- El programa *compactó* los horarios de varios maestros que la programación manual no pudo, de tal forma que se eliminaron las horas libres entre las horas de clase.
- En los casos en que no se pudo compactar el horario a cero horas libres, el programa tubo un avance significativo en la compactación de las horas libres entre las horas de clase, es decir, un maestro que tenía una programación "clase-libre-clase-libre-clase", cambio su programación por "clase-libre-libre-clase-clase", esto es preferido por los maestros pues dos horas libres juntas les son de más utilidad que separadas.

Esto significa que el uso de la técnica de búsqueda tabú, aunado al uso de las técnicas computacionales para la realización de la programación de horarios en el departamento de Matemáticas de la Universidad de Sonora, tuvo un éxito indiscutible y sin precedentes en ésta área.

CAPÍTULO V

CONCLUSIONES

El método de búsqueda tabú mostró ser mucho más efectivo que otros métodos investigados para la programación de horarios y las ventajas se pueden apreciar en:

- La rapidez en la captura de datos,
- Permite manipular un número grande de requerimientos y restricciones en la programación que otros programas no pueden,
- Permite modificar la programación cuantas veces se quiera ya que,
- La obtención de resultados no supera los dos segundos y
- Los resultados obtenidos fueron mucho mejores que los que se habían obtenido en semestres anteriores, ya que se compactaron muchos horarios y se mejoró significativamente en la asignación de salones.

Aunque el programa mostró su efectividad indiscutible, el uso de este programa no ha sido implementado en ningún Departamento de la Universidad de Sonora, por cuestiones que sobrepasan la actuación de los que trabajamos árduamente en la elaboración de este trabajo.

ANEXOS Y REFERENCIAS BIBLIOGRÁFICAS

ANEXOS

1. N.L. Lawrie, 1969.
EL modelo de programación lineal entera para la asignación de horarios desarrollada por N.L. Lawrie en 1969, se llevó a cabo en una escuela de educación básica en Inglaterra. Lawrie desarrolló su trabajo para los 260 alumnos inscritos en el tercer año escolar, la matrícula total de alumnos en esa escuela era de 1060.

2. D.C. Wood en 1968.
La técnica de coloración fue probada por D.C. Wood con matrices del orden de 20, 50 y 100, obteniendo las probabilidades de conflicto de 0.25, 0.5 y 0.75 por ciento respectivamente.

3. JEAN AUBIN and JACQUES A. FERLAND[†].
Ambos doctores han puesto en práctica este modelo matemático para resolver no solo problemas de organización de horarios, sino de optimización, de transportación, de aprovechamiento de recursos y de minimización de rutas.

BIBLIOGRAFÍA

1. LAWRIE, N. L. (1968). School Timetabling by Computer, Aspects of Educational Technology, Vol. II, Menthuen, London, 1968.

2. WOOD, D.C. (1968). A System for Computing University Examination Timetables, The Computer Journal, Vol. II, p. 41.

3. AUBIN, J. AND FERLAND, A., (1987). A Large Scale Timetabling Problem, Computer and Operations Research, Vol. 16, No. 1, p. 67-77, 1989.

4. DE LOS COBOS S, (1994). Tesis Doctoral "La Técnica de la Búsqueda Tabú y sus Aplicaciones". UNAM.

5. GLOVER, F. Tabu Search, Part I, ORSA Journal on Computing (1989), Vol. 1, No. 3, p. 190 - 206.

6. GLOVER, F. Tabu Search, Part II, ORSA Journal on Computing (1990), Vol 2, No. 1, p. 4 - 31.

7. LAGUNA, M. A Guide to Implementing Tabu Search (1994), Investigación Operativa, Vol. 4, No. 1, p. 5 - 23.

8. HERTZ, A. AND WERRA, D. Using Tabu Search Techniques for Graph Coloring, Computing 39 (1987), p. 345 - 351.

9. HERTZ, A. AND WERRA, D. The Tabu Search Metaheuristic: How we used it, Annals of Mathematics and Artificial Intelligence I (1990), p. 111 - 121.

10. DE WERRA, D. An Introduction to timetabling, European Journal of Operational Research, Vol. 19, (1985), p. 151 - 162.

11. FLORES, P. Solución a un problema real de programación de maestros - horarios - cursos con el algoritmo de búsqueda tabú, Actas de Resúmenes Extendidos, Primer Encuentro Latino Iberoamericano de Optimización (Nov. 1997), p. 216 - 224.

12. LOUSEAU, I. Y LAPLAGNE, E. Modelo y Heurísticas para un problema real de timetabling, Actas de Resúmenes Extendidos, Primer Encuentro Latino Iberoamericano de Optimización (Nov. 1997), p. 192 - 195.

13. KRI, F., SOLAR, M., PARADA, V. Y JARA, J.L. Modelos Paralelos para Algoritmos Genéticos, Actas de Resúmenes Extendidos, Primer Encuentro Latino Iberoamericano de Optimización (Nov. 1997), p. 198 - 203.

14. LÓPEZ, M., HETZ, E. Y SAN MARTIN, R. Metodología para la programación de labores agrícolas mecanizadas en predios con cultivos múltiples, Actas de Resúmenes Extendidos, Primer Encuentro Latino Iberoamericano de Optimización (Nov. 1997), p. 204 - 208.

15. PARRA, A.M., BACKHOUSE, P. CH. Y DEL CAMPO, P.P. Sistema de Asignación de Horarios y Salas para la Actividad Académica de una Institución de Educación Superior. Actas de Resúmenes Extendidos, Primer Encuentro Latino Iberoamericano de Optimización (Nov. 1997), p. 210 - 215.