REVISTA INGENIERÍA E INVESTIGACIÓN. VOL. 25 No. 3, DICIEMBRE DE 2005. 92 - 100

Programación de exámenes: un enfoque práctico

A practical approach to scheduling examinations

Luis Gerardo Astaiza A. [1]

RESUMEN

La programación de exámenes es un problema de optimización combinatoria bien estudiado. A pesar de eso, el número creciente de estudiantes y la incorporación de la modularidad en muchas instituciones de educación superior ha resultado en un incremento significativo en su complejidad, imponiendo aún más dificultades a los administradores de la universidad, quienes deben encontrar una solución, a menudo sin la ayuda de un computador. En este artículo presentamos un enfoque practico para la programación de los exámenes, constituido de varias etapas tales como: una para elaborar la matriz de conflictos, otra para seleccionar los salones apropiados en la programación de los exámenes, una otra conformar grupos de cursos que puedan estar programados a la misma hora sin ningún conflicto, otra para permitir restricciones en exámenes conjuntos para todos los grupos de cualquier asignatura o permitir especificaciones para cualquier examen particular, otra para definir la hora y la asignación del salón, y finalmente, otra para publicar la programación a los usuarios (estudiantes, profesores y administradores).

Palabras clave: programación de examen, asignación de salones, interacción con el usuario, sistema basado en hoja electrónica, simulación.

ABSTRACT

The scheduling of exams is a well-studied combinatorial optimisation problem. However, the growth in student numbers and the advent of modularity in many institutions of higher education has resulted in a significant increase in its complexity, imposing even more difficulties on university administrators who must resolve situations, often without having recourse to any computerised aid. This paper presents a practical approach towards timetabling examinations, consisting of several phases. Such phases would include constructing a conflict matrix, selecting suitable rooms for use when scheduling exams, establishing groups of courses which can be timetabled at the same time so that clashes are not incurred, imposing restrictions for joint exams for all groups from any particular course or specifying an exam in particular, fixing the times and allocating rooms and making the timetable available to students, teachers and administrators.

Keywords: exam scheduling, room allocation, user interaction, spreadsheet based system, simulation.

Recibido: agosto 25 de 2004
Aceptado: agosto 26 de 2005

Introducción

Sin lugar a dudas, uno de los aspectos más importantes de la vida universitaria es la programación de las actividades académicas, a saber: el horario de clases y la programación de exámenes. Tanto el primero como el segundo representan objetos abstractos que ejercen una autoridad absoluta en la forma como la institución realiza sus actividades. Cualquier persona que haga parte de la universidad, bien sea como docente o estudiante, debe implícitamente aceptar el hecho de que sus movimientos y pensamientos podrán estar ampliamente determinados por cada uno de ellos.

1 Ingeniero mecánico, M. Sc. en ingeniería de sistemas, M. Sc. en investigación operacional, profesor de la Facultad de Ingeniería, Universidad Nacional de Colombia, e-mail: lgastaizaa@unal.edu.co

El horario de actividades describe la ubicación de profesores y estudiantes en el curso de la semana, las actividades pedagógicas (clases magistrales, conferencias, seminarios, laboratorios, talleres, clínicas y atención a pacientes), o exámenes que se llevan a cabo y la fecha y hora. Este puede ser considerado como un conjunto de encuentros. Un encuentro debe presentar ciertos componentes. En primer lugar, hay la necesidad de la presencia de un docente; en segundo término, hay una participación de estudiantes provenientes de diferentes programas curriculares. En tercer lugar, el encuentro requiere de un espacio docente particular, esto depende del tipo de encuentro. Finalmente, el encuentro debe realizarse en un horario específico. En esta forma, el problema puede ser visto como de asignación de recursos, donde estos son docentes, estudiantes, salones y horarios. Otros recursos podrían ser la dotación y los implementos docentes de los espacios físicos.

En particular, este artículo detalla la programación de exámenes (finales, de habilitación, validación etc.), la cual requiere la asignación de un número dado de exámenes dentro de un marco de tiempo. Esta programación presenta las siguientes características: primero, sólo hay un examen para cada curso. Segundo, ningún estudiante puede tener más de un examen programado en el mismo periodo de tiempo. Tercero, los exámenes pueden distribuirse en forma tal que los estudiantes tengan tiempo de estudio. Cuarto, el examen puede programarse específicamente para un curso, programarse conjuntamente para todos los grupos de un curso o dejar que el sistema realice la programación intentando inicialmente una asignación con un horario igual al de clases. Quinto, Hay un número máximo de exámenes por periodo en consideración al número finito de espacios.

Formulación del problema

El problema de programación en general consiste en asignar un número de eventos en un número limitado de periodos de tiempo cumpliendo un conjunto de restricciones estrictas o duras y otro de restricciones deseables o suaves. Las restricciones estrictas son consideradas fundamentales de cumplir en términos de desarrollar una programación práctica, así: ningún recurso puede estar presente en más de un lugar a la vez y los recursos requeridos no deben exceder los recursos disponibles. De otro lado, las restricciones deseables o suaves no son esenciales de cumplir, tales como: horario del examen, restricciones de tiempo entre exámenes, distribución de los exámenes en el calendario o asignación de recursos entre otros.

En particular, la programación de exámenes consiste en asignar un número de exámenes en un número limitado de periodos de tiempo de forma tal que ningún estudiante requiera presentar más de un examen a la vez. En dicha programación encontramos los dos tipos de conflictos:

- Restricciones estrictas. Este término describe situaciones en las cuales se programan exámenes en conflicto en el mismo periodo. Esta situación es altamente indeseable, ya que implica que algunos estudiantes tengan que presentar en el mismo periodo de tiempo dos o más exámenes de cursos diferentes

- Restricciones deseables. Representan situaciones donde dos exámenes en conflicto no se programan en el mismo periodo pero están programados en periodos demasiado cercanos el uno al otro. Para el caso no deseamos que haya estudiantes que tengan que presentar dos exámenes en periodos consecutivos, o en el mismo día. Satisfacer esta restricción es totalmente impráctico, en cuyo caso es considerado como una restricción suave.

Esto da una idea de los problemas que debe enfrentar el administrador. En un mundo ideal, podría ser hermoso estar en capacidad de optimizar todos los criterios. Sin embargo, es necesario un compromiso, ya que el tiempo total para exámenes es limitado.

Enfoques al problema de programación

Los diferentes enfoques que han sido empleados en la programación de exámenes pueden clasificarse en: Métodos secuenciales, de agrupación, métodos basados en restricciones, metaheurísticos, de criterios múltiples, y métodos basados en el razonamiento de casos.

Los métodos secuenciales están basados en la coloración heurística de grafos. Entre las heurísticas utilizadas podemos citar: en primer lugar, la asignación de los exámenes de mayor grado de conflictos con otros exámenes. La segunda heurística utiliza el grado ponderado más grande. Cada conflicto se pondera por el número de estudiantes involucrados en él. La tercera heurística selecciona el grado de saturación que corresponde al examen que tenga el menor número de periodos válidos para su programación. Finalmente, tenemos la heurística basada en el grado del color, la cual prioriza aquellos exámenes que tengan el número más grande de conflictos con exámenes que ya han sido programados.

Los métodos de agrupación utilizan dos ideas. En primer lugar, el conjunto de exámenes es dividido en grupos que cumplan con las restricciones estrictas, y en segundo lugar, los grupos son asignados a periodos de tiempo para satisfacer las restricciones deseables o suaves.

Los enfoques basados en restricciones consideran que el problema de programación puede modelarse como un conjunto de variables (v. gr. exámenes) a los cuales tienen que ser asignados valores (v. gr. recursos tales como periodos de tiempo y salones) para satisfacer un número de restricciones.

Los enfoques metaheurísticos utilizan las técnicas de recocido simulado, búsqueda tabú, o de algoritmos genéticos. Más aún, estos métodos pueden presentar algunas innovaciones tales como la combinación de algoritmos genéticos, búsqueda local y heurísticas, dando lugar a lo que se conoce como los algoritmos miméticos.

Finalmente, están los enfoques basados en el razonamiento de casos, en los cuales los problemas resueltos previamente son utilizados para resolver nuevos problemas de programación.

Método de solución

La elaboración de la programación de exámenes en cada periodo lectivo es un problema común de todas las instituciones de educación superior. Generalmente se realiza manualmente, pero dado el incremento de estudiantes y la flexibilidad curricular, es necesario desarrollar métodos automáticos que permitan asistir al administrador.

La aplicación de los computadores a los problemas de programación de horarios tiene una larga y variada historia (Burke). La primera generación de programas de computador se presentó en los inicios de la década de los sesenta, los cuales intentaban reducir el trabajo administrativo asociado. Broker, 1964, y Cole, 1964, presentaron enfoques heurísticos a la programación.

Welsh y Powell, 1967, señalaron la similitud entre este problema y el de coloración de un grafo. Aquí, los vértices corresponden a los cursos, y los arcos entre ellos representan los conflictos. Desde las observaciones realizadas por Welsh y Powell, 1967, muchos algoritmos han sido diseñados basados en la coloración de grafos como parte fundamental del sistema.

Desroches, Laporte y Rousseau, 1978, presentaron HOREX, un programa de computador que seguía una serie de pasos para lograr la programación de exámenes. El programa inicialmente encuentra una coloración del gráfico de conflictos, luego trata de igualar el número de exámenes en cada periodo. Puesto que generalmente no hay una secuencia para los exámenes, los periodos son ordenados y los fines de semana incorporados de forma tal que se minimice el número de exámenes consecutivos, los cuales se asignan a salones para lograr un sistema completo.

Burke, Elliman y Were, 1993, presentaron algoritmos de coloración de grafos y asignación de salones. Para intentar vencer los problemas intratables asociaron el modelo a un sistema de hoja electrónica tal que el usuario pueda guiarse de manera informada y útil.

El proceso de encontrar un periodo para cada examen tal que no haya conflictos, es equivalente a asignar colores a vértices de un grafo de manera que vértices adyacentes tengan colores diferentes. Esto a su vez ha demostrado que corresponde al conjunto de problemas NP-completos, lo cual significa que la realización de una investigación exhaustiva para su elaboración no es posible en un tiempo razonable. Muchos algoritmos han sido propuestos, la mayoría utilizando heurísticas basadas en coloración de grafos. Estos pueden producir adecuados resultados, pero usualmente ignoran que haya suficientes asientos para todos los exámenes asignados en un periodo particular y no permiten la realización de ninguna investigación del espacio de soluciones.

Para vencer las dificultades, recurrimos al algoritmo de Burke para la coloración de grafos con miras a la agrupación de exámenes que se puedan programar sin conflictos, y descomponemos el proceso en etapas así:

- Definir calendario de exámenes
- Establecer espacios físicos docentes a utilizar
- Seleccionar grupos de asignaturas
- Administrar grupos de asignaturas
- Elaborar matriz de conflictos
- Definir grupos de exámenes
- Proveer análisis de vecindad
- Establecer día de examen
- Administrar exámenes conjuntos
- Especificar exámenes especiales
- Asignar hora y salón
- Administrar programación
- Proveer reportes consulta
- Proveer informes

Etapa 1. Definir calendario de exámenes

En esta etapa se establece el calendario para cada tipo de examen (incluyendo indicador de día de examen y franja(s) horaria(s) para la programación de exámenes de una facultad. El indicativo z se reserva para exámenes que serán programados por el profesor durante la realización de las clases.

En cada indicador de día de examen, el usuario puede especificar hasta seis franjas de examen (hora 1, hora 2,…hora 6). El dominio del campo será cualquier valor entre 6:00 a.m. y 20:00 p. m. El valor de omisión será 00 (no programado).

Etapa 2. Definir espacios físicos docentes a utilizar

Para la programación de exámenes de una facultad para un periodo académico es necesario definir la planta física a utilizar. El usuario debe estar en capacidad de seleccionar la planta física a utilizar de la base de datos según edificio y tipos de salón (aula teórica, aula seminario, aula torreón, aula máxima, taller, laboratorio, etc.). Así mismo, el sistema debe desplegar la lista de salones del edificio seleccionado, clasificado por tipo de salón y de forma tal que el usuario marque aquellos salones o espacios que no puede utilizar por tener acceso restringido.

Etapa 3. Seleccionar grupos de asignatura

El propósito de esta etapa es seleccionar de la programación de cursos aquellos que requieren la programación de un tipo de examen (final, habilitación, validación) en un periodo académico para una facultad y un nivel de estudios. En esta etapa se deben crear las tablas requeridas en la base de datos para la programación de exámenes. Acorde a la especificación de parámetros, se seleccionan los grupos de asignatura a partir de la programación a la que se le programará el examen. Para determinar en cuál facultad se programará el examen se seguirá el criterio de a qué facultad pertenece el plan de estudios para la cual se programó la asignatura o si no hay especificación del plan, en la facultad a la cual esté adscrito el departamento que ofreció la asignatura

Etapa 4. Administrar grupos de asignatura

En esta etapa se eliminan de las asignaturas y grupos seleccionados aquellos a los cuales no se les programará examen.

Etapa 5. Establecer matriz de conflicto

Establecer tabla de matriz de conflictos entre las asignaturas que se seleccionaron para realizar la programación de un tipo de examen en una facultad y un nivel de estudios. Para este propósito debe existir un registro por cada pareja en el cual se presente el número de estudiantes inscritos simultáneamente en las dos asignaturas.

Etapa 6. Definir agrupación de exámenes

La definición de grupo de examen se puede lograr utilizando la matriz de conflictos y un algoritmo de coloración de franjas de exámenes. Esta definición de grupos de examen implica que las asignaturas pertenecientes a un grupo puedan programarse de forma simultánea (ejemplo: ningún estudiante puede estar inscrito simultáneamente en Matemáticas I y Matemáticas II, dado que la primera es requisito de la segunda o están en niveles de formación tal que es imposible cursarlas simultáneamente). Estos grupos tendrán cada uno un identificador.

Etapa 7. Proveer análisis de vecindad

Con mirar a definir el día de examen de una agrupación es conveniente realizar un análisis de la cantidad de conflictos que se presentan entre una agrupación seleccionada y las restantes. Esto permite programar exámenes de forma tal que exista la menor cantidad de conflictos entre exámenes programados bien sea anterior o posterior al grupo analizado.

Etapa 8. Establecer día de examen de una agrupación

En esta etapa debemos establecer el día de examen de una agrupación de asignaturas de un tipo de examen para una facultad en un periodo académico.

Para este caso el sistema basado en los parámetros suministrados por el usuario debe desplegar las asignaturas que conforman dicha agrupación incluyendo la especificación del día de examen si este ya ha sido asignado.

Finalmente, el usuario debe especificar el nuevo día de examen de los disponibles (lista informativa) si desea asignar o modificar el día asignado.

Etapa 9. Especificar día de examen para una asignatura

El propósito de esta etapa es definir el día de examen a una asignatura considerando la información establecida en la matriz de conflictos para ella y las fechas de examen utilizadas por las asignaturas con las cuales existe conflicto.

El usuario debe especificar periodo, tipo de examen y asignatura. Con base en esta información, el sistema prepara matriz de fechas de examen según la información del calendario, procesa la matriz de conflictos y de programación de examen en relación con las asignaturas en conflicto con la asignatura especificada y las contabiliza en el día de examen en la cual esté programada. Finalmente, despliega la información consolidada para que el usuario seleccione un día de examen.

Etapa 10. Administrar exámenes conjuntos

En una facultad algunas veces se requiere que todos los grupos de una asignatura se programen simultáneamente para realizar una prueba única. Para lograr este propósito es necesario administrar la programación de exámenes conjuntos (hora de examen). Esta etapa debe permitir: inserciones, modificaciones, borrados y consultas a la información de exámenes conjuntos en la programación de exámenes de una facultad en un periodo académico. El usuario suministrará la información requerida en la pantalla diseñada para tal efecto.

Etapa 11. Especificar exámenes especiales

En algunos casos especiales es necesario especificar la programación de examen para un grupo de asignatura (horario y espacio físico). Esta etapa debe permitir establecer la hora y el lugar de examen de un grupo de asignatura en la programación de un tipo de examen de una facultad en un periodo académico. El usuario debe especificar periodo, facultad, tipo de examen, asignatura y grupo. Una vez localizado el registro y desplegada la información de programación de salones para el día considerado, definir edificio, salón, hora del examen, y diligenciar el indicativo de especificación al grabar la información.

Etapa 12. Asignar hora y salón

Aplicar el algoritmo de asignación de hora y salón para un tipo de examen para cada grupo de asignatura que requiere programación de examen. El usuario especifica periodo, facultad, nivel, tipos de examen y de ejecución (definitiva o simulada) y utilizando las tablas requeridas se ejecuta el algoritmo de asignación de hora y salón. La opción de tipo de ejecución simulada es para permitir el análisis de una posible programación y las modificaciones del caso y luego realizar la ejecución definitiva.

Inicialmente se puede realizar la asignación a las asignaturas que correspondan a un departamento académico utilizando para esto la planta física correspondiente y luego asignar los salones indistintamente.

Etapa 13. Administrar la programación de exámenes

Esta etapa debe permitir modificar la programación de examen a un grupo de asignaturas. Para la toma de la decisión el sistema debe proveer la cantidad de conflictos del grupo con las otras asignaturas establecidas en la matriz de conflictos según estén inscritos en el grupo de la asignatura. El usuario debe especificar, además de la información relacionada con el examen que se está procesando, el grupo de asignaturas de interés. El sistema consulta la matriz de conflictos para esta asignatura con el fin de preparar arreglo el que permita consolidar el número de estudiantes inscritos en cada una de estas asignaturas y que a la vez están inscritos en el grupo de asignaturas considerado. Finalmente informa al usuario los datos consolidados incorporando el día de examen de cada una de ellas (tabla de examen de asignatura), permitiendo en esta forma que el usuario programe el grupo de asignaturas.

Etapa 14. Proveer reportes consultas

El sistema debe permitir entre otras, las siguientes consultas:

- Conflictos de una asignatura
- Exámenes conjuntos
- Especificación de exámenes
- Ocupación de planta física
- Exámenes sin asignación de salón
- Programación de exámenes.

Además de realizar la consulta por pantalla, esta debe poderse enviar a otro medio de salida. El usuario suministrará la información requerida en la pantalla diseñada para tal efecto.

Etapa 15. Proveer informes

El sistema debe suministrar información de la programación de exámenes vía web a los involucrados en el proceso (docentes, estudiantes, administradores, etc.) en forma similar al suministro de información de la programación de cursos.

Algoritmos especiales

Definición de los grupos de examen

Con miras a realizar la definición de los grupos de examen de forma automática, podemos recurrir al algoritmo planteado por Burke. [3]

Definiciones

Suponga que tenemos una grafica G con vértices x e y. G: xy es la gráfica obtenida a partir de G al reemplazar x e y por un solo nodo conectado a todos los nodos adyacentes bien sea a x o y o ambos. Los vértices x e y se han integrado para producir x' o (y').

Similarmente G - $\{x\}$ es la grafica obtenida al remover el vértice x y todos los arcos que incluye x desde G. Una gráfica nula es aquella sin ningún vértice o arco.

Ahora, para cualquier vértice dado x, podemos construir un conjunto de ternas de vértices como sigue:

$$\begin{gathered}(x, z_{(1,1)}, y_1)\\(x, z_{(1,2)}, y_1)\\(x, z_{(1,m_1)}, y_1)\\.\\.\\(x, z_{(i,1)}, y_i)\\.\\.\\(x, z_{(n,m_n)}, y_n)\end{gathered} \qquad (1)$$

donde el primer vértice es adyacente al segundo y el segundo lo es al tercero (esto podría significar que el tercer vértice es adyacente a un adyacente al primero).

Podemos asumir que este conjunto no contiene ninguna terna donde el primer vértice sea adyacente a, o es lo mismo que, el tercero por simplemente no incorporar tales ternas. Debemos encontrar i tal que $m_i = \max(m_1, m_2, ..., m_n)$ y definimos y_i por $\hat{x}$.

Esto no necesariamente es único, pero podemos simplemente seleccionar a $\hat{x}$ el que llegue a ser el primer y_i en la lista tal que $M_i = \max(m_1, m_2, ..., m_n)$

Algoritmo para obtener la coloración de un grafo dado

El algoritmo que se detalla a continuación según Burke, 1993, está basado en la combinación del algoritmo de Dutton y Brigham, 1981, y el algoritmo de Tehrani, 1975.

Asuma que disponemos de un grafo simple *G* como entrada

P1. Defina *j*: = 1, *H*:= *G*

P2. Defina v_j como el vértice de máximo grado en H.

P3. Construya todas las ternas según se estableció anteriormente y encuentre $\hat{x}$. Si no existe tal vértice (v. gr. $m_i = 0$) seleccione un vértice de grado máximo no adyacente a v_j.

P4. Fusione v_j y $\hat{x}$ en v_j y repita desde P3 hasta que ninguna selección sea posible, luego inicie de nuevo desde P2 con $H := H - \{v_J\}$, $j := j + 1$.

P5. Cuando no queden vértices, termine y coloree todos los vértices integrados en v_i ($1 \leq i \leq j$) del color i, esta podrá ser la *j*-ésima coloración del grafo G.

El algoritmo trabaja de forma recursiva, encontrando cada conjunto de color sucesivamente removiendo todos los vértices coloreados del grafo y mirando por un nuevo conjunto en el grafo reducido. Para cada color se selecciona el vértice remanente de mayor grado y uno a uno los vértices con el número máximo de vértices comunes adyacente son fusionados a este. Cuando todos los vértices de la componente actual del grafo estén coloreados, el proceso se repite de nuevo con las otras componentes iniciando con el vértice de mayor grado. Ninguno de los vértices fusionados es adyacente el uno al otro tal que pueden ser considerados un conjunto de color.

Ejemplo

Asuma que tiene quince exámenes que requieren ser programados con el siguiente grafo de conflictos.

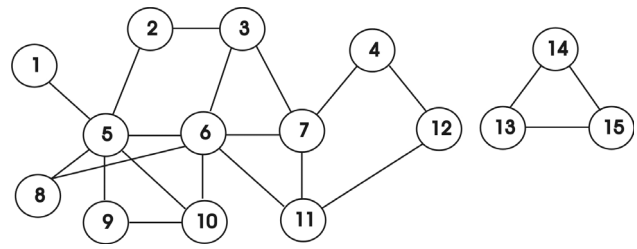

Figura 1

Este grafo puede ser coloreado V_1 utilizando el algoritmo. Los vértices 5 y 6 tienen grado 6, así que inicie con el vértice 5 como el vértice V_1.

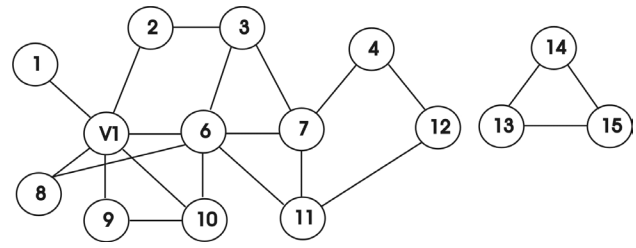

Figura 2

En la Figura 2 únicamente los vértices 3, 7 y 11 tienen algún vértice común adyacente al vértice V_1. Los vértices 7 y 11 tienen un vértice común adyacente con el vértice V_1, mientras que el vértice 3 tiene dos vértices comunes adyacentes al vértice V_1, así que seleccione el vértice 3 para integrarlo con el vértice V_1

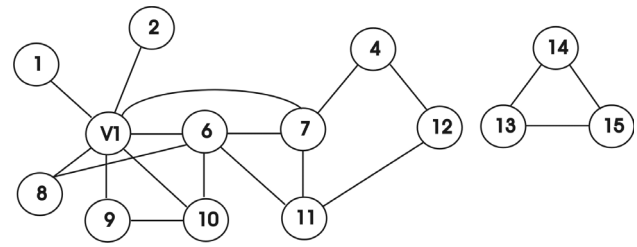

Figura 3

Continuando en esta forma, seleccione el vértice 11 en la Figura 3, para integrarlo con el vértice V_1.

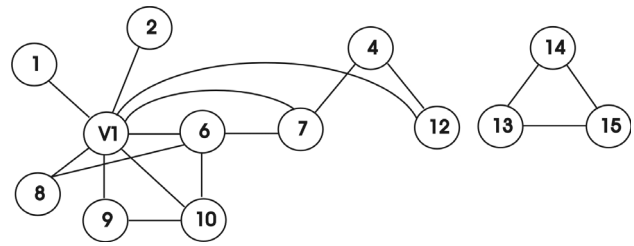

Figura 4

En la Figura 4 seleccione el vértice 4 para integrarlo con el vértice V_1.

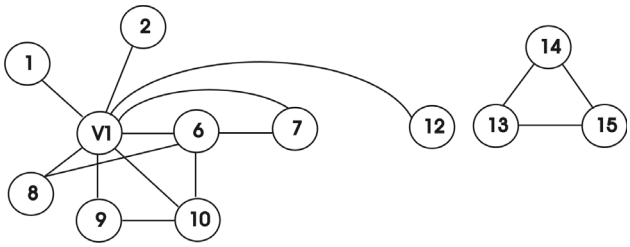

Figura 5

En la Figura 5 seleccionamos el vértice 13 para integrarlo con el vértice V_1.

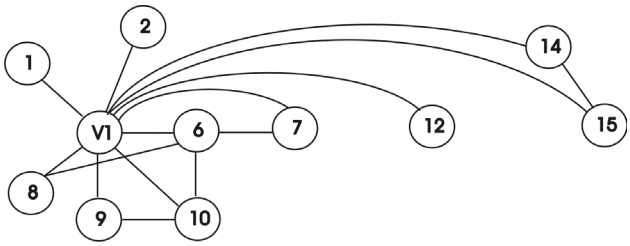

Figura 6

En la Figura 6 todos los vértices son adyacentes al vértice V_1 dando por terminado la conformación de un grupo de examen.

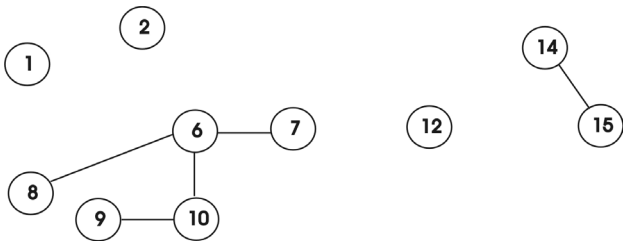

Figura 7

En la Figura 7 se inicia nuevamente el proceso, seleccione el nodo 6 como el nodo V_2.

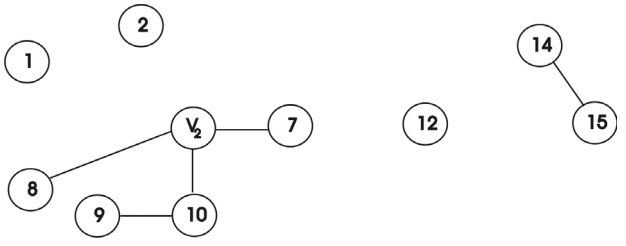

Figura 8

En la Figura 8 repitiendo el proceso, seleccionamos los nodos 9, 1, 14, 1, 2 y 12 para integrarlos al nodo V_2. Finalmente quedan los nodos 7, 8, 10 y 15, los cuales según el procedimiento, conforman el tercer color.

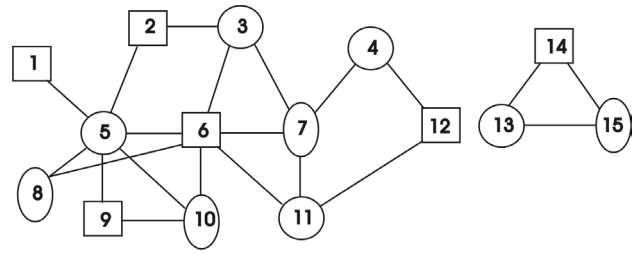

Figura 9

En la Figura 9, se presenta la agrupación final, representada en los tres conjuntos de colores (denotado por tres formas diferentes).

Así, sin ninguna otra restricción, la programación puede realizarse en tres periodos, un periodo por color. En este escenario poco probable e idealista podría haber terminado todo lo que necesitamos hacer. Sin embargo, este no es el caso en la mayoría de las situaciones. En un caso simple, como el de este ejemplo, el algoritmo es probable que logre una solución óptima. Para programaciones o grafos más grandes esto es mucho menos que probable que llegue a ser el caso, dado que el problema de coloración del grafo es un problema NP-completo. Sin embargo, el algoritmo podrá por lo menos dar una coloración razonable si no óptima.

La agrupación de exámenes así elaborada se asocia al día de examen y se le aplica el algoritmo que se detalla a continuación, el cual en pocos minutos puede elaborar la programación de una facultad de aprox. 1000 cursos y 5000 estudiantes.

Algoritmo de asignación de hora y salón

P11. Preparar tablas y áreas de trabajo.

P12. Establecer parámetros (Periodo_ académico, facultad, nivel de estudios, tipo de examen, tipo de actividades, fechas y horas de examen para cada agrupación de examen).

P13. Cargar tablas de asignatura y de grupos de asignaturas según facultad

P131. Defina

N _ asignaturas = 0

N _ grupos = 0

Lea primera asignatura de la base de datos

/* Cargue asignaturas a programar */

P132. Mientras haya asignaturas para leer

Filtrar cargue según facultad, nivel y tipo de examen

Haga

N_ asignaturas = N_ asignaturas + 1

Seleccione datos asignatura en la matriz asignatura

Apuntador _ grupo = N_ grupos + 1.

/* Cargue grupos de la asignatura */

Lea primer grupo de asignaturas.

P133. Mientras haya grupos de asignaturas

N_ grupos = N_ grupos + 1

Cargue datos del grupo en la matriz grupo

Establezca hora _ examen como la primera franja horaria menor más cercana a la hora del grupo de asignaturas

Seleccione el siguiente grupo de asignaturas de la tabla de grupos de examen

/* Cargue salones a programar */

P14. Cargue planta física

P141. Establecer clasificación ascendente por cupo

/* Realice asignación de hora y salón para cada franja */

Seleccione primera franja de la matriz de franjas

P15. Mientras haya franjas de examen, realice asignación

P151. Preparar áreas para la franja

/* Ocupar salones preasignados */

P152. Para cada grupo con examen preasignado incremente contador de ocupación para el salón.

/* Asignación de exámenes conjuntos */

Seleccione primera asignatura de la tabla de asignaturas

P153. Mientras haya asignaturas pendientes

Si asignatura con examen conjunto (hora conjunto > 00) y día de examen = franja seleccionad•a

- Seleccione primer grupo de la asignatura
- Ajuste hora conjunto a la franja más cercana <= hora conjunto
- Mientras haya grupos pendientes
  - Si grupo sin asignación
    - •Buscar salón disponible (ind = 0) con franja horaria conjunta y cupo >= inscritos
    - •Si encontró salón incremente indicador de ocupación enmatriz de salones y asigne salón en la matriz de grupo de examen, y si ejecución definitiva, actualice base de datos de grupo examen
  - Seleccione grupo siguiente de la asignatura

Seleccione siguiente asignatura

/* Asignación de exámenes no conjuntos */

Seleccione primera asignatura de la tabla de asignaturas

P154. Mientras haya asignaturas pendientes

Si asignatura con examen no conjunto (hora conjunto = 00)

- Considere primer grupo de la asignatura
- •Mientras haya grupos pendientes de la asignatura
  - Mientras asignación pendiente
    - Ajuste hora grupo a la franja más cercana <= hora del grupo
    - •Localizar franja en la matriz de salones que corresponda a la hora del grupo
    - Mientras asignación pendiente y haya salones en la franja

Si salón disponible y cupo > = inscritos

Entonces (incremente ocupación en matriz de Salones y asigne salón en la matriz de grupo

examen 7 y si ejecución definitiva, actualice base de datos de grupo examen)

De lo contrario seleccione el siguiente salón de la franja horaria

- Definir FILA = primera fila de la matriz de salones
- Mientras asignación pendiente y FILA <= última fila de la matriz de salones

- Hacer hora = hora 1 de la franja
- Mientras asignación pendiente y haya franja horaria

Si salón disponible y cupo > = inscritos

Entonces (incremente ocupación en matriz de salones y asigne salón en la matriz de grupo examen, y si ejecución definitiva, actualice base de datos de grupo examen)

De lo contrario seleccione la siguiente franja horaria

- Incrementar fila en uno

  - Seleccionar el grupo siguiente de la asignatura

  - Seleccione la siguiente asignatura

P155. Seleccione la siguiente franja y repita desde P15.

P16. Cerrar proceso.

Conclusión

Este artículo presenta un enfoque práctico a la programación de exámenes dada su gran flexibilidad para su utilización por parte de los administradores. Su implementación en etapas permite tomar las decisiones adecuadas en cada una de ellas.

La agrupación de exámenes permite su distribución por días de examen y por franjas horarias de forma tal que el estudiante no tenga que presentar más de un examen por franja o inclusive por día de examen. Así mismo, el análisis de los grupos permite una distribución que conceda al estudiante mayor tiempo de estudio. Más aún, permite incorporar la programación de exámenes conjuntos o especificación del examen para un grupo de asignatura. Adicionalmente, el sistema facilita asignar hora y salón a cada agrupación dentro de sus franjas horarias y considerando las restricciones impuestas bien sea de forma simulada o de manera definitiva. También, el sistema provee informes relacionados con la matriz de conflictos, ocupación de salones por día de examen, cursos sin asignación etc, Adicionalmente, este subsistema debe estar incorporado al Sistema de Información Académica para realizar la integración con las bases de datos del mismo y así proveer los informes a los usuarios del sistema (profesores y estudiantes) utilizando la red. Finalmente, esta investigación continuará considerando otras alternativas o realizando mejoras a esta.

Bibliografía

Broders final Examination Scheduling Comm ACM, 7, 494-498, 1964.

Bullnheimer, Bernd. "*An examination scheduling model to maximize student´ study time*", 2nd international conference on the practice and theory of automated timetabling, 1997.

Burke, E. K.; D.G. Elliman y R. Weare. "*A university timetabling system based on graph colouring and constraint manipulation*", Department of computer science, University of Nottingham, UK.

Burke, E. K.; P.P. Newall y R. F. Weare. "*A memetic algorithm for university exam timetabling*", En edmund burke y peter ross, editors, the practice and theory of automated timetabling. Selected papers from the 1st international conference, lecture notes in computer science 1153, pp. 241-250, Springer-Verlag, Berlin, 1996.

Burke, E. K.; D.G. Elliman y R. Weare. (1993b). "*Automated scheduling of university exams*". Proceedings of I.E.E: Colloquium on "resource scheduling for large scale planning systems", digest No. 1993 /144.

Carter, M. W.; G. Laporte y J.W. Chinneck. "*A general examination scheduling system*". Interfaces, 11: 109-120, 1994

Carter, M. W. "*A survey of practical applications of examinations timetabling algorithms*", OR practice 34, 193-202,1986.

Cole, A. J. The preparation of examination timetables using a small store computer. Comp. Jrnl 7, 117-121,1964.

Desroches, S.; Laporte, G; Rousseau, J.M. Horex, "*A computer program for the construction of examination shedules*", Infor 16, 294-298,1978.

Dutton, R. D. And Brigham R.C. "*A new graph coloring algorithm comp*", The Computer Journal, 24, 85-86,198.

Johnson, D. "*Timetabling university examinations*", Journal of the operations research society, Vol. 1, No. 41, pp. 39-47, 1990.

Karp, R. M., "*Reductibility among combinatorial problems*", In complexity of computer computations, Plenum Press, New York,1972.

Liam, T.; G. Merlot, Natashia Boland, Barry D. Huges; Peter J. Stuckey. "*A hibrid algorithm for the examination timetabling problem*", Department of mathematics and statistics and department of computer science and software engineering, the University of Melbourne, Victoria 3010, Australia, 2002.

Tehran, I A. "*Un algorithme de coloration cahiers du center d'études do recherche opérationnelle*", 17, 395-398,1975.

Welsh, D. J. A. And Powell, M. B. "*An upper bound for the chromatic number of a graph and its application to timetabling problems*" comp. Jrnl. 10, 85-86,1967

Wong, Tony; Pascal, Cote; Paul Gely. "*Final exam timetabling: A practical approach*". Proceedings of the 2002 IEEE canadian conference on electrical & computer engineering, 2002.