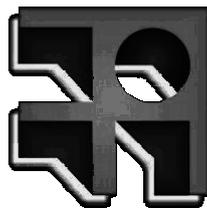


ASIGNACIÓN DE SALONES Y HORARIOS INFORME

PROYECTO DE GRADO
2002/2003



Integrantes:

Ismael Cuevas

Jorge Panzera

Yanil Ugalde

Tutores:

María Urquhart

Joaquín Goyoaga

Taller V - 2002 <i>Asignación de horarios y Salones a Cursos</i>	Tipo de Documento: <i>Informe del Proyecto</i>	Finalizado:
	Versión: 1.0	Aprobado:
Preparado por: IC, JP, YU		Aprobado por:

Contenido

<u>1</u>	<u>Introducción</u>	4
<u>1.1</u>	<u>OBJETIVO</u>	4
<u>1.2</u>	<u>DEFINICIONES</u>	4
<u>1.3</u>	<u>PANORAMA GENERAL DEL PROYECTO</u>	5
<u>2</u>	<u>Contexto</u>	7
<u>2.1</u>	<u>DESCRIPCIÓN DEL PROBLEMA</u>	7
<u>2.2</u>	<u>RESTRICCIONES Y PREFERENCIAS</u>	7
<u>2.3</u>	<u>PRESENTACIÓN DEL PROBLEMA PLANTEADO COMO UN PROBLEMA DE ASIGNACIÓN</u>	9
<u>2.4</u>	<u>ESTUDIO DE TÉCNICAS APLICABLES A PROBLEMAS DE ASIGNACIÓN</u>	9
<u>2.5</u>	<u>TABÚ SEARCH</u>	11
<u>3</u>	<u>Descripción del Producto</u>	14
<u>3.1</u>	<u>FUNCIONALIDAD</u>	14
<u>3.2</u>	<u>ENTORNO DE FUNCIONAMIENTO DEL PRODUCTO</u>	14
<u>4</u>	<u>Decisiones de Diseño e Implementación</u>	16
<u>4.1</u>	<u>PLATAFORMA A UTILIZAR</u>	16
<u>4.2</u>	<u>IMPLEMENTACIÓN DEL TABÚ SEARCH</u>	16
<u>4.3</u>	<u>VISUALIZACIÓN DE LA SOLUCIÓN</u>	20
<u>5</u>	<u>Calibración del Tabu Search</u>	24
<u>6</u>	<u>Conclusiones</u>	26
<u>7</u>	<u>Trabajos Futuros</u>	28
	<u>Referencias</u>	31
	<u>Anexo 1. Organización de la Facultad</u>	32
<u>1.1</u>	<u>CARRERAS</u>	32
<u>1.2</u>	<u>AÑOS Y SEMESTRES</u>	32

Taller V - 2002 <i>Asignación de horarios y Salones a Cursos</i>	Tipo de Documento: <i>Informe del Proyecto</i>	Finalizado:
	Versión: 1.0	Aprobado:
Preparado por: IC, JP, YU		Aprobado por:

<u>1.3</u>	<u>ASIGNATURAS</u>	32
<u>1.4</u>	<u>BLOQUES</u>	34
<u>1.5</u>	<u>GRUPOS</u>	34
<u>1.6</u>	<u>CLASES</u>	34
<u>1.7</u>	<u>SALONES</u>	35
<u>1.8</u>	<u>RECURSOS</u>	35
<u>1.9</u>	<u>HORARIOS DE DICTADO DE CLASES</u>	36
<u>1.10</u>	<u>RESERVA DE SALONES</u>	36

Anexo 2. Formalización del Problema 37

<u>2.1</u>	<u>PRESENTACIÓN DEL PROBLEMA PLANTEADO COMO UN PROBLEMA DE ASIGNACIÓN</u>	37
<u>2.2</u>	<u>FORMALIZACIÓN DEL PROBLEMA</u>	41

Anexo 3. Estudio de Técnicas Aplicables a Problemas de Asignación 47

<u>3.1</u>	<u>ANT SYSTEMS</u>	47
<u>3.2</u>	<u>ALGORITMOS GENÉTICOS</u>	49
<u>3.3</u>	<u>TABÚ SEARCH</u>	50
<u>3.4</u>	<u>SIMULATED ANNEALING</u>	52
<u>3.5</u>	<u>ELECCIÓN DE LA HEURÍSTICA A UTILIZAR</u>	53
<u>3.6</u>	<u>ADAPTACIÓN DEL TABÚ SEARCH A LA ASIGNACIÓN DE SALONES Y HORARIOS</u>	55

Anexo 4. Elección de la Plataforma 60

<u>4.1</u>	<u>SISTEMA OPERATIVO</u>	60
<u>4.2</u>	<u>MOTOR DE BASE DE DATOS</u>	60
<u>4.3</u>	<u>LENGUAJES DE PROGRAMACIÓN</u>	62

Anexo 5. Calibración y Resultados Obtenidos 64

<u>5.1</u>	<u>CALIBRACIÓN DEL TABÚ SEARCH</u>	64
<u>5.2</u>	<u>PESOS UTILIZADOS PARA RESTRICCIONES Y PREFERENCIAS</u>	77

Apéndices 79

<u>APÉNDICE A.</u>	<u>DOCUMENTO DE ANÁLISIS</u>	79
<u>APÉNDICE B.</u>	<u>DOCUMENTO DE DISEÑO</u>	79
<u>APÉNDICE C.</u>	<u>MANUAL DE USUARIO</u>	79

Taller V - 2002 <i>Asignación de horarios y Salones a Cursos</i>	Tipo de Documento: <i>Informe del Proyecto</i>	Finalizado:
	Versión: 1.0	Aprobado:
Preparado por: IC, JP, YU		Aprobado por:

1 Introducción

1.1 Objetivo

La asignación de salones y horarios a los diferentes cursos de la Facultad de Ingeniería constituye una tarea compleja que por cada semestre insume aproximadamente un mes y medio de trabajo. Dicho trabajo es prácticamente manual en su totalidad, con sus posteriores modificaciones durante el semestre. Surge entonces, la necesidad de contar con una herramienta de apoyo que permita facilitar y agilizar dicha tarea a los funcionarios encargados de la misma.

En consecuencia, los objetivos principales de este Proyecto de Taller V son los siguientes:

- ◆ Desarrollar una herramienta que, aplicando un método heurístico, construya una asignación de forma automática. Es deseable que la misma sea una buena aproximación a la solución ideal y se obtenga en un tiempo razonable.
- ◆ Ofrecer interfaces para visualizar y manipular las asignaciones de forma práctica y sencilla. Esta funcionalidad debe permitir la asignación de salones y horarios de forma manual.
- ◆ Se incluirán además, funcionalidades complementarias para la administración de las asignaciones de horarios y salones, como ser listados a extraer, manejo de reservas, y mantenimiento de datos involucrados en la asignación.

1.2 Definiciones

Asignación general:

Conjunto de asignaciones de horarios y salones a clases de un determinado semestre (por ejemplo, la asignación del primer semestre del año 2002).

Asignación de clase:

Asignación de un determinado salón, día y horario a una clase, dentro de una asignación general (por ejemplo, el salón, día y horario asignado a la primer clase semanal del grupo 1 de teórico de Cálculo 1 en el primer semestre del año 2002)

Asignaturas equivalentes:

Dos asignaturas serán consideradas equivalentes si tienen nombres o códigos diferentes, pero se trata de la misma asignatura. Es decir, son dictadas en forma conjunta.

Conjunto de asignaturas equivalentes:

La relación de equivalencia entre dos asignaturas particiona al conjunto de todas las asignaturas en clases, cada una de las cuales será considerada como un conjunto de asignaturas equivalentes.

Asignatura primaria:

Dado un conjunto de asignaturas equivalentes, solamente una de ellas se utilizará en los procesos de asignación. Dicha asignatura es la llamada primaria. Cada conjunto de asignaturas

Taller V - 2002 <i>Asignación de horarios y Salones a Cursos</i>	Tipo de Documento: <i>Informe del Proyecto</i>	Finalizado:
	Versión: 1.0	Aprobado:
Preparado por: IC, JP, YU		Aprobado por:

equivalentes tendrá una asignatura primaria, que es la única que se considerará en representación del conjunto en la asignación.

Asignatura secundaria:

Son todas las asignaturas que pertenecen a un conjunto de asignaturas equivalentes y no son primarias en él.

Rango Horario:

Rangos de horas durante las cuales se lleva a cabo el dictado de las clases; se define, para cada Año/Semestre a tener en cuenta, los días de la semana en los cuales se dictan clases, y el rango de horas para cada día.

Horarios válidos:

Se considerarán horarios válidos los horarios que estén contenidos en el rango horario de dictado de clases.

1.3 Panorama General del Proyecto

En este punto presentamos un breve repaso a todo el proceso de desarrollo de este proyecto, tratando de dar una idea general de las actividades llevadas a cabo para la realización del mismo.

Inicialmente, se mantuvieron varias reuniones con el personal de bedelía encargado de la tarea de asignación de salones y horarios (Adriana Taboas y Alvaro Pereira). Dichas reuniones tuvieron como objetivo interiorizarnos con la organización interna de la Facultad en lo que respecta al manejo de las asignaciones. En este momento tomamos conciencia de que el problema involucra un volumen de datos importante, y de que, para lograr una solución satisfactoria al mismo, es totalmente necesario que estos datos sean completos y correctos. Por lo tanto, se hacía necesario que la herramienta a desarrollar no solamente generara una solución aproximada en forma automática, sino que además debía ser capaz de manejar los datos necesarios para lograrlo. La descripción de la organización de las carreras en la Facultad se encuentra en la Sección 2.1.1. de este documento. De estas reuniones también surgieron los deseos e inquietudes expresados por el personal de bedelía con respecto al problema planteado, una descripción detallada de los requerimientos planteados se encuentra en el Apéndice A – Documento de Análisis de Requerimientos.

Para lograr una mayor correctitud en los datos involucrados, nos contactamos con la persona del Seciu encargada de la Facultad de Ingeniería, Gabriela Luján. En dichas reuniones se logró que el Seciu ampliara la funcionalidad del sistema que se usa en la bedelía de la Facultad, para que el mismo contenga consultas sobre carreras, asignaturas y cantidad de alumnos inscriptos por semestre. Estas consultas se vuelcan en el sistema de asignación utilizando archivos de texto. De esta manera se ahorra una considerable cantidad de tiempo en el ingreso de estos datos, además de evitar los posibles errores en el mismo.

El paso siguiente fue el planteo del problema de asignación de salones y horarios como un problema de optimización y su posterior formalización. El hecho de tener el problema formalizado fue de suma importancia en el momento de la implementación, ya que simplificó sobremanera el desarrollo del método heurístico implementado. Para una descripción detallada del planteo del problema como un problema de optimización y su formalización, referirse al Anexo 2.

Luego se llevó a cabo un extenso estudio de los posibles métodos heurísticos que se podrían aplicar a este tipo de problemas. Las fuentes para este relevamiento fueron

Taller V - 2002 <i>Asignación de horarios y Salones a Cursos</i>	Tipo de Documento: <i>Informe del Proyecto</i>	Finalizado:
	Versión: 1.0	Aprobado:
Preparado por: IC, JP, YU		Aprobado por:

principalmente material encontrado en Internet, en la biblioteca del Instituto de Computación de la Facultad de Ingeniería y también material provisto por nuestra tutora María Urquhart.

Una vez que se tuvo un conocimiento adecuado de los distintos métodos, se procedió a la elección de uno de ellos, el Tabú Search, para realizar la búsqueda de una solución al problema planteado. Para ver en forma detallada el estudio de las distintas heurísticas y los motivos de la elección del Tabú Search, referirse al Anexo 3.

El Tabú Search es un método genérico de resolución de problemas de optimización, por lo que el paso siguiente consistió en adaptar el mismo al problema de asignación de salones y horarios. Por el detalle de esta adaptación, ver el Anexo 3.

En este punto ya contábamos con un amplio conocimiento del problema a resolver, se había elegido el método a aplicar, y se habían especificado los requerimientos a desarrollar. Por lo tanto, se procedió a realizar el diseño de la aplicación. El mismo se realizó orientado a objetos, utilizando UML. El detalle del mismo se puede encontrar en el Apéndice B – Documento de Diseño.

Las siguientes decisiones a tomar correspondían a la plataforma y herramientas a utilizar para implementar la aplicación. Se tuvieron en cuenta varias opciones, así como consideraciones de tipo legal y económico en la elección de las mismas, las herramientas elegidas finalmente fueron Java y MySQL. Si se desea ver en detalle el proceso de elección de la plataforma con las opciones evaluadas, referirse al Anexo 4.

Completado el diseño de la aplicación y elegidas la plataforma a utilizar, se procedió a la implementación de la aplicación. La misma fue trabajosa al principio, ya que ninguno de los integrantes del grupo estaba familiarizado con el lenguaje elegido, y su curva de aprendizaje es bastante pronunciada, pero luego de terminada la misma, creemos que fue una buena elección.

A medida que avanzaba la implementación, y por supuesto también luego de finalizada la misma, se fueron llevando a cabo pruebas al método heurístico desarrollado. En principio estas pruebas se realizaron utilizando datos generados de forma aleatoria y luego utilizando datos reales. Para lograr que el Tabú Search tenga una buena performance, es necesario definir valores para ciertos parámetros que este involucra, a este proceso se le llama calibración. Para ver una descripción detallada de las pruebas realizadas y de la calibración, referirse al Anexo 5 Sección 5.1.

Una vez finalizadas las etapas de implementación y calibración del algoritmo de optimización, se realizó una presentación de la aplicación a los funcionarios de bedelía involucrados y a una comisión locativa de la Facultad. De esta reunión surgió el deseo de ambas partes de usar de forma efectiva la herramienta desarrollada, para realizar la asignación de salones y horarios del segundo semestre del año 2003.

Finalmente, se realizó la confección del Manual de Usuario y la puesta en producción de la aplicación en la bedelía de la Facultad. La instalación inicial incluyó la carga de la base de datos con la información real correspondiente al segundo semestre del año 2002, asignación que se tomará como base para realizar la del segundo semestre del año 2003.

Taller V - 2002 <i>Asignación de horarios y Salones a Cursos</i>	Tipo de Documento: <i>Informe del Proyecto</i>	Finalizado:
	Versión: 1.0	Aprobado:
Preparado por: IC, JP, YU		Aprobado por:

2 Contexto

2.1 Descripción del Problema

El problema a resolver consiste en la asignación de horarios y salones a asignaturas, orientado a la organización de las carreras de la Facultad de Ingeniería. Esta asignación debe cumplir con ciertas restricciones que surgen teniendo en cuenta dicha organización, los recursos disponibles en cada salón, y la cantidad de alumnos inscriptos en las diferentes asignaturas. Es deseable además, que se satisfagan ciertas preferencias que, si bien no son obligatorias, pueden contribuir a que la solución encontrada sea de mejor calidad.

La solución encontrada será una asignación sugerida, que se presentará de forma tal que se puedan visualizar los posibles conflictos y preferencias no satisfechas, y servirá como base para que el usuario pueda modificarla, acercándola más a la solución óptima.

Se llevó a cabo un relevamiento para poder confeccionar una descripción general del problema de asignación, esta surge de los datos aportados por Santiago Rivas, Asistente Académico de la Decana, Adriana Taboas, Jefa de Bedelía, Álvaro Pereira, funcionario de Bedelía (encargado de realizar manualmente las tareas de asignación de horarios y salones a cursos actualmente), Nicolás Conde, funcionario de Recursos Informáticos, y Gabriela Luján, funcionaria del SECIU encargada de soporte a la Bedelía de la Facultad de Ingeniería.

La estructura organizativa de las carreras de la Facultad de Ingeniería es la siguiente:

Las diferentes carreras se componen de años, cada uno de los cuales se divide en dos semestres. A su vez, cada semestre consta de un conjunto de asignaturas teóricas y/o prácticas, que pueden estar divididas en grupos, y se dictan en una determinada cantidad de clases semanales, cada una con una duración establecida. Por lo tanto, la unidad básica que debemos asignar a los salones en un determinado horario son las clases.

Asumiendo que la mayoría de las asignaturas tienen una frecuencia semanal para el dictado de sus clases y que la organización de las carreras está dada por semestres, determinamos que la asignación básica que debemos realizar es para una semana de un determinado semestre.

En el Anexo 1 se detallan las principales características relevadas para cada una de las componentes que forman la estructura descrita, las cuales servirán de base para la conformación de una estructura que permita almacenar todos los datos necesarios que servirán ya sea como entrada de datos para resolver el problema de asignación como para visualizar y cambiar la solución sugerida.

2.2 Restricciones y Preferencias

Una asignación de horarios y salones debe cumplir con ciertos criterios para poder ser una solución válida al problema, algunos de estos criterios son más importantes que otros por lo que los dividimos en dos grupos:

Llamaremos restricciones a aquellos criterios de mayor importancia, los cuales es obligatorio que se cumplan dentro de una asignación.

Llamaremos preferencias a aquellos criterios de menor importancia, los cuales sería deseable (pero no imprescindible) que se cumplan. Por lo general estos criterios se utilizarán

Taller V - 2002 <i>Asignación de horarios y Salones a Cursos</i>	Tipo de Documento: <i>Informe del Proyecto</i>	Finalizado:
	Versión: 1.0	Aprobado:
Preparado por: IC, JP, YU		Aprobado por:

para diferenciar cuales asignaciones son “mejores” que otras. Más adelante en este documento veremos como se maneja el hecho de que una asignación es “mejor” que otra.

Estos criterios fueron definidos conjuntamente con los funcionarios de la Bedelía de la Facultad luego de diversas reuniones.

2.2.1 Restricciones

1. La cantidad de horas asignadas a una clase no puede ser menor a la duración de la clase.
2. Una clase debe asignarse a un único salón.
3. Una pareja (horario, salón) debe asignarse a una única clase.
4. Las clases deben asignarse dentro de los horarios válidos.
5. Las clases de un mismo semestre, carrera y grupo no pueden superponerse. Esta restricción no aplica a las asignaturas electivas, y pueden superponerse entre sí clases que correspondan a asignaturas con más de un grupo.
6. Las horas asignadas a una clase deben ser contiguas.
7. Las horas asignadas a una clase deben estar dentro de un mismo día.
8. Los grupos que tienen un bloque especificado sólo pueden asignarse dentro de la franja horaria correspondiente al bloque.
9. No pueden dictarse dos clases de un mismo grupo y asignatura un mismo día.

2.2.2 Preferencias

1. Es deseable que no queden clases sin asignar.
2. Las clases deberían asignarse dentro del rango horario de preferencia especificado.
3. Una clase no debería asignarse a un salón con menor capacidad de la requerida.
4. Es deseable que el salón asignado a una clase cuente con los recursos necesarios para el dictado de la misma. Por ej., si la clase necesita enchufe para un retroproyector, el salón debe tener enchufes.
5. Es deseable que los teóricos y los prácticos correspondientes se dicten en horarios contiguos.
6. No sería bueno que se utilizaran salones de gran capacidad para clases de pocos alumnos.
7. Es conveniente que las clases teóricas de un grupo se repartan en forma uniforme en la semana.
8. Similitud de horarios en las distintas clases teóricas de un grupo.

Taller V - 2002 <i>Asignación de horarios y Salones a Cursos</i>	Tipo de Documento: <i>Informe del Proyecto</i>	Finalizado:
	Versión: 1.0	Aprobado:
Preparado por: IC, JP, YU		Aprobado por:

2.3 Presentación del Problema Planteado Como Un Problema De Asignación

El problema de asignación de horarios y salones es, en definitiva, un problema de optimización, o sea que consiste en encontrar un elemento x perteneciente a un dominio D de manera que se minimice el valor de una función f , sujeta a ciertas condiciones.

En este caso, los elementos x son las asignaciones pertenecientes a un espacio de soluciones D , el valor de la función f representa el costo del no cumplimiento de las restricciones y preferencias.

Existen varios tipos de problemas de optimización, el problema de asignación de salones y horarios (llamado en inglés timetable problem) se puede clasificar dentro de los llamados *problemas de asignación (ATP)*.

Dados n ítems y m recursos, el ATP consiste en determinar una asignación de los recursos a los ítems de tal forma de optimizar una función objetivo f y satisfaciendo un conjunto R de restricciones.

En el Anexo 2 Sección 2.1, se presenta la formalización del problema.

2.4 Estudio de Técnicas Aplicables a Problemas de Asignación

En el área de la Investigación Operativa, considerando los problemas de optimización, han surgido hace algunos años métodos de resolución heurísticos. Esto fue motivado por la complejidad de algunos de estos problemas, como el que nos tocó afrontar, que son NP-completos [13], por lo tanto no resulta práctico resolverlos por técnicas exactas, ya que llevarían un tiempo de cálculo inaceptable para la aplicación real.

Con la utilización de *Metaheurísticas*, se han logrado buenas aproximaciones a las soluciones ideales, obtenidas en tiempos razonables.

Se realizó un estudio de las distintas heurísticas que pueden ser aplicadas a problemas de este tipo, así como también se tuvieron en cuenta estudios realizados anteriormente inclusive en la propia Facultad de Ingeniería en un Proyecto anterior.

En la evaluación de antecedentes realizada, se encontró que las heurísticas más utilizadas para resolver este tipo de problema son: Algoritmos Genéticos[8,10,11] (GA), Simulated Annealing [8,9] (SA), Tabú search [2,3,8,14,15] (TS) y Ant Systems (AS) [12].

La información aquí expresada surge en su mayoría de artículos, los cuales podemos encontrar como referencias de este documento. Para obtenerlos, se recurrió a tres fuentes principales: Internet, la biblioteca del Instituto de Computación de la Facultad de Ingeniería y nuestra tutora María Urquhart, quien nos facilitó algunos de ellos.

Para una descripción detallada de las heurísticas evaluadas, referirse al Anexo 3.

Taller V - 2002 <i>Asignación de horarios y Salones a Cursos</i>	Tipo de Documento: <i>Informe del Proyecto</i>	Finalizado:
	Versión: 1.0	Aprobado:
Preparado por: IC, JP, YU		Aprobado por:

2.4.1 Elección de la Heurística a Utilizar

Ant Systems es una de las técnicas que se podría utilizar en este problema, pero debido a que en su planteo original construye una solución totalmente nueva, sin tomar en cuenta soluciones anteriores no será la que se utilizará en este caso. Por otra parte, se tuvo en cuenta que este método fue utilizado en un proyecto anterior [12] y, según las conclusiones de dicho proyecto, no se obtuvieron resultados muy alentadores, incluso trabajando con un conjunto reducido de datos de prueba.

GA (Algoritmos Genéticos) se basa en la manipulación de un conjunto de soluciones al problema. Dada la complejidad de la estructura de éstas, parece complejo poder generar varias soluciones de partida para el algoritmo. Este aspecto negativo, el de generar varias soluciones de partida, es una de las principales razones por la cual no se eligió esta heurística, aunque es una de las posibles candidatas para una extensión futura del proyecto.

Tanto TS (Tabú Search) como SA (Simulated Annealing) utilizan las técnicas de búsqueda local, las cuales, como se dijo anteriormente, dan la posibilidad de partir de cualquier solución inicial, como así también la posibilidad de interactuar modificando la solución en cualquier punto de la búsqueda.

De hecho, una vez que una asignación ha sido generada puede servir como punto de partida para una nueva búsqueda luego de haber modificado algunas de las restricciones del problema, como por ejemplo, cambiar los rangos horarios de dictado de clases o las preferencias horarias de alguna de ellas. Dicho proceso generalmente lleva a una "buena" asignación para el nuevo problema en un tiempo razonable.

Estas técnicas son muy adecuadas para el problema de asignación de horarios y salones, ya que para la asignación de un semestre en particular, podemos partir de la solución ya obtenida en el mismo semestre para el año anterior, cambiando algunas de las restricciones o preferencias correspondientes al semestre a asignar. Otra ventaja de esto es que la solución encontrada va a ser similar a la anterior, por lo que se mantendría una coherencia semestre a semestre entre las asignaciones.

La tabla 2-1 ilustra algunas de las características que fueron consideradas para la elección del algoritmo a utilizar.

	Tabú Search	Sim. Annealing	Genetic Algor.	Ant Systems
Solución Inicial	Puede usar cualquier solución inicial	Puede usar cualquier solución inicial	Parte de un conjunto de soluciones	No parte de una solución inicial.
Adaptabilidad al problema	Muy buena	Buena	Muy buena	Buena
Usado anteriormente para problemas similares	Si, con buenos resultados	Si, con resultados relativamente buenos	Si, con buenos resultados	Si, pero con un conjunto reducido de datos..
Complejidad de desarrollo	Media	Simple	Compleja	Media
Factibilidad de las Soluciones	Es posible encontrar buenas soluciones.	Es posible encontrar buenas soluciones, lleva más tiempo que otros.	Es posible encontrar buenas soluciones.	Es posible encontrar buenas soluciones, en teoría.

Tabla 2-1: Comparación de metaheurísticas.

Taller V - 2002 <i>Asignación de horarios y Salones a Cursos</i>	Tipo de Documento: <i>Informe del Proyecto</i>	Finalizado:
	Versión: 1.0	Aprobado:
Preparado por: IC, JP, YU		Aprobado por:

Se eligió utilizar Tabú Search teniendo en cuenta lo anterior y estudios ya realizados [8], donde se aplicaron TS, SA y GA al problema de 'Timetable' obteniéndose los mejores resultados para TS. También Tabú Search ha sido reportado por ser muy efectivo en problemas de 'Timetable' en [2,3,8,14,15].

En resumen, las principales razones para utilizar el Tabú Search son:

La posibilidad de partir de cualquier solución inicial

Este punto es muy significativo, ya que nos permite utilizar como solución inicial cualquier asignación de años anteriores, ya que es muy probable que esta no cambie mucho de un año a otro. El partir de una solución anterior reduce significativamente el tiempo de proceso para encontrar una nueva solución. Permite además mantener cierta coherencia en los horarios de un año a otro.

También es posible generar una solución a partir de cualquier solución inicial, lo cual le permitiría al usuario de la herramienta elegir como punto de partida una solución anterior o no. En el caso de que no elija partir de una anterior, se generará una solución de forma aleatoria como punto de partida.

La posibilidad de interactuar modificando la solución actual en cualquier punto de la búsqueda.

Este es otro punto decisivo para la elección. El usuario podría en cualquier momento del proceso de generación de una asignación, frenar el mismo, ver como ha evolucionado la solución propuesta, hacer sus modificaciones y comenzar nuevamente el proceso a partir de esta asignación modificada manualmente.

Estudios anteriores

De la información obtenida en los trabajos tomados como referencia, deducimos que el Tabú Search es la técnica con la cual se obtuvieron los mejores resultados, comparados con las demás técnicas aquí evaluadas.

Por un informe más detallado acerca de el porqué de la elección del Tabú Search como la heurística a utilizar, referirse al Anexo 3 Sección 3.5.

2.5 Tabú Search

Tabú Search es una estrategia para resolver problemas de optimización combinatoria cuyo rango de aplicación es muy amplio.

Es un procedimiento adaptativo con la habilidad de usar otros métodos (como algoritmos de programación lineal o heurísticas especializadas) lo cual hace que pueda obviar las limitaciones de la optimalidad local. Se lo puede clasificar como una meta-heurística diseñada para obtener una aproximación a la solución óptima de un problema de optimización combinatoria.

Para mejorar la eficiencia del proceso de búsqueda, se necesita no solamente llevar registro de la información local (como el valor actual de la función objetivo) sino también la información del proceso de búsqueda. El uso sistemático de la *memoria* es una característica esencial del Tabú Search, como veremos más adelante.

Taller V - 2002 <i>Asignación de horarios y Salones a Cursos</i>	Tipo de Documento: <i>Informe del Proyecto</i>	Finalizado:
	Versión: 1.0	Aprobado:
Preparado por: IC, JP, YU		Aprobado por:

2.5.1 Descripción del Método

Partiendo de una solución inicial s_0 , el algoritmo de Tabú Search explora un subconjunto V del vecindario $N(s)$ de la solución actual s . El elemento de V que proporciona el mínimo valor para la función objetivo se convierte en la nueva solución independientemente del hecho de que su valor sea mejor o peor que el valor de s .

Para prevenir los ciclos, existe una lista que llamaremos *tabu-list*, que es la lista de los movimientos prohibidos de realizar. En esta lista se van guardando los últimos k movimientos aceptados (donde k es un parámetro del método) y se maneja como una cola de largo fijo. Cuando un nuevo movimiento es agregado, el más viejo es sacado de la lista. Aquí es donde podemos observar el uso de la memoria por este método, evitando de esta manera volver a soluciones anteriores.

Puede pensarse que la decisión de que algunos movimientos son *tabú* es demasiado absoluta, por lo que se introduce un mecanismo para cancelar el status de *tabú* de los mismos. Si un movimiento proporciona una mejora significativa de la función objetivo, entonces su status de *tabú* es dejado de lado y la solución resultante de aplicar este movimiento es aceptada como la nueva solución actual. Más precisamente, se define una *función de aspiración* A que, para cada valor v de la función objetivo, devuelve otro valor v' que representa el valor que el algoritmo *aspira* a llegar desde v . Dada la solución actual s , la función objetivo f y una solución vecina s' obtenida a partir de un movimiento m , si $f(s') \leq A(f(s))$, entonces s' puede ser aceptada, incluso si m esta en la *tabu-list*.

El procedimiento se detiene cuando se cumplen algunas de estas condiciones:

- ◆ Se cumplió con un máximo dado T_{smax} de iteraciones sin mejoras en la función objetivo.
- ◆ El valor de la función objetivo para la solución actual es mejor que un límite fijado previamente para el mismo f^* .

Los principales parámetros de este método son:

- ◆ El largo de la *tabu-list* k .
- ◆ La cardinalidad del conjunto V de soluciones vecinas testeadas en cada iteración.
- ◆ El máximo de iteraciones T_{smax} sin conseguir mejorar el valor de la función objetivo.

Seudo código

```

f* := limite fijado para la función objetivo;
s0 := solución inicial en  $S$ ;
nlter := 0; // iteración actual
blter := 0; // iteración en donde se encontró la mejor solución
bestSol := s0;
TL :=  $\emptyset$  // Inicializar la tabú list;
Inicializar función de aspiración  $A$ ;

while (  $f(\text{bestSol}) > f^*$  ) and (  $n\text{lter} - b\text{lter} > T_{smax}$  ) do

    nlter := nlter + 1;
    Generar un conjunto  $V$  de soluciones  $s_i$  en  $N(s)$ , las cuales no son tabú o
        se cumple que  $A(f(s)) \geq f(s_i)$ ;
    s* := solución que minimiza  $f$  sobre  $V$ ;
    Actualizar la tabú list  $TL$  y la función de aspiración  $A$ ;
    if  $f(s^*) < f(\text{bestSol})$  then
        bestSol := s*;

```

Taller V - 2002 <i>Asignación de horarios y Salones a Cursos</i>	Tipo de Documento: <i>Informe del Proyecto</i>	Finalizado:
	Versión: <i>1.0</i>	Aprobado:
Preparado por: <i>IC, JP, YU</i>		Aprobado por:

2.5.2 Adaptación del Tabú Search a la Asignación de Salones y Horarios

El Tabú Search es una estrategia para resolver problemas de optimización con un rango de aplicación muy amplio. Por lo tanto, es necesario adaptar la forma genérica del algoritmo para resolver el problema de Asignación de Clases a Salones y Horarios. Los puntos a tener en cuenta del algoritmo para hacer posible su adaptación son los siguientes:

- ◆ Restricciones y Preferencias
- ◆ La solución inicial s_0
- ◆ Definición de un movimiento y generación de soluciones vecinas
- ◆ La Tabú List
- ◆ La función de Aspiración
- ◆ Calibración del Método
- ◆ Criterio de parada

Para cada uno de estos puntos, es necesario definir de que forma se optimiza su uso con respecto al problema planteado. Esta adaptación se detalla en el Anexo 3 Sección 3.6.

Taller V - 2002 <i>Asignación de horarios y Salones a Cursos</i>	Tipo de Documento: <i>Informe del Proyecto</i>	Finalizado:
	Versión: 1.0	Aprobado:
Preparado por: IC, JP, YU		Aprobado por:

3 Descripción del Producto

3.1 Funcionalidad

En líneas generales, las funciones del producto son las descritas en el Objetivo.

Se desarrolló una herramienta de apoyo a los funcionarios de bedelía, que además de ofrecer interfaces para visualizar y manipular las asignaciones de forma práctica y sencilla, permite optimizar las mismas en forma automática. Dicha optimización se realiza mediante la aplicación del Tabú Search, partiendo de soluciones existentes o creando soluciones completamente nuevas.

Se incluyen además, funcionalidades complementarias para la administración de las asignaciones de horarios y salones, como ser listados a extraer, manejo de reservas, y mantenimiento de datos involucrados en la asignación.

En el documento de Análisis (Apéndice A) se presenta una descripción detallada de los requerimientos implementados.

3.2 Entorno de funcionamiento del producto

3.2.1 Interacción con otros sistemas

El funcionamiento del producto a desarrollar será totalmente autónomo, y no interactuará directamente con otras aplicaciones existentes. Si bien sería interesante acceder a la información que se encuentra en la base de datos del SECIU, esto no es posible debido a las políticas de acceso que dicho organismo posee. Por lo tanto la única interfaz que se prevé es la obtención de datos sobre las carreras, asignaturas e inscriptos, pero la misma se realizará a través de archivos de texto generados por el sistema de Bedelías provisto por el SECIU.

3.2.2 Plataforma

Las máquinas con las que se cuenta actualmente en Bedelía, y sobre las cuales deberá ejecutarse tienen las siguientes características:

Procesadores 486 /Pentium I, II, III
 32 a 512 Mb de Memoria RAM
 Sistema Operativo Windows 95 / Windows NT
 (en este punto se debe tener en cuenta que en un futuro la bedelía puede optar por utilizar otros sistemas operativos, como por ejemplo Linux)

Taller V - 2002 <i>Asignación de horarios y Salones a Cursos</i>	Tipo de Documento: <i>Informe del Proyecto</i>	Finalizado:
	Versión: 1.0	Aprobado:
Preparado por: IC, JP, YU		Aprobado por:

3.2.3 Características del Usuario

Los usuarios a los cuales está destinado el producto poseen conocimientos informáticos a nivel de operador, y están familiarizados con el problema de asignación de salones y horarios a clases, dado que son quienes lo resuelven actualmente en forma manual.

3.2.4 Asunciones y Dependencias

Para realizar una correcta asignación, el producto debe contar con los datos actualizados y correctos referentes a carreras, asignaturas, recursos, cantidad de clases con sus respectivas preferencias horarias, cantidad de inscriptos por asignatura, salones, asignaciones fijas y equivalencia de asignaturas.

Se asumirá entonces, que los datos mencionados están completos y son correctos.

Se recopilan a continuación algunos aspectos fundamentales a tener en cuenta para que los datos sirvan como entrada al algoritmo de optimización.

Capacidades y Recursos de los salones

La información de capacidades y de recursos de infraestructura disponibles en los salones es dinámica, debe actualizarse todos los años haciendo un relevamiento de los mismos.

Asignaturas, carreras y cantidad de alumnos inscriptos por asignatura

Estos datos se encuentran en la base de datos del SECIU. Se proporciona una interfaz para poder utilizar los mismos y no tener que ingresarlos manualmente.

Grupos, clases, carga horaria semanal y horarios de preferencia

Se asumen conocidos en todos los casos para todas las asignaturas. Esta información debe ingresarse manualmente; como por lo general se mantiene año tras año, se estima que solamente requerirá una carga inicial importante, y luego requerirá una cantidad limitada de modificaciones.

3.2.5 Datos y Sistemas previos

El sistema de apoyo a las tareas de asignación de salones y horarios que se utiliza actualmente no realiza las asignaciones, sino que las mismas se hacen manualmente y se ingresan al programa, y lo que éste hace es detectar superposiciones de horarios por salón y reportarlas. No detecta otros conflictos, como superposiciones de horarios por carrera y semestre.

Permite extraer diferentes reportes de horarios asignados por asignatura y salón, así como listados de horas libres por salón, para el manejo de reservas circunstanciales de salones.

Vale la pena destacar que este sistema no permite guardar asignaciones de años anteriores, solo mantiene los datos de la asignación vigente, a diferencia del sistema que se implementó, el cual brinda dicha funcionalidad.

Taller V - 2002 <i>Asignación de horarios y Salones a Cursos</i>	Tipo de Documento: <i>Informe del Proyecto</i>	Finalizado:
	Versión: 1.0	Aprobado:
Preparado por: IC, JP, YU		Aprobado por:

4 Decisiones de Diseño e Implementación

4.1 Plataforma a Utilizar

Para la elección de la plataforma se consideró, por sobre otras características, el costo de los productos a utilizar, que en general son altos. Debido a esto, las decisiones se tomaron priorizando el licenciamiento y la portabilidad de las herramientas consideradas.

Durante esta etapa elegimos el sistema operativo a utilizar, el motor de base de datos en el cual se almacenan los datos de la aplicación y el lenguaje de programación a utilizar para el desarrollo de la misma.

Para ver en detalle las distintas opciones que se tuvieron en cuenta, así como también las características utilizadas para la comparación de los productos, referirse al Anexo 4.

4.2 Implementación del Tabú Search

4.2.1 Forma de la solución

Como se expresa en la formalización del problema, la solución al mismo es una matriz de 3 dimensiones (salones, horarios y clases), cuyos elementos son ceros y unos. Los horarios representan todos los intervalos validos de media hora contenidos en el rango semanal de dictado de clase.

De esta forma, tenemos que:

$$s_{sl,h,c} = \begin{cases} 1 & \text{si la clase "c" se asignó al salón "sl" a la hora "h"} \\ 0 & \text{en otro caso.} \end{cases}$$

Inicialmente, por la complejidad que representa trabajar con esta forma de solución, llevamos esta matriz a dos dimensiones, agrupando dos de ellas (salones y horarios) en una sola dimensión, por lo que la forma de la solución era la que se visualiza en la figura 4-1

	Clase 1	Clase 2	Clase 3	Clase 4	...	Clase n
Salon 1 Dia 1 horario 1	1	0	0	0	...	0
Salon 1 Dia 1 Horario 2	1	0	0	0	...	0
Salon 1 Dia 1 Horario 3	0	0	0	0	...	0
⋮	0⋮0	0⋮0	0⋮0	⋮	⋮	⋮
Salon 1 Dia n Horario 1	0	1	0	0	...	0
Salon 1 Dia n Horario 2	0	1	0	0	...	0
Salon 1 Dia n Horario 3	0	1	0	0	...	0
⋮	0⋮0	0⋮0	0⋮0 1	⋮	...	0⋮0
Salon n Dia n Horario n	0	0	1	0	...	0

Taller V - 2002 <i>Asignación de horarios y Salones a Cursos</i>	Tipo de Documento: <i>Informe del Proyecto</i>	Finalizado:
	Versión: 1.0	Aprobado:
Preparado por: IC, JP, YU		Aprobado por:

Figura 4-1: Matriz solución.

Como se puede observar, esta forma de representar la solución es una matriz dispersa, con muchos ceros y muy pocos unos (que son los elementos significativos de la matriz, ya que representan asignaciones de clase).

Esta forma de representar la solución presenta dos problemas importantes:

- ◆ Desperdicio de memoria (ya que los elementos en cero no representan nada)

Tomando como referencia la asignación del segundo semestre del año 2002, se realizó la siguiente estimación de uso de memoria por cada solución:

Clases	500
Salones	40
Horarios (en medias horas)	158
Celdas (bits)	3,160,000
Bytes por matriz	395,000
Kbytes por matriz	386

Tabla 4-1: Uso de memoria inicial.

Como se explicó anteriormente, en cada iteración el algoritmo debe construir n soluciones vecinas a la actual de entre las cuales se debe escoger la mejor. Es importante destacar, que cada vecino que se generaba durante la ejecución del Tabú Search, estaba representado por una nueva solución. Esto implica que teniendo n vecinos, se requerían $386 \times n$ Kbytes para almacenarlos.

- ◆ Tiempo de procesamiento excesivo (el Tabú Search se basa en muchas recorridas de la solución actual, por lo cual esta forma de representar la solución implicaba recorrer muchas veces una matriz de gran tamaño).

Por estas razones se cambió la representación de la solución, como se describe a continuación.

Cambio en las estructuras de datos

Como parte de los datos, se contaba con dos arrays auxiliares:

Array de clases: Conteniendo todas las clases con datos necesarios para el manejo de restricciones, como la carrera, el grupo, práctica o teórica, recursos necesarios, etc. Los índices de dicho array coincidían con los índices de la dimensión de la matriz que representaba las clases.

Array de Salones-Días-Horarios: Conteniendo el salón, el día y el horario correspondientes a cada índice de la dimensión de la matriz que representaba los salones y horarios.

Taller V - 2002 <i>Asignación de horarios y Salones a Cursos</i>	Tipo de Documento: <i>Informe del Proyecto</i>	Finalizado:
	Versión: 1.0	Aprobado:
Preparado por: IC, JP, YU		Aprobado por:

Aprovechando dichas estructuras auxiliares, y con el objetivo de optimizar el uso de memoria y el tiempo de procesamiento, se decidió cambiar la forma de representar la solución de la siguiente manera:

Se agregaron dos datos al array de clases:

- el índice en el array de Salones, días y horarios en el cual se inicia el dictado de la clase
- la duración de la misma.

De esta forma solamente se almacena por cada clase el índice del salón y horario inicial al que está asignada, y se ahorra el almacenamiento de celdas con ceros para todo el resto de los salones y horarios iniciales posibles. Es claro que almacenar un índice y una duración ocupa más bits que algunos valores booleanos, pero el siguiente cálculo ilustra que la utilización de memoria se reduce considerablemente (en un 99%):

Clases	500
Celdas	500
Bytes por índice	4
Bytes por duración	2
Bytes totales	3000
Kbytes totales	3

Tabla 4-2: Uso de memoria optimizado.

En el Tabú Search, cada vecino se representaría únicamente con el índice del horario inicial al que está asignada cada clase, por lo cual el almacenamiento total para n vecinos se reduce a $2 \times n$ Kbytes.

El tiempo de procesamiento se reduce también considerablemente, dado que para el cálculo de la mayoría de las restricciones, que requerían obtener el horario asignado a cada clase había que recorrer la dimensión de salones, días y horarios hasta encontrar el primer uno. Con la nueva solución, el dato se tiene directamente en el array de clases.

Cambio en la representación de los vecinos

El hecho de representar cada vecino como una solución llevaba a que se debían almacenar n soluciones. Con el objetivo de reducir el costo de almacenamiento, se decidió representar cada solución vecina a través del movimiento que hay que aplicar para obtenerla a partir de la solución actual. Cabe recordar, que un movimiento es cambiar de salón y horario el inicio de una sola clase de la solución.

Con esta nueva representación, pasamos de almacenar simultáneamente $n+1$ soluciones a representar:

- ◆ Una única solución (la actual)
- ◆ Un array de vecinos, cada uno representado por:
 - La clase a mover
 - El nuevo índice de salón y horario a asignarle a la clase

De esta forma, se redujo el tamaño de almacenamiento de cada vecino de 2 Kbytes a 8 bytes.

Taller V - 2002 <i>Asignación de horarios y Salones a Cursos</i>	Tipo de Documento: <i>Informe del Proyecto</i>	Finalizado:
	Versión: 1.0	Aprobado:
Preparado por: IC, JP, YU		Aprobado por:

4.2.2 Restricciones y Preferencias

Consideración de restricciones como preferencias de costo superior

De los trabajos anteriores estudiados se dedujo que si se considera que una solución factible debe satisfacer todas las restricciones es muy difícil encontrarla. Por lo tanto, se decidió considerar las restricciones como preferencias y manejar dos rangos de costos, uno con valores muy altos para las restricciones y otro con valores menores para las preferencias. De esta forma, se evita que el algoritmo quede “estancado” en el espacio de soluciones por no encontrar soluciones factibles vecinas, pero moviéndose hacia regiones en las cuales se cumplan la mayoría de las restricciones.

Cumplimiento implícito

Debido a la implementación, la forma de armado de la solución inicial y la forma de elección de los movimientos, algunas restricciones y preferencias se cumplen para todas las soluciones consideradas, como se detalla a continuación:

Restricciones

Restricción 1: La cantidad de horas asignadas a una clase no puede ser menor a la duración de la clase.

Se cumple por la implementación de la solución. La asignación de una clase se representa por la hora de inicio y su duración, por lo cual no es posible que el algoritmo le asigne menos tiempo del requerido.

Restricción 2: Una clase debe asignarse a un único salón.

Se cumple por la implementación de la solución y por la forma de elección de los movimientos. El array de salones, días y horarios está ordenado por salón, día y horario. La elección de los movimientos controla que el movimiento no deje la clase asignada en dos días diferentes, ajustándose para que la clase entera cambie de día en esos casos. Dado que en situaciones reales se tendrá más de un día, este ajuste implica que tampoco se cambie de salón. Los tipos de movimientos a ajustar se encuentran ejemplificados en el punto Adaptación del Tabú Search a la Asignación de Salones y Horarios. Adicionalmente, al comienzo del algoritmo se crea una solución inicial que ajusta las clases que hayan quedado en dicha situación por modificaciones de los rangos horarios.

Restricción 4: Las clases deben asignarse dentro de los horarios válidos

Se cumple por la implementación de la solución y por la forma de elección de los movimientos. En el array de salones y horarios solamente están representados los horarios válidos, por lo cual es imposible que el algoritmo asigne clases fuera de los mismos. Adicionalmente, al comienzo del algoritmo se crea una solución inicial que ajusta las clases que hayan quedado en dicha situación por modificaciones de los rangos horarios.

Restricción 6: Las horas asignadas a una clase deben ser contiguas.

Se cumple por la misma razón que la restricción 1.

Taller V - 2002 <i>Asignación de horarios y Salones a Cursos</i>	Tipo de Documento: <i>Informe del Proyecto</i>	Finalizado:
	Versión: 1.0	Aprobado:
Preparado por: IC, JP, YU		Aprobado por:

Restricción 7: Una clase debe comenzar y finalizar dentro del mismo día

Se cumple por la misma razón que la restricción 2.

Preferencias

Preferencia 1: Es deseable que no queden clases sin asignar.

Se cumple por la implementación de la solución. La solución inicial que se crea al principio del algoritmo asigna todas las clases, y no se considera como movimiento válido dejar una clase sin asignar.

Preferencia no implementada

La preferencia 5 especifica que es deseable que los teóricos y los prácticos correspondientes se dicten en horarios contiguos. Del análisis de la misma surgió que agregaba demasiada complejidad y tiempo de procesamiento al algoritmo, en comparación con los beneficios que ofrece. Por este motivo se decidió no implementar dicha preferencia.

4.3 Visualización de la solución

En el documento de diseño (Apéndice B) se especificó que la Visualización de una Asignación General se realizaría de la siguiente manera:

La interfaz para la visualización general de una Asignación General se compone de una tabla, en cuyo eje horizontal estarán los días de la semana y en el eje vertical estarán los horarios de las clases.

Cada celda de la tabla corresponde a una pareja día/horario y estará codificada con colores que representan lo siguiente:

- ◆ *Blanco Sin uso, no hay clases asignadas para ese día/horario.*
- ◆ *Verde Sin conflictos, hay clases asignadas y no violan ninguna restricción ni preferencia.*
- ◆ *Amarillo Alguna clase asignada en ese día/horario no cumple con alguna preferencia*
- ◆ *Rojo Alguna clase asignada en ese día/horario no cumple con alguna restricción*

También se pondrá un botón en esta pantalla que nos muestra una lista de las clases que quedaron sin asignar en la Asignación General que está siendo visualizada.

Al presionar doble clic sobre una de las celdas de la tabla, se accederá a una nueva pantalla en la que se muestran con detalle las clases asignadas en el par día/horario representado por la celda. En esta pantalla se mostrará el salón, la asignatura y el grupo correspondiente a cada clase, así como también el detalle de incumplimiento con las restricciones y preferencias, utilizando la misma codificación de colores que la pantalla general de visualización.

A partir de esta especificación se implementó la interfaz de visualización de una asignación general que se observa en la figura 4-2.

Taller V - 2002 <i>Asignación de horarios y Salones a Cursos</i>	Tipo de Documento: <i>Informe del Proyecto</i>	Finalizado:
	Versión: 1.0	Aprobado:
Preparado por: IC, JP, YU		Aprobado por:

Horario	Lunes	Martes	Miércoles	Jueves	Viernes	Sábado
08:00						
08:30						
09:00						
09:30						
10:00						
10:30						
11:00						
11:30						
12:00						
12:30						
13:00						
13:30						
14:00						
14:30						
15:00						
15:30						
16:00						
16:30						
17:00						
17:30						
18:00						
18:30						
19:00						

Figura 4-2: Visualización de una asignación general por horario

Posteriormente, de la utilización de dicha pantalla durante las pruebas, surgió la necesidad de contar con otras formas de visualización, que permitieran el análisis de la asignación según diferentes criterios. Teniendo en cuenta, entonces, que uno de los objetivos del proyecto era proporcionar una herramienta que facilitara entre otras cosas la visualización y modificación de una asignación, se introdujeron otras formas de visualización. En cada una, se agregaron menús desplegables con diferentes acciones que permiten al usuario visualizar el detalle de conflictos y la modificación de la asignación. Finalmente, se implementaron tres criterios de visualización generales, con diferentes filtros de información en cada uno, según el siguiente detalle:

- ◆ Vista por horario
 - Para un salón o todos
- ◆ Vista por asignatura
 - Para una carrera o todas
 - Para un año de las carreras o todos
 - Para una asignatura o todas.
- ◆ Vista por salón

A continuación se presenta una breve descripción e imágenes de las mismas.

4.3.1 Vista por horario

Es la vista original, como se describió en el diseño. Se le agregó únicamente un filtro por salón, el cual permite visualizar la asignación semanal de un solo salón, desplegando además en cada celda la descripción de la clase asignada al día y horario correspondiente a la misma.

El menú desplegable aparecerá sobre las celdas ocupadas, ofreciendo las siguientes opciones:

- ◆ Modificar asignación
- ◆ Borrar asignación
- ◆ Mostrar detalle de conflictos (si los hay)

Taller V - 2002 <i>Asignación de horarios y Salones a Cursos</i>	Tipo de Documento: <i>Informe del Proyecto</i>	Finalizado:
	Versión: 1.0	Aprobado:
Preparado por: IC, JP, YU		Aprobado por:

Visualizar Asignación General

Año: 2002, Segundo Semestre, Desplegar Conflictos:

Por Horario
 Por Asignat...
 Por Salón

Salón: 112, Carrera: Todas, Asignatura: Todas

Cargar, Procesar

Horario	Lunes	Martes	Miércoles	Jueves	Viernes	Sábado
08:00		Pra/TALLER DE P		Pra/TALLER DE P		
08:30		Pra/TALLER DE P		Pra/TALLER DE P		
09:00		Pra/TALLER DE P		Pra/TALLER DE P		
09:30		Pra/TALLER DE P		Pra/TALLER DE P		
10:00		Pra/TALLER DE P		Pra/TALLER DE P		
10:30		Pra/TALLER DE P		Pra/TALLER DE P		
11:00		Teo/RESIDUOS I	Pra/TALLER DE D			
11:30		Teo/RESIDUOS I	Pra/TALLER DE D			
12:00			Pra/TALLER DE D			
12:30			Pra/TALLER DE D			
13:00		Pra/TALLER DE D	Pra/TALLER DE D			
13:30		Pra/ECUACIONE	Pra/TALLER DE D			
14:00		Pra/ECUACIONE				
14:30		Pra/ECUACIONE				
15:00		Pra/TALLER DE D	Teo/MATEMATICA			
15:30		Pra/TALLER DE D	Teo/MATEMATICA			
16:00	Teo/MATEMATICA		Teo/MATEMATICA		Teo/MATEMATICA	
16:30	Teo/MATEMATICA		Teo/MATEMATICA		Teo/MATEMATICA	
17:00	Teo/MATEMATICA		Teo/MATEMATICA		Teo/MATEMATICA	
17:30	Teo/MATEMATICA		Teo/MATEMATICA		Teo/MATEMATICA	
18:00		Teo/METALURGIA	Teo/FISICA 1 (TE		Teo/MAQUINAS E	
18:30		Teo/METALURGIA	Teo/FISICA 1 (TE		Teo/MAQUINAS E	
19:00		Teo/METALURGIA	Teo/FISICA 1 (TE		Teo/MAQUINAS E	

Figura 4-3: Visualización de una asignación general por horario para un salón.

4.3.2 Vista por asignatura

Permite visualizar la asignación semanal de cada grupo dentro de las asignaturas activas, desplegando además en cada celda el horario y salón asignados al día y grupo correspondiente a la misma. Inicialmente se despliega la información de todas las asignaturas, pudiendo el usuario filtrarlas por año, carrera o asignatura.

El menú desplegable aparecerá sobre las celdas ocupadas, ofreciendo las siguientes opciones:

- ◆ Modificar asignación
- ◆ Borrar asignación
- ◆ Mostrar detalle de conflictos (si los hay)

Además, sobre las celdas no ocupadas, se desplegará un menú que permitirá asignar un salón y horario a una clase del grupo correspondiente a la fila.

Taller V - 2002 <i>Asignación de horarios y Salones a Cursos</i>	Tipo de Documento: <i>Informe del Proyecto</i>	Finalizado:
	Versión: 1.0	Aprobado:
Preparado por: IC, JP, YU		Aprobado por:

Visualizar Asignación General

Año: 2002 | Por Horario | Salón: Todos | Año: Todos | Cargar

Segundo Semestre | Por Asignat... | Carrera: Todas

Desplegar Conflictos | Por Salón | Asignatura: Todas | Procesar

Grupo	Lunes	Martes	Miércoles	Jueves	Viernes	Sábado
Pra./PROGRAMACION 3I...				18:00 - 19:30/3		
Pra./PROGRAMACION 3I...		10:00 - 11:30/0				
Pra./PROGRAMACION 3I...				18:00 - 19:30/3		
Teo./PROYECTO/Gr. 1	18:30 - 20:00/0...	20:00 - 22:00/0...	18:00 - 20:00/0...	20:00 - 21:30/0...		
Teo./PROYECTO DE RIE...	08:00 - 10:00/0			08:00 - 10:00/0		
Teo./PROYECTO DE TRA...				09:00 - 11:00/1		
Teo./PROYECTO INDUST...			18:30 - 20:30/1			
Teo./REDES ELECTRICA...		08:00 - 10:00/0		08:00 - 10:00/0		
Teo./RESIDUOS INDUST...		11:00 - 12:00/1		10:00 - 12:00/1		
Teo./RESIDUOS LIQUID...			18:00 - 20:00/0			
Teo./RESISTENCIA DE M...		08:00 - 11:00/1			08:00 - 11:00/05	
Teo./RESISTENCIA DE M...		08:00 - 10:00/1...	11:00 - 12:00/1	08:00 - 10:00/1	09:30 - 10:00/201	
Teo./SERVICIOS Y TECN...	19:30 - 21:30/1				08:00 - 20:00/115	
Teo./SIMULACION A EVE...		10:00 - 12:00/1				
Teo./SISTEMAS DISTRIB...	20:00 - 21:30/2		20:00 - 21:30/2		00:00 - 21:30/201	
Teo./SISTEMAS LINEALE...		08:00 - 10:00/1				
Pra./SISTEMAS LINEALE...	08:00 - 10:00/1					
Pra./SISTEMAS LINEALE...			08:00 - 10:00/1			
Teo./TALLER DE DESAR...		18:30 - 20:30/1		18:00 - 20:00/1		
Teo./TALLER DE DISEÑO...				11:00 - 12:00/2		
Teo./TALLER DE DISEÑO...				15:00 - 16:00/2		
Pra./TALLER DE DISEÑO...				08:00 - 11:00/2		
Pra./TALLER DE DISEÑO...			18:00 - 19:00/2			

Figura 4-4: Visualización de una asignación general por asignatura

4.3.3 Vista por salón

Permite visualizar la asignación semanal de todos los salones.

El menú desplegable aparecerá sobre las celdas ocupadas, ofreciendo la opción de Mostrar detalle de conflictos (si los hay).

Visualizar Asignación General

Año: 2002 | Por Horario | Salón: Todos | Año: Todos | Cargar

Segundo Semestre | Por Asignat... | Carrera: Todas

Desplegar Conflictos | Por Salón | Asignatura: Todas | Procesar

Salón	Lunes	Martes	Miércoles	Jueves	Viernes	Sábado
000						
001						
002						
006						
007						
008						
009						
010						
011						
031						
101						
103						
105						
107						
108						
109						
110						
111						
112						
113						
115						
201						
202						

Taller V - 2002 <i>Asignación de horarios y Salones a Cursos</i>	Tipo de Documento: <i>Informe del Proyecto</i>	Finalizado:
	Versión: 1.0	Aprobado:
Preparado por: IC, JP, YU		Aprobado por:

Figura 4-5: Visualización de una asignación general por salón.

5 Calibración del Tabu Search

Para lograr una performance óptima del Tabú Search, es necesario encontrar los valores adecuados para sus parámetros, así como también para los pesos correspondientes a cada restricción y preferencia; dichos pesos intervienen en el cálculo del costo de una solución, o sea el valor de la función objetivo. Al proceso de encontrar estos valores se le denomina calibración.

Los principales parámetros de este método son:

- ◆ k1 y k2: largo de las dos Tabú-list que intervienen.
- ◆ La cardinalidad del conjunto V de soluciones vecinas testeadas en cada iteración
- ◆ Tmax: el máximo de iteraciones sin conseguir mejorar el valor de la función objetivo.

Se realizaron distintas pruebas para lograr este objetivo, y analizando los resultados de las mismas, se definieron los valores que se consideraron óptimos para el volumen de datos con las cuales se llevaron a cabo las pruebas. Los datos que se utilizaron son datos reales obtenidos de la situación actual de la asignación de salones y horarios en la Facultad de Ingeniería, utilizando también en algunas pruebas como punto de partida la asignación del segundo semestre del año 2002.

Para cada restricción y preferencia que interviene en el valor de la función objetivo, también es necesario determinar los costos adecuados, aunque esto no depende del volumen de datos con que se este trabajando, sino que depende de cuales restricciones o preferencias se quieren priorizar en la búsqueda de una determinada solución; por lo tanto los valores elegidos para los pesos de cada una de ellas no dependen del análisis de los resultados de las pruebas sino de los conflictos que se quieren priorizar. Los valores finales fueron determinados en conjunto con los funcionarios de bedelía que intervinieron en el desarrollo de este proyecto, y estos mismos valores pueden ser cambiados en la base de datos si en algún momento se desea aumentar o disminuir el costo de alguna restricción o preferencia por cualquier motivo.

Para una descripción detallada de las pruebas realizadas y de la calibración del Tabú Search, referirse al Anexo 5, sección 5.1.

5.1.1 Resumen de Calibración del Tabu Search

Luego de haber ejecutado todas las pruebas y de haber analizado los resultados de las mismas, llegamos a determinar los siguientes valores para los parámetros que intervienen en el Tabú Search. Cabe acotar, que dichos valores fueron utilizados en reiteradas oportunidades durante las pruebas del programa, observándose un comportamiento favorable del algoritmo.

Parámetro	Valor Optimo
Cantidad de Soluciones Vecinas	25
Largo Tabú List Clases	2
Largo Tabú List Clase/Salón/Horario	10
Cantidad de Iteraciones sin Mejora	8000

Tabla 5-1: Valores determinados por la calibración del método.

Taller V - 2002 <i>Asignación de horarios y Salones a Cursos</i>	Tipo de Documento: <i>Informe del Proyecto</i>	Finalizado:
	Versión: <i>1.0</i>	Aprobado:
Preparado por: <i>IC, JP, YU</i>		Aprobado por:

Los pesos utilizados para las restricciones y preferencias se pueden encontrar en el Anexo 5 Sección 5.2.

Taller V - 2002 <i>Asignación de horarios y Salones a Cursos</i>	Tipo de Documento: <i>Informe del Proyecto</i>	Finalizado:
	Versión: 1.0	Aprobado:
Preparado por: IC, JP, YU		Aprobado por:

6 Conclusiones

Luego de realizado el desarrollo, testeo y puesta en producción de la herramienta implementada, las principales conclusiones a las que llegamos son las siguientes:

♦ **Solución Manual vs. Solución del Tabú Search**

Uno de los objetivos principales del proyecto era la construcción de una herramienta de apoyo a los funcionarios de bedelía que generara, en forma automática, la asignación de salones y horarios en un tiempo razonable. Luego de efectuadas y analizadas las pruebas, basta con mirar los resultados obtenidos para ver claramente que las soluciones encontradas por el Tabú Search son mejores en costo que las soluciones manuales.

Esto resulta bastante lógico, ya que dado el volumen de datos involucrado y la cantidad de restricciones a tener en cuenta es prácticamente imposible construir una solución en forma manual que contemple los posibles conflictos generados por las asignaciones de clase.

También se observa una ventaja en el tiempo que lleva construir estas soluciones, ya que para la solución manual el promedio de su confección es aproximadamente de un mes y medio, mientras que con el Tabú Search es cuestión de horas.

Creemos, con los resultados a la vista, que la solución encontrada por el algoritmo supera ampliamente los costos de las soluciones manuales. Claro que las soluciones generadas por el Tabú Search no son la solución ideal al problema, pero pensamos que proporcionan una buena aproximación, que con un poco de trabajo manual puede llevarse a una asignación de salones eficiente.

♦ **Elección de la Solución Inicial**

Como se ha expresado anteriormente, el algoritmo de optimización tiene la capacidad de partir de cualquier solución inicial (en nuestras pruebas estas fueron la del 2do. semestre del 2002 y una generada en forma aleatoria).

Observando los resultados de las pruebas realizadas, claramente los costos a los que se llega utilizando como solución inicial la del 2do. semestre del 2002 son menores que los que se obtienen partiendo de una solución aleatoria. Este resultado concuerda con las teorías que sustentan al algoritmo utilizado, las cuales expresan que el mismo logra mejores resultados si el punto de partida está más cerca de las regiones de costos óptimos del espacio de soluciones.

Por lo tanto, creemos que es mucho mejor utilizar como solución inicial del proceso de optimización alguna solución de los semestres de años anteriores, ya que esto hace que se obtengan mejores costos. Además utilizando estas soluciones, se puede lograr cierta coherencia entre las asignaciones de años anteriores y la que se está buscando para el semestre actual, marcando como fijas las asignaciones de clase que se desee mantener incambiables.

Taller V - 2002 <i>Asignación de horarios y Salones a Cursos</i>	Tipo de Documento: <i>Informe del Proyecto</i>	Finalizado:
	Versión: 1.0	Aprobado:
Preparado por: IC, JP, YU		Aprobado por:

◆ **Visualización y Manipulación de Datos**

Uno de los problemas a los que se enfrenta quien está confeccionando la asignación de los salones y horarios a las clases, es que el gran volumen de datos involucrados dificulta tener una visión global del estado de la misma.

Con el producto desarrollado se ha logrado brindar herramientas que facilitan la visualización del estado de las asignaciones utilizando diferentes criterios de ordenamiento y filtrado de los datos, que permiten ver la información desde diferentes ángulos. Las mismas incluyen además, una codificación de colores que permite fácilmente al usuario identificar los conflictos que deben resolverse y la gravedad de los mismos.

Algunas de estas formas de visualización ofrecen, además, la posibilidad de realizar desde la misma pantalla modificaciones a las asignaciones, refrescando automáticamente la visualización de la influencia de dichas modificaciones en el estado de la asignación.

◆ **Portabilidad**

Una de las principales características que se tuvo en cuenta en el momento de elegir la plataforma a utilizar fue la portabilidad, hecho determinado por la actual situación económica que enfrenta la Universidad toda, a la cual no escapa nuestra Facultad.

Por lo tanto, si bien en este momento el sistema operativo utilizado por los funcionarios de bedelía es una versión de Windows, se tuvo en cuenta la posible migración a Linux y/o cualquier sistema operativo basado en Unix que fuera de libre comercialización. Estos factores fueron predominantes a la hora de elegir la combinación Java/MySQL como la plataforma de desarrollo a utilizar para el sistema de asignación.

El sistema fue probado en varias versiones de Windows (95, 98, NT, 2000), así como también en varias implementaciones de Linux (Red Hat, Suse). En todas estas plataformas el sistema se comporta en forma estable con respecto a su funcionalidad, inclusive en lo que respecta a su interfase gráfica, de manera que consideramos que la portabilidad del mismo es muy buena. Por lo tanto, concluimos que cumple con el deseo de brindar la funcionalidad requerida independientemente del sistema operativo elegido por bedelía para operar, el cual suponemos va a ser alguno de los testeados, tanto ahora como en el futuro.

◆ **Uso Efectivo en Bedelía**

Sin duda la conclusión más importante de este proyecto, y de la cual nos sentimos más satisfechos, es el hecho de que los funcionarios de bedelía han expresado su conformidad con el producto desarrollado, así como su compromiso de utilizarlo de forma efectiva para la futura confección de las asignaciones de la Facultad.

El producto fue instalado en la bedelía de la Facultad de Ingeniería, los funcionarios encargados de la asignación fueron capacitados para utilizarlo y han expresado su satisfacción debido a que la aplicación les brinda una herramienta de apoyo efectiva, tanto para efectuar la asignación de los salones como para mantener los datos relacionados con la misma.

Este hecho constituye la mejor confirmación de que el proyecto desarrollado realmente cumple los objetivos trazados inicialmente, dejando puertas abiertas para su posterior continuación y mejoramiento, tal cual se expresa en el siguiente punto de este documento.

Taller V - 2002 <i>Asignación de horarios y Salones a Cursos</i>	Tipo de Documento: <i>Informe del Proyecto</i>	Finalizado:
	Versión: 1.0	Aprobado:
Preparado por: IC, JP, YU		Aprobado por:

7 Trabajos Futuros

Pensamos que el proyecto llevado a cabo puede servir como punto de partida para varios emprendimientos que lo extiendan y lo mejoren. Planteamos aquí algunas de las ideas que se podrían realizar sobre el mismo:

◆ **Implementación de Otras Heurísticas**

Tal como se ha expresado anteriormente, existen varias heurísticas que se pueden aplicar para la resolución de problemas de asignación. Algunas de ellas (Algoritmos Genéticos, Ant Systems, Simulated Annealing) han sido comentadas en este mismo documento e inclusive han sido utilizadas ya para resolver el problema de asignación de salones y horarios en estudios anteriores.

Uno de los posibles trabajos futuros puede ser el de implementar alguna de estas heurísticas para resolver el problema de asignación de salones y horarios en la Facultad de Ingeniería. Esto brindaría un marco de comparación entre los resultados obtenidos por los distintos métodos y ampliaría las opciones del usuario, ya que podría elegir el método que mejor se adapte a sus necesidades en el momento de construir una asignación en el futuro.

La implementación de otra heurística es muy factible en este caso, ya que los datos necesarios ya se encuentran disponibles en este producto, y debido al uso de orientación a objetos en su implementación, lo único que hay que cambiar en el mismo es una clase, que es la que implementa el Tabú Search, sustituyéndola por otra clase que implemente cualquier otro de los algoritmos.

Como sugerencia, creemos que sería interesante implementar el algoritmo de optimización utilizando Algoritmos Genéticos, ya que según lo que hemos visto en la etapa de planificación y elección del método a utilizar, este se adapta bien para este tipo de problemas.

◆ **Ampliación de las Restricciones y Preferencias**

Las restricciones y preferencias utilizadas para calcular el costo de la función objetivo del Tabú Search fueron definidas conjuntamente con los funcionarios de bedelía y responden a las necesidades actuales de una asignación, priorizando algunas de ellas sobre otras.

Seguramente existan muchas restricciones que una asignación debería cumplir, aparte de las ya implementadas, pero forzar el cumplimiento de muchas de ellas resultaría poco práctico ya que estaríamos haciendo que el método pierda flexibilidad y no pueda encontrar soluciones factibles. También podría pasar que en un futuro algunas restricciones que hoy en día no son importantes pasen a tener un rol protagónico en la factibilidad de una asignación, hecho que sería suficiente para que su implementación sea totalmente necesaria.

Sin embargo, pensamos que algunas de las siguientes restricciones se podrían implementar sin influir demasiado sobre la performance del Tabú Search, o quizás su implementación resultaría más sencilla si se utilizara otro algoritmo para optimizar la asignación, tal cual lo expresado en el punto anterior.

Taller V - 2002 <i>Asignación de horarios y Salones a Cursos</i>	Tipo de Documento: <i>Informe del Proyecto</i>	Finalizado:
	Versión: 1.0	Aprobado:
Preparado por: IC, JP, YU		Aprobado por:

Las restricciones que se podrían implementar son:

Distribución geográfica

Se podría minimizar el traslado masivo de alumnos de un salón a otro. Para esto se deben tener en cuenta datos geográficos, de distancia entre salones y horarios del día en donde hay un mayor tráfico de alumnos. Si bien no es una restricción de fácil implementación, creemos que sería muy útil, con un alto margen de costo/beneficio.

Puentes

Lograr que no existan horas puente para los alumnos es algo que, según nos expresaron los funcionarios de bedelía, sería muy beneficioso ya que, por ejemplo, es muy común el hecho de que un alumno tenga algunas clases de mañana y otras de tarde. Con respecto a su implementación, parece bastante sencilla con el esquema actual del producto.

Horarios de los Profesores

Tener en cuenta los horarios de los profesores al momento de hacer la asignación de clases. En este momento las preferencias de los profesores se pueden tener en cuenta en forma indirecta, ya que se le pueden asignar preferencias a las clases, simplemente habría que asignar a una clase el horario de preferencia del profesor que la va a dictar.

◆ **Extensión para Exámenes y Parciales**

Se podría extender la funcionalidad de asignación de la aplicación para tener en cuenta los exámenes y/o parciales que se rinden en la Facultad.

La asignación de exámenes y parciales es un problema muy similar al tratado en este proyecto, por lo que en principio extenderlo para que los tenga en cuenta no sería muy difícil. La mayor dificultad radica en que no hay que considerar una asignación semanal sino una asignación dentro de un período de tiempo dado (por ejemplo, los períodos de exámenes actuales duran 3 semanas). Otra dificultad que puede surgir, es que quizás la función objetivo de un método que trate de hacer este tipo de asignaciones deba tener en cuenta algunas restricciones que hoy no se encuentran representadas o no tener en cuenta algunas de las que ya están implementadas.

Muchos de los datos necesarios para llevar a cabo la asignación de exámenes o parciales ya se encuentran en la aplicación, como por ejemplo los datos de inscriptos a las asignaturas o los rangos horarios válidos. En consecuencia, extender la aplicación para que tenga esta funcionalidad parece uno de los pasos más lógicos a dar como futura extensión de la aplicación.

Taller V - 2002 <i>Asignación de horarios y Salones a Cursos</i>	Tipo de Documento: <i>Informe del Proyecto</i>	Finalizado:
	Versión: <i>1.0</i>	Aprobado:
Preparado por: <i>IC, JP, YU</i>		Aprobado por:

♦ **Aplicación Multiusuario / Interfase Web**

El producto desarrollado es monousuario, es decir que no puede ser usado por más de un usuario a la vez. Si bien en la actualidad el encargado de la asignación es un solo funcionario, esta situación puede cambiar en el futuro. Lo cual haría necesario que la aplicación de asignación de salones sea transformada en una aplicación multiusuario.

Creemos que la forma más adecuada de llevar la aplicación actual a un sistema multiusuario es utilizando una interfase web, accesible desde browsers. La aplicación está hecha en Java, o sea que las clases que implementan la lógica de la implementación se pueden mantener incambiadas reconstruyendo las interfaces con el usuario para que éstas sean páginas html aptas para visualizarse desde browsers.

Las ventajas de un sistema multiusuario son obvias, aunque seguramente haya que tener en cuenta muchas situaciones que pueden darse en un entorno concurrente, con varios usuarios realizando modificaciones a la asignación de un mismo semestre. La base de datos utilizada actualmente (MySQL) soporta transacciones o sea que ya se tendrían las herramientas básicas para mantener la integridad de los datos en el caso de que el sistema pase a ser multiusuario.

Taller V - 2002 <i>Asignación de horarios y Salones a Cursos</i>	Tipo de Documento: <i>Informe del Proyecto</i>	Finalizado:
	Versión: 1.0	Aprobado:
Preparado por: IC, JP, YU		Aprobado por:

Referencias

- [1] A. Colomi, M. Dorigo, F. Maffioli, V. Maniezzo, G. Righini, M. Trubian. (1996) **Heuristics from Nature for Hard Combinatorial Optimization Problems**. IFORS. Published by Elsevier Science Ltd.
- [2] A. Hertz (1991) **Tabú Search for Large Scale Timetabling Problems**. European Journal of Operational Research 54 (1991) 39-47.
- [3] A. Schaerf (1996) **Tabú Search Techniques for Large High-School Timetabling Problems**. Centrum voor Wiskunfe en Informatica Cs-R9611.
- [4] A. Herz, E.Taillard, D. De Werra **A Tutorial on Tabú Search**.
- [5] F. Glover (1989) **Tabú Search, Part I**. ORSA Journal on Computing, (1989) 190-206.
- [6] F. Glover (1990) **Tabú Search, Part II**. ORSA Journal on Computing, (1990) 4-32.
- [7] M. Zachariasen (2001) **Memory-Based Metaheuristics Tabú Search and Guided Local Search**. DIKU (2001).
- [8] A. Colomi, M. Dorigo, V. Maniezzo **Metaheuristics for High-School Timetabling**. Computational Optimization and Applications Journal.
- [9] A. Abramson (1991) **Construction School Timetables Using Simulated Annealing: Sequential and Parallel Algorithms**. Management Science 37 (1991), 98-113.
- [10] A. Colomi, M. Dorigo, V. Maniezzo **A Genetic Algorithm to Solve the Timetable Problem**. Politecnico di Milano. Technical Report N° 90-060.
- [11] W. Erben, J. Keppler (1995) **A Genetic Algorithm Solving a Weekly Course-Timetabling Problem**. Department of Computer Science, Fachhochschule Konstanz.
- [12] F. Rey, M. Carlevaro (1998) **Problema de Asignación de Horarios y Salones a Cursos**. Proyecto de Grado, Facultad de Ingeniería.
- [13] T. Cooper, J. Kingston **The Complexity of Timetable Construction Problems**. Basser Department of Computer Science, University of Sydney.
- [14] A. Hertz (1992) **Finding a Feasible Course Schedule Using Tabú Search**. DiscreteApplied Mathematics 35 (1992) 255-270.
- [15] D. Costa (1994) **A Tabú search Algorithm for computing an operational Timetable**. European Journal of Operational Research 76 (1994), 98-110.

Taller V - 2002 <i>Asignación de horarios y Salones a Cursos</i>	Tipo de Documento: <i>Informe del Proyecto</i>	Finalizado:
	Versión: 1.0	Aprobado:
Preparado por: IC, JP, YU		Aprobado por:

Anexo 1. Organización de la Facultad

1.1 Carreras

Las carreras se consideran principalmente en el momento de la asignación para evitar las superposiciones de asignaturas de un mismo semestre y carrera, y para filtrar los datos que se presentarán al usuario en algunos Listados.

Las características a tener en cuenta de las carreras son:

- ◆ Código y Ciclo
- ◆ Nombre
- ◆ Cantidad de años

1.2 Años y Semestres

Los años y semestres serán considerados como atributos de las carreras y asignaturas. Aunque sean componentes de la estructura real de la Facultad, no tienen características propias que debamos considerar.

1.3 Asignaturas

Una asignatura puede pertenecer a más de una carrera y no necesariamente al mismo año dentro de ellas, pudiendo además tener nombres y códigos diferentes dependiendo de la carrera. Este es un punto a tener en cuenta al ingresar los datos, para no realizar varias asignaciones de la misma asignatura. Cada conjunto de asignaturas equivalentes tendrá una asignatura primaria, que es la única que se considerará en representación del conjunto en la asignación.

Las asignaturas tienen diferentes niveles de importancia, los cuales determinan prioridades a la hora de establecer la asignación. Por ejemplo, las asignaturas del básico tienen mayor prioridad que las de otras carreras, y dentro de una misma carrera son más importantes las asignaturas obligatorias que las electivas.

Generalmente se tienen estimados para cada asignatura los porcentajes de abandono y recursantes, dentro de los alumnos inscriptos. Es importante que, sobre todo si la cantidad de inscriptos por asignatura se ingresa automáticamente desde otra aplicación, se puedan establecer estos porcentajes, y sean tenidos en cuenta para realizar asignaciones aunque las capacidades de los salones no cubra la cantidad de inscriptos, sino que se considere suficiente que cubra la cantidad estimada de alumnos que van a concurrir a las clases durante todo el semestre.

Cada asignatura puede estar dividida en varios grupos, dependiendo de la cantidad de alumnos y docentes, para aquellas asignaturas que no se dividan en grupos, se considerará que tiene un solo grupo.

Taller V - 2002 <i>Asignación de horarios y Salones a Cursos</i>	Tipo de Documento: <i>Informe del Proyecto</i>	Finalizado:
	Versión: 1.0	Aprobado:
Preparado por: IC, JP, YU		Aprobado por:

Es deseable que los teóricos y prácticos de las asignaturas tengan horarios contiguos, por lo cual se debe manejar la relación entre los cursos teóricos y prácticos de las asignaturas.

Se definen tres estados posibles para las asignaturas:

- ◆ Cargada: Asignatura ingresada automáticamente a partir de los datos obtenidos del sistema central (SECIU), cuyos datos están incompletos por lo que no puede participar en una asignación.
- ◆ Activa: Asignatura para la cual se dictan clases en forma regular, esta en condiciones de ser utilizada en el proceso de asignación
- ◆ Inactiva: Asignatura para la cual no se dictan clases actualmente, no debe considerarse en el proceso de asignación

Las asignaturas tienen distintos niveles de importancia, dependiendo de que pertenezcan al ciclo básico, el cual es común para muchas carreras de la facultad, que sean obligatorias o electivas. Por lo tanto se definen los siguientes niveles de importancia, en orden descendente:

- ◆ **Asignaturas Básicas**
Asignaturas pertenecientes al ciclo básico de la Facultad de Ingeniería, deben tener precedencia al momento de asignarlas.
- ◆ **Asignaturas Obligatorias**
Asignaturas que son obligatorias para las carreras a las cuales pertenece.
- ◆ **Asignaturas Electivas**
Asignaturas electivas para alguna o todas las carreras a las cuales pertenece, su importancia dentro de la asignación es menor a la de las obligatorias.

Debido a que las asignaturas pueden pertenecer a más de una carrera, el criterio de importancia es una característica de la relación entre una asignatura y una carrera, ya que una misma asignatura puede ser obligatoria para una carrera y electiva para otra.

En resumen, las características a tener en cuenta de las asignaturas son:

- ◆ Código
- ◆ Nombre
- ◆ Carreras a las que pertenece
- ◆ Año y semestre dentro de cada carrera
- ◆ Nivel de importancia dentro de cada carrera
- ◆ Anual o Semestral
- ◆ Asignatura primaria
- ◆ Cantidad de grupos teóricos
- ◆ Cantidad de grupos prácticos
- ◆ Cantidad de clases teóricas por semana
- ◆ Cantidad de clases prácticas por semana
- ◆ Estado

Taller V - 2002 <i>Asignación de horarios y Salones a Cursos</i>	Tipo de Documento: <i>Informe del Proyecto</i>	Finalizado:
	Versión: 1.0	Aprobado:
Preparado por: IC, JP, YU		Aprobado por:

1.4 Bloques

Debido a la gran cantidad de estudiantes que ingresan en primer año, se decidió particionar los rangos de dictado de clase en bloques o turnos. De esta forma tenemos bloques horarios dentro de los cuales hay que asignar las clases de los distintos grupos de las asignaturas con muchos alumnos (generalmente las del ciclo básico).

Para ser coherentes con esta organización en bloques de los horarios, se sugiere definir bloques para la mañana, tarde y noche y es necesaria la existencia de un bloque especial que puede denominarse "bloque total" que comprenda todo el rango horario de dictado de clases. Aquellas asignaturas que no tengan preferencias horarias se asignarán al bloque total, de esta forma se pueden asignar sus clases en cualquier momento del día, sin ninguna restricción de horario.

Las características principales a tener en cuenta de los bloques son:

- ◆ Hora Inicio
- ◆ Hora Fin
- ◆ Nombre

1.5 Grupos

Las asignaturas pueden organizarse en grupos, entre los cuales se distribuirá la cantidad de alumnos inscriptos a las mismas. Los grupos pueden ser teóricos o prácticos, y cada asignatura deberá tener al menos un grupo teórico.

A cada grupo se le deberá especificar un bloque, que determinará el rango de horarios dentro del cual deberán ser asignadas sus clases.

Es importante que se especifique correctamente la cantidad de alumnos que tiene cada grupo, ya que de aquí es de donde se tomará la información para realizar la asignación automática.

Las características a considerar de los grupos son:

- ◆ Asignatura a la cual pertenece
- ◆ Tipo de clases (teóricas o prácticas)
- ◆ Bloque
- ◆ Cantidad de estudiantes

1.6 Clases

Las clases son la unidad básica a asignar a los salones y horarios. Cada clase tendrá una determinada duración, días y horario de preferencia. La cantidad de alumnos que asistirán a una clase quedará determinada por la cantidad de alumnos asignados al grupo al cual pertenece.

Existen asignaciones de clase que deben ser fijas, es decir que el proceso de búsqueda de una asignación no puede modificar (por ejemplo las clases dictadas en algunos Institutos, que ya vienen prefijadas en su horario y salón).

Una clase puede necesitar que el salón en que se dicta disponga de cierta infraestructura, como por ejemplo enchufes, pantallas o cortinas.

Taller V - 2002 <i>Asignación de horarios y Salones a Cursos</i>	Tipo de Documento: <i>Informe del Proyecto</i>	Finalizado:
	Versión: 1.0	Aprobado:
Preparado por: IC, JP, YU		Aprobado por:

Las características a considerar de las clases son:

- ◆ Asignatura
- ◆ Grupo
- ◆ Duración
- ◆ Días y horas de preferencia
- ◆ Recursos de infraestructura necesarios

1.7 Salones

La principal característica a tener en cuenta de los salones es su capacidad locativa, siendo éste un dato que se actualiza semestre a semestre. También se tendrán en cuenta las características de infraestructura (recursos) mencionados anteriormente, y datos informativos como el nombre y la ubicación.

Como se expresó anteriormente, hay asignaciones de clases a horarios que son fijas, por lo cual no deben modificarse en el proceso de asignación. Del mismo modo, existen salones que no se pueden utilizar libremente, y no se pueden tomar en cuenta en la asignación automática.

Se definen tres estados posibles para los salones:

- ◆ **Asignación libre:**
Salón en uso, disponible para el proceso automático de asignación
- ◆ **Asignación limitada:**
Salón en uso, solo utilizable en asignaciones manuales, por ejemplo los salones pertenecientes a los distintos Institutos, los cuales son asignados por los Institutos mismos.
- ◆ **No disponible**
Salón en desuso, no disponible para el proceso automático de asignación, ni para asignaciones manuales.

Las características a considerar de los salones son:

- ◆ Nombre
- ◆ Ubicación
- ◆ Capacidad
- ◆ Recursos
- ◆ Estado

1.8 Recursos

Existen clases que pueden requerir de cierta infraestructura para que su dictado sea posible o mejorable, llamaremos recursos a esta infraestructura necesaria. Tomemos como ejemplo proyectores, pizarrones especiales, cortinas, etc.

Las características de los recursos son:

- ◆ Nombre
- ◆ Descripción

Taller V - 2002 <i>Asignación de horarios y Salones a Cursos</i>	Tipo de Documento: <i>Informe del Proyecto</i>	Finalizado:
	Versión: 1.0	Aprobado:
Preparado por: IC, JP, YU		Aprobado por:

1.9 Horarios de dictado de clases

Actualmente las clases se dictan desde las 8:00 a las 23:00 horas de Lunes a Viernes, pero esto puede cambiar, por lo cual los rangos horarios a asignar deberán ser parametrizables.

Se asumirá que no hay clases que duren menos de media hora, por lo tanto esa será la unidad horaria a asignar y la duración de las clases se expresará en múltiplos de la misma.

1.10 Reserva de Salones

Durante un semestre puede surgir la necesidad de dictar alguna clase especial, recuperar clases perdidas por diversas causas (huelgas, apagones, etc.), o de realizar otra clase de eventos (asambleas, conferencias, etc.). Por este motivo se hace necesaria una herramienta para poder utilizar salones por fuera de la asignación semanal que se construye para todo el semestre, a esta funcionalidad se le llama reserva de salones.

Se debe tener en cuenta al reservar un salón un cierto día y horario que allí no haya una asignación de clase ni una reserva anterior.

Taller V - 2002 <i>Asignación de horarios y Salones a Cursos</i>	Tipo de Documento: <i>Informe del Proyecto</i>	Finalizado:
	Versión: 1.0	Aprobado:
Preparado por: IC, JP, YU		Aprobado por:

Anexo 2. Formalización del Problema

2.1 Presentación del Problema Planteado Como Un Problema De Asignación

En este punto se resumirán los planteamientos generales de los problemas de optimización y de asignación, viendo como el problema a resolver se asocia con los mismos, por lo que es posible verlo como un caso particular de ellos.

2.1.1 Planteo General de un Problema de Optimización

En su forma más general, un problema de optimización consiste en encontrar un elemento x perteneciente a un dominio D de manera de minimizar el valor de una función f .

Utilizando notación matemática, se podría ver como:

$$\text{Min}(f(x)) / x \in D$$

donde D es el dominio considerado al que se denominará región factible y f es la llamada función objetivo.

Restringiéndose únicamente a los problemas de programación matemática, se puede considerar que:

$$\begin{aligned} x &\in \mathbb{R}^n \\ f : \mathbb{R}^n &\rightarrow \mathbb{R} \\ D &\subseteq \mathbb{R}^n \end{aligned}$$

La región D esta delimitada por las restricciones del problema. Esto mismo puede expresarse como los elementos x que cumplen con cierto conjunto de inecuaciones:

$$D = \{ x / g(x) \leq 0 \} \text{ con } x \in \mathbb{R}^n \text{ y } g: \mathbb{R}^n \rightarrow \mathbb{R}^m$$

$$\text{con } g = (g_1, g_2, \dots, g_m) \text{ y } x = (x_1, x_2, \dots, x_n)$$

y el conjunto de inecuaciones expresado de la siguiente manera:

$$\begin{aligned} g_1(x_1, x_2, \dots, x_n) &\leq 0 \\ g_2(x_1, x_2, \dots, x_n) &\leq 0 \\ \dots & \\ g_m(x_1, x_2, \dots, x_n) &\leq 0 \end{aligned}$$

Las variables x_i se denominan variables independientes o variables de decisión.

Taller V - 2002 <i>Asignación de horarios y Salones a Cursos</i>	Tipo de Documento: <i>Informe del Proyecto</i>	Finalizado:
	Versión: 1.0	Aprobado:
Preparado por: IC, JP, YU		Aprobado por:

2.1.2 Planteo General de un Problema de Asignación

El planteo general de un problema de asignación (ATP – Assignment Type Problem) es el siguiente:

Dados n ítems y m recursos, el ATP consiste en determinar una asignación de los recursos a los ítems de tal forma de optimizar una función objetivo f y satisfaciendo un conjunto R de restricciones.

$$\begin{aligned} &\text{Hallar } \min(f(x)) \\ &\text{sujeto a } \sum_{j \in J_i} x_{i,j} = 1 \quad 1 \leq i \leq n \end{aligned} \quad (1)$$

$$G_r(x) \leq 0 \quad 1 \leq r \leq R \quad (2)$$

$$x_{i,j} \in \{0,1\} \quad 1 \leq i \leq n \quad j \in J_i \quad (3)$$

dónde $x_{i,j}$ son las variables de decisión

con $x_{i,j} = 1$ si el recurso j es asignado al ítem i

$x_{i,j} = 0$ sino

$J_i \subseteq \{1,2,\dots,m\}$ es el conjunto de recursos admitidos para el ítem i

Los puntos (1) y (3) implican que cada ítem i debe tener asignado exactamente un recurso j .

La función objetivo f y las restricciones R_i no necesitan cumplir ninguna propiedad en particular.

2.1.3 Planteo del Problema como un Problema de Optimización

El problema de Asignación de Cursos y Salones (en inglés se le denomina como Timetable Problem) se puede plantear como un problema de optimización combinatoria, y más específicamente como un problema de asignación.

Un problema de optimización combinatoria se especifica como un conjunto de instancias de problema. Una instancia se define como un par (S,f) donde S es un conjunto finito de soluciones factibles, llamado espacio de soluciones y f una función de costo $f: S \rightarrow R$.

El valor óptimo de f es:

$$f_0 = \min\{f(s) / s \in S\}$$

y el conjunto de soluciones óptimas es:

$$S_0 = \{s \in S / f(s) = f_0\}$$

El objetivo del problema a resolver es entonces encontrar un elemento s^* del conjunto S_0 .

Taller V - 2002 <i>Asignación de horarios y Salones a Cursos</i>	Tipo de Documento: <i>Informe del Proyecto</i>	Finalizado:
	Versión: 1.0	Aprobado:
Preparado por: IC, JP, YU		Aprobado por:

Para que una asignación sea una solución válida al problema, esta debe cumplir con ciertas restricciones. Obviamente, dentro de las restricciones, hay algunas más importantes que otras, por lo tanto, aquellas restricciones de menor relevancia serán denominadas preferencias.

Las preferencias son restricciones de menor orden, y no se considerará que una asignación no es una solución por no cumplir una preferencia. La utilidad que aportan las preferencias es el poder comparar soluciones, ya que una solución es “mejor” cuanto mayor cantidad de preferencias cumpla.

En el problema planteado, la función objetivo f va a indicar el grado de inaceptabilidad de una determinada asignación, es decir cuanto se aleja la asignación considerada de una asignación ideal en la cual se cumplen todas las restricciones y se satisfacen todas las preferencias. Expresado esto, definiremos que una solución s_1 es mejor que una s_2 cuando el valor de la función objetivo f es menor para s_1 que para s_2 .

Dicho problema puede identificarse fácilmente como un problema de asignación ya que se trata de asignar salones y horarios a clases. La diferencia principal es que consta de tres dimensiones básicas (salones, horarios y clases) en lugar de dos (ítems y recursos). Las tres dimensiones pueden reducirse a dos agrupando dos de ellas, o sea que se podrán agrupar salones y horarios y asignar éstos a las clases.

A continuación se recordarán las restricciones y preferencias definidas previamente y se definirá una función objetivo para problema de asignación de cursos y salones.

Restricciones

1. La cantidad de horas asignadas a una clase no puede ser menor a la duración de la clase.
2. Una clase debe asignarse a un único salón.
3. Una pareja (horario, salón) debe asignarse a una única clase.
4. Las clases deben asignarse dentro de los horarios válidos.
5. Las clases de un mismo semestre, carrera y grupo no pueden superponerse. Esta restricción no aplica a las asignaturas electivas, y pueden superponerse entre sí clases que correspondan a asignaturas con más de un grupo.
6. Las horas asignadas a una clase deben ser contiguas.
7. Las horas asignadas a una clase deben estar dentro de un mismo día.
8. Los grupos que tienen un bloque especificado sólo pueden asignarse dentro de la franja horaria correspondiente al bloque.
9. No pueden dictarse dos clases de un mismo grupo y asignatura un mismo día.

Preferencias

1. Es deseable que no queden clases sin asignar.
2. Las clases deberían asignarse dentro del rango horario de preferencia especificado.
3. Una clase no debería asignarse a un salón con menor capacidad de la requerida.
4. Es deseable que el salón asignado a una clase cuente con los recursos necesarios para el dictado de la misma. Por ej., si la clase necesita enchufe para un retroproyector, el salón debe tener enchufes.
5. Es deseable que los teóricos y los prácticos correspondientes se dicten en horarios contiguos.
6. No sería bueno que se utilizaran salones de gran capacidad para clases de pocos alumnos.
7. Es conveniente que las clases teóricas de un grupo se repartan en forma uniforme en la semana.
8. Similitud de horarios en las distintas clases teóricas de un grupo.

Taller V - 2002 <i>Asignación de horarios y Salones a Cursos</i>	Tipo de Documento: <i>Informe del Proyecto</i>	Finalizado:
	Versión: 1.0	Aprobado:
Preparado por: IC, JP, YU		Aprobado por:

Función Objetivo

Para evaluar el grado de aceptación de una determinada asignación, definimos una función que dada la asignación devuelve un valor, que será menor cuanto más se acerque la asignación considerada a la asignación ideal.

Para cada preferencia, debe existir un componente de la función objetivo que la represente. Si el valor de este componente es cero, quiere decir que la asignación satisface esa preferencia, y si es mayor que cero, no lo hace. El valor será cada vez mayor cuanto más nos alejamos de cumplir con la preferencia.

2.1.4 Complejidad del Problema

Ya se ha expresado el problema como un problema de optimización combinatoria. Existe una clase de problemas más simple, los problemas de decisión que tiene la siguiente forma general:

Dado un espacio de soluciones S , una función de costo f , una cota θ , ¿existe alguna solución factible $s \in S$ tal que $f(s) \leq \theta$?

Es evidente que si es posible resolver un problema de optimización combinatoria de búsqueda entonces uno puede resolver la correspondiente versión de decisión, mientras que el recíproco no necesariamente se cumple.

Sea n el largo (el número de dígitos binarios) de la codificación de una instancia del problema de optimización combinatoria. Si la cantidad máxima de tiempo de cálculo necesario para resolver una instancia de largo n está acotada superiormente por un polinomio en n , para cada n , diremos que el problema de optimización combinatoria en estudio es resoluble en tiempo polinómico y que el algoritmo solución correspondiente es polinómico. Si k es el máximo exponente de tal polinomio diremos que el problema de optimización combinatoria es resoluble en un tiempo de orden n^k ($O(n^k)$).

P denota la clase de problemas de decisión para los cuales existe un algoritmo que determine en tiempo polinómico para cada instancia si la respuesta es "sí" o "no".

NP denota la clase de problemas de decisión para los cuales existe un algoritmo que verifique en tiempo polinómico para cada instancia si la respuesta "sí" es correcta.

Obviamente $P \subseteq NP$ y el problema de encontrar si esta inclusión es propia o no, es uno de los problemas abiertos más importantes de la matemática actual.

La mayoría de los problemas combinatorios de decisión están en NP mientras que sólo unos pocos de ellos, aunque en la práctica muy relevantes, pertenecen a P .

Algunos de los más importantes problemas de optimización combinatoria son de hecho NP -completos, es decir tan "difíciles" de resolver como el problema más complejo en NP . Los métodos exactos para resolver problemas conocidos de esta clase requieren de un número exponencial de pasos: entonces cuando la dimensión n de la instancia se hace más y más grande como ocurre en la mayoría de las aplicaciones de la vida real, ningún algoritmo exponencial podría ser de utilidad práctica.

Se puede clasificar el problema a resolver como NP -completo [13], y por lo tanto no es posible aplicar ninguna técnica exacta para resolverlo, por lo que el uso de técnicas heurísticas es indispensable en este caso. También de esta última afirmación, es posible concluir que la solución que se obtendrá no va a ser la ideal, sino una aproximación a la misma.

Taller V - 2002 <i>Asignación de horarios y Salones a Cursos</i>	Tipo de Documento: <i>Informe del Proyecto</i>	Finalizado:
	Versión: 1.0	Aprobado:
Preparado por: IC, JP, YU		Aprobado por:

2.2 Formalización del Problema

La notación utilizada para esta formalización no es la estándar en problemas de optimización, se utilizó una notación que contiene elementos de lenguaje natural para obtener una mayor claridad en los conceptos expresados.

2.2.1. Definición de Conjuntos

Se definen los siguientes conjuntos:

SL: conjunto de salones
 H: conjunto de horarios válidos (intervalos de ½ hora en la semana)
 C: conjunto de clases
 G: conjunto de grupos
 AS: conjunto de asignaturas
 SM: conjunto de semestres
 A: conjunto de años
 CR: conjunto de carreras.
 B: conjunto de bloques
 SMB: { (sm, b) / sm ∈ SM, b ∈ B }
 CRSM: { (cr, sm, conj_clases) / cr ∈ CR, sm ∈ SM, conj_clases ⊂ C }
 “conj_clases” es el conjunto de clases correspondientes a un semestre de una carrera.
 R: conjunto de recursos

Expresaremos los conjuntos con letras mayúsculas y los elementos con minúsculas.

2.2.2. Forma de la Solución

Se define una matriz **S** de tres dimensiones que representará las asignaciones realizadas:

$$S_{m \times n \times p} (SL \times H \times C)$$

Sus elementos son de la forma:

$$S_{sl,h,c} = \begin{cases} 1 & \text{si la clase "c" se asignó al salón "sl" a la hora "h"} \\ 0 & \text{en otro caso.} \end{cases}$$

2.2.3. Función Objetivo

La función objetivo estará compuesta por un término que expresará el grado de insatisfacción de cada una de las preferencias especificadas, ponderadas según su importancia. Podemos entonces expresar la función objetivo como:

$$f = \sum_{i=1}^n peso_i \times costo_i$$

Taller V - 2002 <i>Asignación de horarios y Salones a Cursos</i>	Tipo de Documento: <i>Informe del Proyecto</i>	Finalizado:
	Versión: 1.0	Aprobado:
Preparado por: IC, JP, YU		Aprobado por:

donde n es la cantidad de preferencias, costo_i corresponde al costo según el grado de insatisfacción de la preferencia i , y peso_i es el ponderador asociado a la preferencia i , el cual le asigna una importancia relativa dentro de la función objetivo f .

2.2.4. Restricciones

Para la siguiente formalización se define el operador “[]” como una función que aplicada a una expresión booleana devuelve un entero (0 o 1).

$$[\text{expr}] = \begin{cases} 1 & \text{si expr es verdadero} \\ 0 & \text{si expr es falso.} \end{cases}$$

1. La cantidad de horas asignadas a una clase debe ser igual a la duración de la clase.

$$\sum_{h \in H} \sum_{sl \in SL} s_{sl,h,c} = \text{duración}(c), \forall c \in C$$

2. Una clase debe asignarse a un único salón.

$$\sum_{sl \in SL} \left[\left(\sum_{h \in H} s_{sl,h,c} \right) \geq 1 \right] \leq 1, \forall c \in C$$

El operador [] toma el valor 1 si existe una asignación para la clase c en el salón sl , 0 si no. Por lo tanto la sumatoria que recorre los salones es mayor que 1 si la clase c esta asignada a mas de un salón.

3. Una pareja (horario, salón) debe asignarse a una única clase.

$$\sum_{c \in C} s_{sl,h,c} \leq 1, \forall sl \in SL, \forall h \in H$$

4. Las clases deben asignarse dentro de los horarios válidos.

Se cumple implícitamente por la forma en que se construye la solución.

5. Las clases de un mismo semestre, carrera, y grupo no pueden superponerse. Esta restricción no aplica a las asignaturas electivas, y pueden superponerse entre sí clases que correspondan a asignaturas con más de un grupo.

$$\sum_{\substack{x \in CS \\ \text{grupos}(\text{asignatura}(x))=1}} \sum_{\substack{cs \in CS \\ cs \neq x}} \left(\sum_{sl \in SL} s_{sl,h,x} \times \sum_{sl \in SL} s_{sl,h,cs} \right) \forall h \in H, \forall CS \in CRSM$$

Taller V - 2002 <i>Asignación de horarios y Salones a Cursos</i>	Tipo de Documento: <i>Informe del Proyecto</i>	Finalizado:
	Versión: 1.0	Aprobado:
Preparado por: IC, JP, YU		Aprobado por:

6. Las horas asignadas a una clase deben ser contiguas.

Representando la unidad horaria siguiente a una unidad horaria h como $h+1$, h_{primero} como la primer unidad horaria y h_{ultimo} como la última, se expresa como:

$$s_{sl, h_{\text{primero}}, c} \times s_{sl, h_{\text{ultimo}}, c} + \sum_{\substack{h \in H \\ h \neq h_{\text{ultimo}}}} [s_{sl, h, c} \neq s_{sl, h+1, c}] \leq 2, \forall sl \in SL, \forall c \in C$$

Se controlará que no haya más de dos cambios de 0 a 1 y de 1 a 0 en el conjunto ordenado de horarios correspondientes a cada salón y clase, contemplando también que no estén asignadas a la misma clase y salón la primer y última unidad horaria al mismo tiempo.

El primer termino de la suma vale 1 si la clase c esta asignada tanto al primer como al ultimo horario disponible en el salón sl .

El operador $[\]$ toma el valor 0 si el salón sl esta ocupado por una clase c en horarios contiguos (h y $h+1$) o si están libres ambos horarios, 1 si esta ocupado para uno de los dos horarios y libre para el otro.

7. Una clase debe comenzar y finalizar dentro del mismo día.

$$\sum_{\substack{h \in H \\ h \neq h_{\text{ultimo}}}} s_{sl, h, c} \times s_{sl, h+1, c} \times [dia(h) \neq dia(h+1)] = 0, \forall sl \in SL, \forall c \in C$$

El operador $[\]$ toma el valor 0 si los horarios contiguos h y $h+1$ pertenecen al mismo día, 1 si no.

8. Los grupos que tienen un bloque especificado solo pueden asignarse dentro de la franja horaria correspondiente al bloque.

$$\sum_{sl \in SL} \sum_{\substack{h \in H \\ h \notin \text{bloque}(c)}} s_{sl, h, c} = 0, \forall c \in C$$

9. No pueden dictarse dos clases de un mismo grupo y asignatura un mismo día.

$$dia(c_1) \neq dia(c_2)$$

$$\forall as \in AS, \forall g \in G, \forall c_1, c_2 \in C / \text{asignatura}(g) = as \text{ y } grupo(c_1) = grupo(c_2) = g$$

2.2.5. Preferencias

1. Es deseable que no queden clases sin asignar.

$$costo_1 = \sum_{c \in C} \left[\sum_{h \in H} \sum_{sl \in SL} s_{sl, h, c} = 0 \right]$$

El operador $[\]$ toma el valor 1 si la clase c no esta asignada, 0 si lo esta.

Taller V - 2002 <i>Asignación de horarios y Salones a Cursos</i>	Tipo de Documento: <i>Informe del Proyecto</i>	Finalizado:
	Versión: 1.0	Aprobado:
Preparado por: IC, JP, YU		Aprobado por:

2. Las clases deberían asignarse dentro del rango horario de preferencia especificado.

$$costo_2 = \sum_{c \in C} \left[\sum_{\substack{h \in H \\ h \notin rango(c)}} \sum_{sl \in SL} S_{sl,h,c} \neq 0 \right]$$

El operador [] toma el valor 1 si la clase c esta asignada en un horario h , que no pertenece a su rango de preferencia, 0 en caso contrario.

3. Una clase no debería asignarse a un salón con menor capacidad de la requerida.

$$costo_3 = \sum_{c \in C} \sum_{\substack{sl \in SL \\ capacidad(sl) < cant_alumnos(c)}} \left[\sum_{h \in H} S_{sl,h,c} \neq 0 \right] \times (cant_alumnos(c) - capacidad(sl))$$

El operador [] toma el valor 1 si la clase c esta asignada al salón sl en cualquier horario, 0 si no lo esta.

4. Es deseable que el salón asignado a una clase cuente con los recursos necesarios para el dictado de la misma. Por ej., si la clase necesita enchufe para un retroproyector, el salón debe tener enchufes.

$$costo_4 = \sum_{c \in C} \left[\sum_{h \in H} \sum_{\substack{sl \in SL \\ recursos(c) \not\subseteq recursos(sl)}} S_{sl,h,c} \neq 0 \right]$$

El operador [] toma el valor 1 si la clase c esta asignada a un salón sl , el cual no dispone de algún recurso necesario para el dictado de dicha clase, 0 en caso contrario.

5. Es deseable que los teóricos y los prácticos correspondientes se dicten en horarios contiguos.

Si no se dan las condiciones de preferencia de horarios y/o turnos adecuadas para poder exigir la adyacencia, se considera que la misma se cumple, es decir, no se penalizará la asignación dada. Por ejemplo, si en un grupo existe una única clase de teórico y una única clase de práctico, y tienen distintos días como preferencia horaria, no es correcto pedir la adyacencia de estas dos clases. Al aplicar esta preferencia a clases teóricas y prácticas, si la cantidad de clases teóricas es mayor que la de clases prácticas, se controlará que todo práctico sea adyacente a algún teórico, mientras que si existen más clases prácticas que teóricas, se controlará que todo teórico sea adyacente a algún práctico. Se definen los siguientes conjuntos:

AS-TEO: conjunto de asignaturas con mayor o igual cantidad de clases teóricas que prácticas, pero con teóricos y prácticos.

AS-PRA: conjunto de asignaturas con más clases prácticas que teóricas, pero con teóricos y prácticos.

MAS-C: conjunto de clases prácticas correspondientes a asignaturas de AS-PRA y clases teóricas correspondientes a AS-TEO

MENOS-C: conjunto de clases que no pertenecen a MAS-C y corresponden a asignaturas de AS-PRA o AS-TEO

Taller V - 2002 <i>Asignación de horarios y Salones a Cursos</i>	Tipo de Documento: <i>Informe del Proyecto</i>	Finalizado:
	Versión: 1.0	Aprobado:
Preparado por: IC, JP, YU		Aprobado por:

Para expresar esta preferencia utilizamos una matriz P que indica entre qué clases sería deseable la adyacencia horaria, $P_{c1 \times c2}$ (MAS-C \times MENOS-C)

$$costo_5 = \sum_{c_1 \in MAS-C} \sum_{c_2 \in MENOS-C} \left[p_{c_1 c_2} \times \sum_{\substack{h \in H \\ h \in rango(c_1) \\ h+1 \in rango(c_2) \\ h \neq h_{ultimo}}} \left(\sum_{sl \in SL} S_{sl,h,c_1} \right) \times \left(\sum_{sl \in SL} S_{sl,h+1,c_2} \right) + \sum_{\substack{h \in H \\ h+1 \in rango(c_1) \\ h \in rango(c_2) \\ h \neq h_{ultimo}}} \left(\sum_{sl \in SL} S_{sl,h,c_2} \right) \times \left(\sum_{sl \in SL} S_{sl,h+1,c_1} \right) = 0 \right]$$

6. No sería bueno que se utilizaran salones de gran capacidad para clases de pocos alumnos.

$$costo_6 = \sum_{c \in C} \sum_{\substack{sl \in SL \\ capacidad(sl) > cant_alumnos(c)}} \left[\sum_{h \in H} S_{sl,h,c} \neq 0 \right] \times (capacidad(sl) - cant_alumnos(c))$$

El operador [] toma el valor 1 si la clase c esta asignada al salón sl en cualquier horario, 0 si no lo esta.

7. Es conveniente que las clases teóricas de un grupo se repartan en forma uniforme en la semana.

Esta preferencia permite expresar que es conveniente que las clases de un mismo grupo se asignen a días distantes de la semana. Por ejemplo, si un curso tiene 3 clases por semana, no es conveniente que éstas se asignen a 3 días consecutivos. Para controlar esta preferencia deben tenerse en cuenta las preferencias de horarios ingresadas.

Una forma de evaluar esta preferencia es medir la distancia en días entre la primera y la última clase de un grupo. Dependiendo de la cantidad de clases por semana, existirá un mínimo exigible para esta distancia:

- ◆ 1 clase por semana: no aplicable
- ◆ 2 clases por semana: distancia ≥ 2 (o sea al menos un día libre entre ambas clases)
- ◆ 3 clases por semana: distancia ≥ 3
- ◆ 4 clases por semana: distancia ≥ 4
- ◆ 5 clases por semana: distancia ≥ 5
- ◆ 6 clases por semana: no aplicable.

$$costo_7 = \sum_{\substack{g \in G \\ tipo_clase(g)=teórico}} \left[dia(ultima_clase(g)) - dia(primer_clase(g)) < distancia_minima(g) \right] \times \left[dia(rango(ultima_clase(g))) - dia(rango(primer_clase(g))) < distancia_minima(g) \right]$$

El primer operador [] toma el valor 1 si la distancia entre los días asignados a la primer y ultima clase del grupo g es menor a la distancia mínima definida para ese grupo, 0 en caso contrario.

El segundo operador [] toma el valor 1 si la distancia entre los días del rango de preferencia de la primer y ultima clase del grupo g es menor a la distancia mínima definida para ese grupo, 0 en caso contrario.

Taller V - 2002 <i>Asignación de horarios y Salones a Cursos</i>	Tipo de Documento: <i>Informe del Proyecto</i>	Finalizado:
	Versión: 1.0	Aprobado:
Preparado por: IC, JP, YU		Aprobado por:

8. Similitud de horarios en las distintas clases teóricas de un grupo.

La idea de esta preferencia es que los horarios asignados a los teóricos se asignen en horas cercanas, por lo que se controlará que, los horarios asignados a dos clases teóricas de un mismo grupo coincidan en por lo menos media hora. No se aplica si los rangos de preferencia de ambas clases son disjuntos.

$$\sum_{\substack{g \in G \\ \text{tipo_clase}(g)=\text{teórico}}} \sum_{\substack{c_1 \in C \\ \text{grupo}(c_1)=g}} \sum_{\substack{c_2 \in C \\ c_2 \neq c_1 \\ \text{grupo}(c_2)=g}} \left[\sum_{h_1 \in H} \sum_{\substack{h_2 \in H \\ \text{hora}(h_1)=\text{hora}(h_2)}} \left(\sum_{sl \in SL} S_{sl,h_1,c_1} \right) \times \left(\sum_{sl \in SL} S_{sl,h_2,c_2} \right) = 0 \right] \times \left[\sum_{h_1 \in \text{rango}(c_1)} \sum_{h_2 \in \text{rango}(c_2)} [\text{hora}(h_1) = \text{hora}(h_2)] \neq 0 \right]$$

El primer operador [] toma el valor 1 si las clases c_1 y c_2 no están asignadas a horarios que difieren en más de media hora, 0 en caso contrario.

El segundo operador [] toma el valor 0 si los rangos de preferencia de las clases c_1 y c_2 son disjuntos 1 en caso contrario.

2.2.6. Aclaraciones

- Los pesos de cada preferencia que intervienen en la función objetivo se determinan en la etapa de calibración
- Debido a la complejidad del problema, en la etapa de implementación se optó por considerar las restricciones como preferencias con costo sensiblemente superior, ver sección 4.2.2 del informe.
- Debido a las estructuras elegidas para implementar la solución, algunas restricciones y preferencias se cumplen de forma implícita, ver sección 4.2.2 del informe.

Taller V - 2002 <i>Asignación de horarios y Salones a Cursos</i>	Tipo de Documento: <i>Informe del Proyecto</i>	Finalizado:
	Versión: 1.0	Aprobado:
Preparado por: IC, JP, YU		Aprobado por:

Anexo 3. Estudio de Técnicas Aplicables a Problemas de Asignación

En la evaluación de antecedentes realizada, se encontró que las heurísticas más utilizadas para resolver este tipo de problema son: Algoritmos Genéticos[8,10,11] (GA), Simulated Annealing [8,9] (SA) y Tabú search [2,3,8,14,15] (TS).

También se ha utilizado Ant Systems (AS) , en un proyecto realizado para la bedelía de la Facultad de Ingeniería [12].

La información aquí expresada surge en su mayoría de papers, los cuales podemos encontrar como referencias de este documento. Para obtenerlos, se recurrió a tres fuentes principales: Internet, la biblioteca del Instituto de Computación de la Facultad de Ingeniería y nuestra tutora María Urquhart, quien nos facilitó algunos de ellos.

A continuación se describen cada una de las heurísticas mencionadas.

3.1 Ant Systems

Esta heurística se basa en imitar el comportamiento de las colonias de hormigas. La idea básica es simular el comportamiento de un conjunto de agentes simples que cooperan para llegar a un objetivo.

Las hormigas, a pesar de ser insectos casi ciegos, son capaces de encontrar el camino más corto desde el hormiguero a una fuente de comida y regresar. Estudios etológicos han descubierto que esta capacidad es el resultado de una interacción vía comunicación química entre las hormigas y un fenómeno emergente causado por la presencia simultánea de muchas hormigas.

La comunicación entre ellas se realiza a través de una sustancia llamada feromona, la cual es depositada por estas (en variadas cantidades) sobre la tierra, el camino es marcado entonces por los rastros de esta sustancia. Los movimientos de las hormigas son esencialmente aleatorios, pero si detectan un camino marcado previamente por dicha sustancia, es altamente probable que decidan seguirlo, y así lo refuerzan con su propia feromona. El comportamiento colectivo resultante es "autocatalítico", cuanto más hormigas siguen una huella, más atractiva se torna la misma de ser seguida. El proceso entonces se caracteriza por un loop de retroalimentación positiva, donde la probabilidad con que una hormiga elige un camino aumenta con el número de hormigas que previamente eligieron el mismo camino.

Taller V - 2002 <i>Asignación de horarios y Salones a Cursos</i>	Tipo de Documento: <i>Informe del Proyecto</i>	Finalizado:
	Versión: 1.0	Aprobado:
Preparado por: IC, JP, YU		Aprobado por:

Dado el siguiente ejemplo:

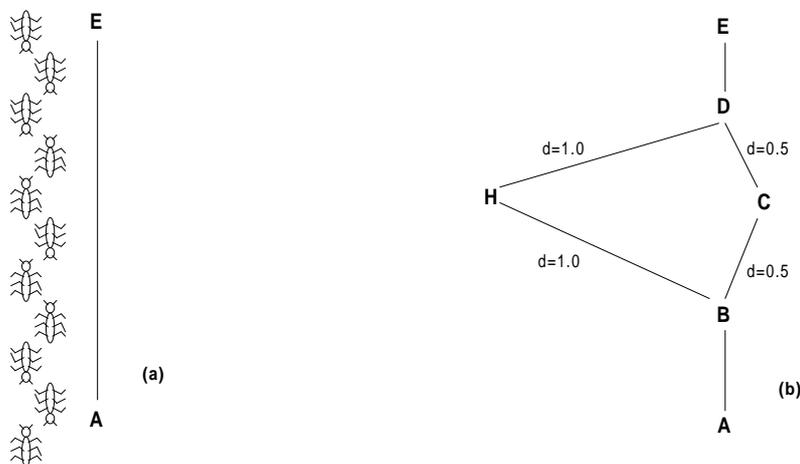


Figura A3 - 1: Comportamiento de las hormigas

En la figura (a) se muestra un camino a través del cual las hormigas están caminando (por ejemplo de la comida A hasta el hormiguero E y viceversa). Luego aparece un obstáculo y el camino es cortado. Entonces en el punto B las hormigas que están caminando desde A hacia E (o en D las que están caminando en la dirección opuesta) deben decidir si girar hacia la izquierda o la derecha. Esta elección es realizada teniendo en cuenta la cantidad de feromona dejada por las hormigas que pasaron previamente. La primera hormiga que llega al punto B (o D) tiene la misma probabilidad de ir hacia la izquierda o derecha (ya que no hay feromona en ninguna de las dos alternativas). Debido a que el camino BCD es más corto que el BHD, la primera hormiga que siga este, llegará primero a D, que las que sigan el camino BHD. El resultado es que una hormiga retornando desde E hacia D encontrará un trazo más fuerte en el camino DCBA causada por la mitad de todas las hormigas que decidieron recorrer el obstáculo vía DCBA y por las que ya llegaron vía BCD. Por lo tanto ahora la probabilidad de elegir el camino DCB será mayor que la del camino DHB. Como consecuencia, el número de hormigas siguiendo el camino BCD por unidad de tiempo será mayor que el número de hormigas siguiendo BHD. Esto provoca que la cantidad de feromona crezca más rápido en el camino más corto, que en el largo, y por lo tanto la probabilidad de seguir este camino aumenta rápidamente. Finalmente el camino más largo será descartado.

Basándose en esta idea, se desarrolló una técnica útil para resolver problemas de optimización combinatoria estocástica. Sus principales características son:

- ◆ retroalimentación positiva: permite el rápido descubrimiento de buenas soluciones.
- ◆ cálculo distribuido: evita la convergencia prematura.
- ◆ uso de una heurística 'greedy' constructiva: ayuda a encontrar soluciones aceptables en las primeras etapas del proceso de búsqueda.

Los 'agentes' que se utilizan difieren de las hormigas reales en que:

- ◆ tienen memoria
- ◆ no son completamente ciegos (tienen cierta 'inteligencia')
- ◆ viven en un ambiente donde el tiempo es discreto

Taller V - 2002 <i>Asignación de horarios y Salones a Cursos</i>	Tipo de Documento: <i>Informe del Proyecto</i>	Finalizado:
	Versión: 1.0	Aprobado:
Preparado por: IC, JP, YU		Aprobado por:

3.2 Algoritmos Genéticos

Un problema de optimización se traduce en el problema de encontrar el individuo más apto (a veces llamado cromosoma) dentro de una población. La aptitud se mide por medio de una función de aptitud (fitness) la cual se relaciona con la función objetivo del problema a resolver. Los individuos (cromosomas) son equivalentes a soluciones y una población es un conjunto de N individuos.

Cada individuo consiste en una secuencia (usualmente un string) de elementos atómicos llamados genes. Cada gen puede tomar valores (allele) en un conjunto predefinido. Es usual que cada gen tome valores en el conjunto {0,1} o en el conjunto de los dígitos.

El algoritmo genético actúa sobre la población de cromosomas modificando sus componentes. Las modificaciones ocurren de acuerdo a las reglas de la genética a través de la aplicación de operadores genéticos.

El operador Reproducción obtiene un nuevo individuo tomando en cuenta el fitness de cada individuo en la población actual, es decir de la población actual elige de acuerdo al fitness de los individuos cuáles se van a reproducir en la población siguiente (Newpop).

El operador Crossover toma como entrada la nueva población NewPop y devuelve otra población CrossPop. Dicho operador extrae aleatoriamente dos individuos (padres), elige con probabilidad de distribución uniforme un punto de cruzamiento en los cromosomas que representan a los dos padres y entonces intercambia los valores a la derecha de este punto recombinándolos y generando dos hijos. El operador CrossOver es aplicado con una probabilidad p_c independiente de los individuos específicos sobre los cuales es aplicado. Sirve para crear nuevos individuos que preserven las mejores características de sus padres.

El operador Mutación toma como entrada la población CrossPop y devuelve una nueva población MutPop en la cual algunos individuos han sufrido mutaciones. Cada individuo puede ser seleccionado con probabilidad p_m y aquellos que lo son sufren alguna mutación. Por ejemplo, en el caso que los individuos estén codificados como strings de ceros y unos la mutación cambia un uno por cero o viceversa. El operador mutación introduce variaciones de base en la población garantizando la posibilidad de explorar todo el espacio de búsqueda independientemente de la población inicial específica.

Para resolver un problema de optimización usando algoritmos genéticos debemos elegir la estructura de los cromosomas para codificar en ellos las soluciones. Se debe determinar la función de fitness como función de los valores de los genes, la cual estará directamente relacionada con la función objetivo del problema a resolver. Para hallar la solución se aplican sucesivamente los tres operadores estudiados a cada población resultante hasta que finalmente la población final estará solo compuesta por un tipo de individuos, el más apto, que constituirá la solución al problema.

Taller V - 2002 <i>Asignación de horarios y Salones a Cursos</i>	Tipo de Documento: <i>Informe del Proyecto</i>	Finalizado:
	Versión: 1.0	Aprobado:
Preparado por: IC, JP, YU		Aprobado por:

3.3 Tabú Search

Tabú Search es una estrategia para resolver problemas de optimización combinatoria cuyo rango de aplicación va desde teoría de grafos hasta problemas de programación entera.

Es un procedimiento adaptativo con la habilidad de usar otros métodos (como algoritmos de programación lineal o heurísticas especializadas) lo cual hace que pueda obviar las limitaciones de la optimalidad local. Se lo puede clasificar como una meta-heurística diseñada para obtener un óptimo global de un problema de optimización combinatoria.

Para mejorar la eficiencia del proceso de búsqueda, se necesita no solamente llevar registro de la información local (como el valor actual de la función objetivo) sino también la información del proceso de búsqueda. El uso sistemático de la *memoria* es una característica esencial del Tabú Search, como veremos más adelante.

Algunos de los principios básicos del Tabú Search provienen de las técnicas de búsqueda local, por lo que veremos un breve repaso de las mismas:

4.3.1. Técnicas de Búsqueda Local

Las técnicas de búsqueda local son una familia de técnicas de propósito general para la solución de problemas de optimización. Están basadas en la noción de *vecindario* (*neighborhood*).

Considérese un problema de optimización con un espacio de soluciones S y una función objetivo f a minimizar. Existe una función N , que depende de la estructura de cada problema en particular, que asigna a cada solución factible $s \in S$ su vecindario $N(s) \in S$. Cada solución $s' \in N(s)$ se llama vecina de s .

Una técnica de búsqueda local, partiendo de una solución inicial s_0 (que puede ser obtenida por cualquier otro método o de forma aleatoria), entra en un loop que navega por el espacio de soluciones, pasando en cada iteración de una solución a otra solución vecina. Llamaremos un *movimiento* a la modificación que transforma una solución en una de sus vecinas.

Dentro de las técnicas de búsqueda local tenemos la siguiente clasificación:

- ◆ Steepest Descendent Method
- ◆ Randomized Descendent Meted
- ◆ Randomized Non-Ascendent Method
- ◆ Steepest Non-Ascendent Method

El *steepest descendent method* analiza todos los posibles movimientos y elige aquel que produce la mejora más significativa. Acepta el movimiento candidato solamente si este mejora el valor de la función objetivo y se detiene apenas encuentra un mínimo local. Este método requiere de la exploración de todo el vecindario.

El *randomized descendent method*, analiza un vecindario aleatorio y lo acepta si es mejor que el actual, de lo contrario la solución actual se mantiene y se genera otro vecindario. Se detiene luego de un número fijo de iteraciones sin haber mejorado la función objetivo. Este método también queda atrapado en el primer mínimo local al que llega.

Taller V - 2002 <i>Asignación de horarios y Salones a Cursos</i>	Tipo de Documento: <i>Informe del Proyecto</i>	Finalizado:
	Versión: 1.0	Aprobado:
Preparado por: IC, JP, YU		Aprobado por:

El *randomized non-ascendent method* analiza un vecindario aleatorio y lo acepta si es mejor o igual al actual. Al igual que el método anterior, también se detiene luego de un número fijo de iteraciones sin haber mejorado el valor de la función objetivo. Este método permite *movimientos oblicuos*, por lo cual tiene la particularidad de seguir caminos descendentes que pasan por *llanuras (plateaus)* en las cuales puede caer la búsqueda, y le da la habilidad de salir de estas llanuras y continuar la búsqueda.

El *steepest non-ascendent method* combina la búsqueda por el movimiento más profundo con movimientos oblicuos, y estas son sus principales características.

Las técnicas de búsqueda local tienen la ventaja de permitir la interactividad, es decir que se puede partir de una solución inicial preelaborada o bien modificarse la misma a medida que se va construyendo.

4.3.2. Descripción del Método

Partiendo de una solución inicial s_0 , el algoritmo de Tabú Search explora un subconjunto V del vecindario $N(s)$ de la solución actual s . El elemento de V que proporciona el mínimo valor para la función objetivo se convierte en la nueva solución independientemente del hecho de que su valor sea mejor o peor que el valor de s .

Para prevenir los ciclos, existe una lista que llamaremos *tabu-list*, que es la lista de los movimientos prohibidos de realizar. En esta lista se van guardando los últimos k movimientos aceptados (donde k es un parámetro del método) y se maneja como una cola de largo fijo. Cuando un nuevo movimiento es agregado, el más viejo es sacado de la lista. Aquí es donde podemos observar el uso de la memoria por este método, evitando de esta manera volver a soluciones anteriores.

Puede pensarse que la decisión de que algunos movimientos son *tabú* es demasiado absoluta, por lo que se introduce un mecanismo para cancelar el status de *tabú* de los mismos. Si un movimiento proporciona una mejora significativa de la función objetivo, entonces su status de *tabú* es dejado de lado y la solución resultante de aplicar este movimiento es aceptada como la nueva solución actual. Más precisamente, se define una *función de aspiración* A que, para cada valor v de la función objetivo, devuelve otro valor v' que representa el valor que el algoritmo *aspira* a llegar desde v . Dada la solución actual s , la función objetivo f y una solución vecina s' obtenida a partir de un movimiento m , si $f(s') \leq A(f(s))$, entonces s' puede ser aceptada, incluso si m está en la *tabu-list*.

El procedimiento se detiene cuando se cumplen algunas de estas condiciones:

- ◆ Se cumplió con un máximo dado T_{smax} de iteraciones sin mejoras en la función objetivo.
- ◆ El valor de la función objetivo para la solución actual es mejor que un límite fijado previamente para el mismo f^* .

Los principales parámetros de este método son:

- ◆ El largo de la *tabu-list* k .
- ◆ La cardinalidad del conjunto V de soluciones vecinas testeadas en cada iteración.
- ◆ El máximo de iteraciones T_{smax} sin conseguir mejorar el valor de la función objetivo.

Taller V - 2002 <i>Asignación de horarios y Salones a Cursos</i>	Tipo de Documento: <i>Informe del Proyecto</i>	Finalizado:
	Versión: 1.0	Aprobado:
Preparado por: IC, JP, YU		Aprobado por:

Seudo código

```

f* := limite fijado para la función objetivo;
s0 := solución inicial en S ;
nlter := 0 ; // iteración actual
blter := 0 ; // iteración en donde se encontró la mejor solución
bestSol := s0 ;
TL := ∅ // Inicializar la tabú list ;
Inicializar función de aspiración A ;

while ( f(bestSol) > f* ) and ( nlter – blter > TSmax ) do

    nlter := nlter + 1;
    Generar un conjunto V de soluciones si en N(s), las cuales no son tabú o
        se cumple que A( f(si ) ) ≥ f(si ) ;
    s* := solución que minimiza f sobre V ;
    Actualizar la tabú list TL y la función de aspiración A ;
    if f(s*) < f(bestSol) then
        bestSol := s* ;

```

3.4 Simulated Annealing

Es básicamente un método de búsqueda local. Partiendo de una solución factible inicial 'i' se genera mediante algún procedimiento una solución vecina $j \in V(i)$ donde $V(i)$ es el conjunto de soluciones vecinas a 'i'. En esta heurística, la solución 'j' se aceptará como la nueva solución factible con cierta probabilidad aún cuando la misma sea peor que la solución de partida 'i'. Si consideramos una función objetivo a minimizar 'f', sea $\Delta = f(j) - f(i)$. La probabilidad de aceptar a 'j' como la nueva solución está dada por:

$$\text{Prob (aceptar } j) = \begin{cases} 1 & \text{si } \Delta < 0 \\ e^{-\Delta/t} & \text{si } \Delta \geq 0 \end{cases}$$

Donde 't' es un parámetro de control del algoritmo, a menudo llamado 'temperatura'.

Este proceso se repite para un valor dado de 't' hasta que se llegue a una condición de *equilibrio* que dependerá del problema. En su forma más simple esta condición se reduce a un número fijo de iteraciones. Luego de llegar al equilibrio la temperatura 't' se decrementa de acuerdo a alguna regla y el proceso vuelve a comenzar.

El algoritmo termina al llegar a un valor mínimo de 't' establecido, en el que prácticamente 'j' se acepta sólo si $\Delta < 0$; en este punto el algoritmo actúa como un algoritmo simple de búsqueda local.

El conjunto de parámetros que incluye temperatura inicial, final, regla para decrementar la temperatura y condición de equilibrio se conoce como 'estrategia de enfriamiento', y ésta es fundamental para que el algoritmo sea exitoso.

Taller V - 2002 <i>Asignación de horarios y Salones a Cursos</i>	Tipo de Documento: <i>Informe del Proyecto</i>	Finalizado:
	Versión: 1.0	Aprobado:
Preparado por: IC, JP, YU		Aprobado por:

3.5 Elección de la Heurística a Utilizar

Ant Systems es una de las técnicas que se podría utilizar en este problema, pero debido a que en su planteo original construye una solución totalmente nueva, sin tomar en cuenta soluciones anteriores no será la que se utilizará en este caso. Por otra parte, se tuvo en cuenta que este método fue utilizado en un proyecto anterior [12] y, según las conclusiones de dicho proyecto, no se obtuvieron resultados muy alentadores, incluso trabajando con un conjunto reducido de datos de prueba.

GA se basa en la manipulación de un conjunto de soluciones al problema. Dada la complejidad de la estructura de éstas, parece complejo poder generar varias soluciones de partida para el algoritmo. Este aspecto negativo, el de generar varias soluciones de partida, es una de las principales razones por la cual no se eligió esta heurística, aunque es una de las posibles candidatas para una extensión futura del proyecto.

Tanto TS como SA utilizan las técnicas de búsqueda local, las cuales, como se dijo anteriormente, dan la posibilidad de partir de cualquier solución inicial, como así también la posibilidad de interactuar modificando la solución en cualquier punto de la búsqueda.

De hecho, una vez que una asignación ha sido generada puede servir como punto de partida para una nueva búsqueda luego de haber modificado algunas de las restricciones del problema, como por ejemplo, cambiar los rangos horarios de dictado de clases o las preferencias horarias de alguna de ellas. Dicho proceso generalmente lleva a una buena asignación para el nuevo problema en un tiempo razonable.

Estas técnicas son muy adecuadas para el problema de asignación de horarios y salones, ya que para la asignación de un semestre en particular, podemos partir de la solución ya obtenida en el mismo semestre para el año anterior, cambiando algunas de las restricciones o preferencias correspondientes al semestre a asignar. Otra ventaja de esto es que la solución encontrada va a ser parecida a la anterior, por lo que se mantendría una coherencia semestre a semestre entre las asignaciones.

La siguiente tabla ilustra algunas de las características que fueron consideradas para la elección del algoritmo a utilizar:

	Tabú Search	Sim. Annealing	Genetic Algor.	Ant Systems
Solución Inicial	Puede usar cualquier solución inicial	Puede usar cualquier solución inicial	Parte de un conjunto de soluciones	La solución inicial es totalmente nueva
Adaptabilidad al problema	Muy buena	Buena	Muy buena	Buena
Usado anteriormente para problemas similares	Si, con buenos resultados	Si, con resultados relativamente buenos	Si, con buenos resultados	Si, con resultados no muy alentadores.
Complejidad de desarrollo	Media	Simple	Compleja	Media
Factibilidad de las Soluciones	Es posible encontrar buenas soluciones.	Es posible encontrar buenas soluciones, lleva más tiempo que otros.	Es posible encontrar buenas soluciones.	Es posible encontrar buenas soluciones, en teoría.

Tabla A3 - 1: Comparación de algoritmos.

Se eligió utilizar Tabú Search teniendo en cuenta lo anterior y estudios ya realizados [8], donde se aplicaron TS, SA y GA al problema de 'Timetable' obteniéndose los mejores resultados

Taller V - 2002 <i>Asignación de horarios y Salones a Cursos</i>	Tipo de Documento: <i>Informe del Proyecto</i>	Finalizado:
	Versión: 1.0	Aprobado:
Preparado por: IC, JP, YU		Aprobado por:

para TS. También Tabú Search ha sido reportado por ser muy efectivo en problemas de 'Timetable' en [2,3,8,14,15].

A continuación, se muestran algunos resultados obtenidos en estudios anteriores:

En la referencia [8], Metaheuristics for High-School Timetabling
A. Colomi, M. Dorigo, V. Maniezzo, Computational Optimization and Applications Journal

En este paper, se aplican tres de los algoritmos que hemos estado viendo al mismo problema de Asignación, en un liceo de Milan, Italia.

No se detallan la cantidad de clases y salones a asignar. El método para comparar los resultados obtenidos con los diferentes algoritmos fue ejecutar 10 veces cada uno sobre el problema planteado.

Los resultados obtenidos son los siguientes:

Algoritmo	Mejor Costo	Costo Promedio
Solución Manual	234	234
Genetic Algorithms	91	111
Tabú Search	85	97
Simulated Annealing	164	174

Tabla A3 - 2: Resultados obtenidos en el estudio de la referencia [8].

En la referencia [15], A Tabú search Algorithm for computing an operational Timetable, D. Costa, European Journal of Operational Research 76 (1994), 98-110

En este paper, se aplica el Tabú Search a dos casos distintos de asignación de salones y horarios:

El primer caso se trata de un liceo un Francia. Se tienen para asignar 375 asignaturas las cuales se traducen en 780 clases y se cuenta con 12 salones y 50 períodos de horario.

Los resultados obtenidos son los siguientes:

Método de Asignación	Costo	Tiempo
Solución Manual	735	Tres semanas
Solución con Tabú Search	722	41 minutos

Tabla A3 - 3: Resultados obtenidos en el estudio de la referencia [15], liceo de Francia.

El segundo caso se trata de un liceo en Suiza. Se tienen para asignar 162 asignaturas, las cuales representan 355 clases y se cuenta con 17 salones y 50 períodos de horario.

Los resultados obtenidos son los siguientes, la solución manual no estaba disponible para comparar costos.

Método de Asignación	Costo	Tiempo
Solución Manual	n/d	Dos semanas
Solución con Tabú Search	380	79 minutos

Tabla A3 - 4: Resultados obtenidos en el estudio de la referencia [15], liceo de Suiza.

Taller V - 2002 <i>Asignación de horarios y Salones a Cursos</i>	Tipo de Documento: <i>Informe del Proyecto</i>	Finalizado:
	Versión: 1.0	Aprobado:
Preparado por: IC, JP, YU		Aprobado por:

En resumen, las principales razones para utilizar el Tabú Search son:

La posibilidad de partir de cualquier solución inicial

Este punto es muy significativo, ya que nos permite utilizar como solución inicial cualquier asignación de años anteriores, ya que es muy probable que esta no cambie mucho de un año a otro. El partir de una solución anterior reduce significativamente el tiempo de proceso para encontrar una nueva solución. Permite además mantener cierta coherencia en los horarios de un año a otro.

También es posible generar una solución a partir de cualquier solución inicial, lo cual le permitiría al usuario de la herramienta elegir como punto de partida una solución anterior o no. En el caso de que no elija partir de una anterior, se generará una solución de forma aleatoria como punto de partida.

La posibilidad de interactuar modificando la solución actual en cualquier punto de la búsqueda.

Este es otro punto decisivo para la elección. El usuario podría en cualquier momento del proceso de generación de una asignación, frenar el mismo, ver como ha evolucionado la solución propuesta, hacer sus modificaciones y comenzar nuevamente el proceso a partir de esta asignación modificada manualmente.

Estudios anteriores

De la información obtenida en los trabajos tomados como referencia, deducimos que el Tabú Search es la técnica con la cual se obtuvieron los mejores resultados, comparados con las demás técnicas aquí evaluadas.

3.6 Adaptación del Tabú Search a la Asignación de Salones y Horarios

El Tabú Search es una estrategia para resolver problemas de optimización combinatoria con un rango de aplicación muy amplio, en este punto se describe de que forma se adaptará el algoritmo para resolver el problema de Asignación de Clases a Salones y Horarios.

Podemos plantear una aplicación general del Tabú Search de la siguiente forma:

```

f*:= limite fijado para la función objetivo;
s0 := solución inicial en S ;
nIter := 0 ; // iteración actual
bIter := 0 ; // iteración en donde se encontró la mejor solución
bestSol := s0 ;
TL := φ // Inicializar la tabú list ;
Inicializar función de aspiración A ;

while ( f(bestSol) > f* ) and ( nIter - bIter > TSmax ) do

  nIter := nIter + 1 ;
  Generar un conjunto V de soluciones si en N(s), las cuales no son tabú o
  se cumple que A( f(s) ) ≥ f(si) ;
  s* := solución que minimiza f sobre V ;
  Actualizar la tabú list TL y la función de aspiración A ;
  if f(s*) < f(bestSol) then
    bestSol := s* ;

```

Taller V - 2002 <i>Asignación de horarios y Salones a Cursos</i>	Tipo de Documento: <i>Informe del Proyecto</i>	Finalizado:
	Versión: 1.0	Aprobado:
Preparado por: IC, JP, YU		Aprobado por:

Se deberá adaptar este planteo general a las condiciones del problema de Asignación, por lo que se definirán los siguientes puntos a tener en cuenta del algoritmo.

- ◆ Restricciones y Preferencias
- ◆ La solución inicial s_0
- ◆ Definición de un movimiento y generación de soluciones vecinas
- ◆ La Tabú List
- ◆ La función de Aspiración
- ◆ Calibración del Método
- ◆ Criterio de parada

3.6.1. Restricciones y Preferencias

A cada restricción y preferencia se le asignará un peso, que será el que tenga en cuenta la función objetivo cuando realiza el cálculo del costo de una asignación. El peso de las restricciones deberá ser considerablemente más alto que el de las preferencias, ya que es más importante para una asignación satisfacer las restricciones sobre las preferencias, estas simplemente sirven para comparar soluciones que ya cumplen con las anteriores restricciones.

El peso de cada restricción y preferencia se definirá en la etapa de calibración del método.

3.6.2. Solución Inicial

Se tomará como solución inicial s_0 una asignación anterior o una solución generado en forma aleatoria, a la cual se le ira aplicando las sucesivas iteraciones de las que se compone el Tabú Search buscando una mejor asignación de la que se partió.

Para el caso de la primera vez que se ejecute el proceso de optimización de la solución, se hará un volcado de los datos de las asignaciones del año 2002 a partir del programa actualmente utilizado en Bedelía, para usar estas como solución inicial.

3.6.3. Movimientos y Soluciones Vecinas

Se define un movimiento como la asignación de una clase a la pareja salón/horario distinta a la actual (Con horario se quiere decir hora de inicio de una clase). Como se especificó anteriormente, la forma de la solución es una matriz de tres dimensiones, que se pueden reducir a dos, por lo tanto el movimiento a la siguiente pareja puede provocar tres situaciones:

1. que todo el horario de la clase quede dentro del mismo día y salón; en ese caso, queda cumplido el movimiento
2. que la clase quede asignada a horarios de días diferentes; en ese caso, el movimiento se hará trasladando la hora de inicio de la clase al comienzo del día siguiente.
3. que la clase quede asignada a dos salones diferentes; por la forma de la solución, este caso está incluido en el anterior.

Taller V - 2002 <i>Asignación de horarios y Salones a Cursos</i>	Tipo de Documento: <i>Informe del Proyecto</i>	Finalizado:
	Versión: 1.0	Aprobado:
Preparado por: IC, JP, YU		Aprobado por:

Ejemplos:

Situación 1: para una clase X de una hora y media de duración, asignada inicialmente, al salón 001, los lunes a las 20:00 horas. En este caso, el movimiento deja toda la clase dentro del mismo día y salón, por lo cual queda cumplido el movimiento

	Clase X		Clase X	
001-lunes-20:00	1	→	001-lunes-20:00	0
001-lunes-20:30	1		001-lunes-20:30	1
001-lunes-21:00	1		001-lunes-21:00	1
001-lunes-21:30	0		001-lunes-21:30	1
001-martes-8:00	0		001-martes-8:00	0
001-martes-8:30	0		001-martes-8:30	0
001-martes-9:00	0		001-martes-9:00	0

Figura A3 - 2: Ajuste de un movimiento. Situación 1.

Situación 2: la clase X de una hora y media de duración, asignada inicialmente, al salón 001, los lunes a las 20:30 horas. En este caso, la clase queda asignada en dos días diferentes, por lo cual el movimiento tendrá que realizarse hasta la situación de la tercer tabla:

	Clase X		Clase X	
001-lunes-20:00	0	→	001-lunes-20:00	0
001-lunes-20:30	1		001-lunes-20:30	0
001-lunes-21:00	1		001-lunes-21:00	1
001-lunes-21:30	1		001-lunes-21:30	1
001-martes-8:00	0		001-martes-8:00	1
001-martes-8:30	0		001-martes-8:30	0
001-martes-9:00	0		001-martes-9:00	0

	Clase X
001-lunes-20:00	0
001-lunes-20:30	0
001-lunes-21:00	0
001-lunes-21:30	0
001-martes-8:00	1
001-martes-8:30	1
001-martes-9:00	1

Figura A3 - 3: Ajuste de un movimiento. Situación 2.

Taller V - 2002 <i>Asignación de horarios y Salones a Cursos</i>	Tipo de Documento: <i>Informe del Proyecto</i>	Finalizado:
	Versión: 1.0	Aprobado:
Preparado por: IC, JP, YU		Aprobado por:

Situación 3: la clase X de una hora y media de duración, asignada inicialmente, al salón 001, los viernes a las 20:30 horas. En este caso, la clase queda asignada a dos salones diferentes, pero por la forma de la solución también queda asignada en dos días diferentes, por lo cual el movimiento a realizar es análogo al caso anterior:

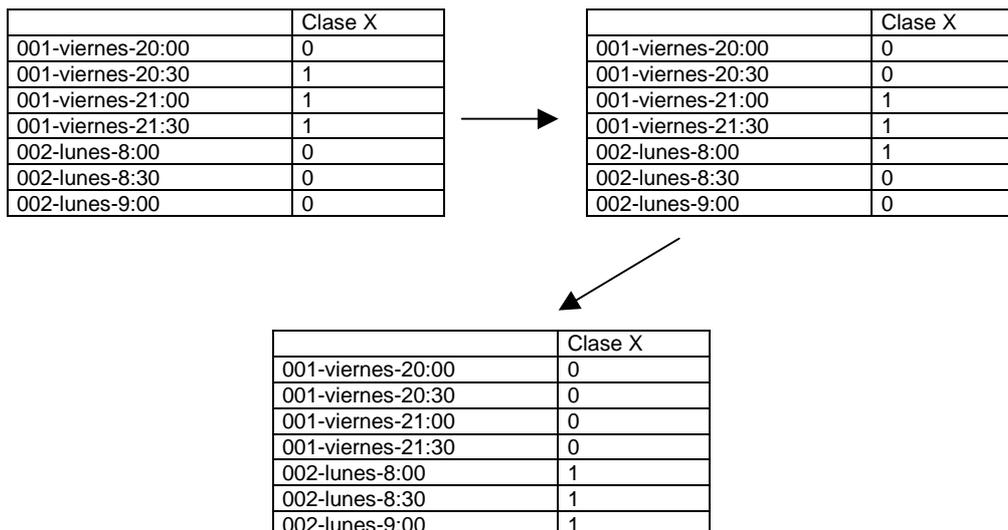


Figura A3 - 4: Ajuste de un movimiento. Situación 3.

Teniendo en cuenta la definición anterior de movimiento, se define una solución vecina de una solución s como cualquier solución que difiere en una asignación de clase con s .

Para poder llegar más rápido a buenas regiones del espacio de soluciones, se van a mover primero las clases que están en conflicto (es decir que no satisfacen alguna restricción). Mientras la mejor solución encontrada hasta el momento tiene superposiciones de clases, solamente se moverán las clases que no respetan las restricciones referidas a la superposición.

Cuando se llega a una solución sin superposiciones, puede moverse cualquier clase que tenga cualquier tipo de conflicto.

3.6.4. Tabú List

Para prevenir los ciclos, el método implementa una lista llamada Tabú List, que es la lista de los movimientos prohibidos de realizar. En esta lista se van guardando los últimos k movimientos aceptados (donde k es un parámetro a definir en la calibración del método)

En este caso, para lograr una mayor prevención a que se formen ciclos en las iteraciones del algoritmo, que llevarían a pérdidas importantes de tiempo de optimización, se definirán dos Tabú List, cada una cumplirá la siguiente función:

1. Se guardan las clases que se mueven
2. Se guardan las combinaciones clase/salón/horario que se mueven

Con este enfoque, considerando que $T1$ es el largo de la primer Tabú List y $T2$ el largo de la segunda, lo que sucederá es lo siguiente:

Taller V - 2002 <i>Asignación de horarios y Salones a Cursos</i>	Tipo de Documento: <i>Informe del Proyecto</i>	Finalizado:
	Versión: 1.0	Aprobado:
Preparado por: IC, JP, YU		Aprobado por:

Cuando una clase C se mueva a una pareja salón/horario (S,H), se introduce C en la Tabú List 1 y el trío (C,S,H) en la Tabú List 2. Esto significa que durante T1 iteraciones del Tabú Search la clase C no se puede mover y que durante T2 iteraciones del Tabú Search la clase C no puede ser movida hacia la pareja de salón/horario (S,H). La implementación de la segunda Tabú List tiene sentido solamente si $T1 < T2$.

Cuando el número de clases en conflicto es pequeño (menos de un 7% de la cantidad de clases que se manejan), la primer Tabú List puede prohibir los movimientos que llevan a una solución vecina adecuada. En este caso, no se tendrá en cuenta el status de tabú de un movimiento si es generado por la primer Tabú List.

3.6.5. Función de Aspiración

La función de aspiración indica el valor deseado para la función objetivo, y si al aplicar un movimiento se alcanza este valor, este puede realizarse a pesar de encontrarse en cualquiera de las Tabú List planteadas.

Se define la función de aspiración como:

$$A(f(s)) = f(s_{\text{mejor solución obtenida hasta el momento}})$$

o sea que un movimiento que tiene status de tabú por estar en alguna de las Tabú List solamente podrá ser aplicado si mejora el valor de la función objetivo de la mejor solución obtenida hasta el momento.

3.6.6. Calibración del Método

Durante las etapas de implementación y pruebas se definirán en base al comportamiento del algoritmo los siguientes parámetros:

- ◆ El largo de las Tabú-list.
- ◆ La cardinalidad del conjunto V de soluciones vecinas testeadas en cada iteración.
- ◆ El máximo de iteraciones Tsmax sin conseguir mejorar el valor de la función objetivo.

3.6.7. Criterio de parada

Se implementaron dos posibles criterios de parada para el algoritmo:

- La cantidad de iteraciones sin obtener una mejor solución a la actual, sea mayor que el parámetro Tsmax
- Parada solicitada explícitamente por el usuario.

Taller V - 2002 <i>Asignación de horarios y Salones a Cursos</i>	Tipo de Documento: <i>Informe del Proyecto</i>	Finalizado:
	Versión: 1.0	Aprobado:
Preparado por: IC, JP, YU		Aprobado por:

Anexo 4. Elección de la Plataforma

En este anexo se detallan los distintos productos evaluados, así como también las características que se tuvieron en cuenta para la elección de los mismos.

4.1 Sistema Operativo

El sistema operativo sobre el cual se va a implementar el producto deberá ser Windows en cualquiera de sus versiones, debido principalmente a que la frecuencia de uso del mismo es baja, y la máquina en la cual esté instalado deberá funcionar como estación de trabajo para los funcionarios de bedelía, los cuales utilizan dicho sistema operativo para su tarea diaria, por lo tanto no tienen conocimiento de otras plataformas. No obstante, es probable que en el futuro se migre a Linux, por lo cual se tomaron en cuenta las características de portabilidad.

4.2 Motor de Base de Datos

Si bien la cantidad de tablas que surgieron del análisis y diseño del programa es reducida, se estimó que algunas de ellas irán creciendo en forma constante semestre a semestre, lo cual implica hace necesaria la utilización de un motor de base de datos; un ejemplo de estas tablas es la que almacenará el histórico de asignaciones. Por otra parte, las funcionalidades que modifican la base de datos requieren en algunos casos manejo de transacciones, que en general ya está resuelto por el motor de base de datos.

Por lo especificado en el punto anterior, el motor de base de datos a utilizar debe poder ejecutarse sobre Windows, pero es deseable que pueda correr también en Linux.

Otro punto a tener en cuenta, es la complejidad del motor de base de datos a utilizar. Se llegó a la conclusión de que no se requiere la utilización de un motor de base de datos de grandes dimensiones, considerando los siguientes puntos:

- ◆ la dificultad del sistema se basa principalmente en el método de asignación, y no en el acceso a los datos
- ◆ se trata de un sistema monousuario, sin concurrencia de acceso.
- ◆ no hay requerimientos de seguridad de base de datos
- ◆ se debe poder ejecutar en un PC de escritorio.

5.2.1. Evaluación de Alternativas

Las características a evaluar de cada motor de base de datos son el licenciamiento, el soporte disponible, la performance, la robustez, y la portabilidad.

Taller V - 2002 <i>Asignación de horarios y Salones a Cursos</i>	Tipo de Documento: <i>Informe del Proyecto</i>	Finalizado:
	Versión: 1.0	Aprobado:
Preparado por: IC, JP, YU		Aprobado por:

MS Access

Característica	Calificación	Observaciones
Licenciamiento	Necesario	Actualmente existe licencia del producto en bedelía, ya que el sistema anterior es una base de datos Access.
Soporte disponible	Amplio	Se cuenta con ayuda del programa, y es posible obtenerla también en Internet.
Performance	Regular	Se comporta satisfactoriamente con cantidades pequeñas de datos, pero el desempeño empeora con el tamaño. Requiere la ejecución periódica del proceso de compactación para mantener una performance regular.
Robustez	Mala	Se corrompe fácilmente, y es necesario ejecutar los procesos de restauración con frecuencia.
Portabilidad	Mala	Solamente existe para sistemas operativos MS Windows

Tabla A4 - 1: Características de MS Access.

MySQL

Característica	Calificación	Observaciones
Licenciamiento	Gratuito	
Soporte disponible	Amplio	Se cuenta con ayuda del programa, y es posible obtenerla también en Internet.
Performance	Buena	La calificación se basa en opiniones encontradas en Internet y en el contacto realizado con el responsable de sistemas de una cadena de supermercados, en la cual utilizan dicho motor de base de datos, con resultados satisfactorios.
Robustez	Aceptable	La calificación se basa en opiniones encontradas en Internet y en el contacto realizado con el responsable de sistemas de una cadena de supermercados, en la cual utilizan dicho motor de base de datos, con resultados satisfactorios.
Portabilidad	Buena	Corre en MS Windows y en una amplia variedad de sistemas Unix, y los datos y estructuras pueden migrarse fácilmente a través de scripts.

Tabla A4 - 2: Características de MySQL.

Personal Oracle

Característica	Calificación	Observaciones
Licenciamiento	Necesario	No se cuenta con licencia en bedelía.
Soporte disponible	Medio	Se cuenta con ayuda del programa, y es posible obtenerla también en Internet, pero en menor escala que los anteriores.
Performance	Buena	La calificación se basa en opiniones encontradas en Internet y en la experiencia de uno de los integrantes del equipo de trabajo.
Robustez	Buena	La calificación se basa en opiniones encontradas en Internet y en la experiencia de uno de los integrantes del equipo de trabajo.
Portabilidad	No tiene	Solamente existe en versiones para Windows.

Tabla A4 - 3: Características de Personal Oracle.

5.2.2. Elección del Motor de Base de Datos

La posibilidad de utilizar Access se descartó considerando las calificaciones en cuanto a Performance, Robustez y portabilidad. Entre las dos posibilidades restantes, se definió la utilización de MySQL, dado que el licenciamiento es gratuito, cuenta con un amplio soporte y cumple los requisitos de portabilidad.

Taller V - 2002 <i>Asignación de horarios y Salones a Cursos</i>	Tipo de Documento: <i>Informe del Proyecto</i>	Finalizado:
	Versión: 1.0	Aprobado:
Preparado por: IC, JP, YU		Aprobado por:

4.3 Lenguajes de Programación

Debido a que la funcionalidad principal del sistema es el procesamiento automático de la asignación, se considerará importante para la elección del lenguaje de programación a utilizar. El algoritmo a implementar deberá realizar gran cantidad de cálculos, principalmente recorriendo estructuras de datos como vectores y matrices, y repetir este proceso varios cientos de veces hasta alcanzar una buena solución. Por lo tanto se hace necesario un lenguaje que permita crear y manejar eficientemente estas estructuras y que tenga gran potencia de cálculo. Se tendrá en cuenta además el soporte disponible, los ambientes de desarrollo, y la portabilidad.

4.3.1. Evaluación de Alternativas

Se manejaron dos alternativas: C o Java.

C

Característica	Calificación	Observaciones
Rapidez	Muy buena	Actualmente es el lenguaje más rápido para la manipulación de matrices y vectores.
Soporte disponible	Bueno	Se cuenta con variada bibliografía sobre el lenguaje, y es posible obtenerla también en Internet.
Ambiente de desarrollo	Buena	Se cuenta con ambientes amigables de desarrollo.
Portabilidad	Regular	Requiere al menos recompilación de los programas para adaptarlos a nuevas plataformas. En el caso de las interfaces gráficas, en general hay que rehacerlas.

Tabla A4 - 4: Características del lenguaje C.

Java

Característica	Calificación	Observaciones
Rapidez	Buena	Es eficiente para la manipulación de matrices y vectores, pero es más lento que C debido a que la ejecución es interpretada.
Soporte disponible	Bueno	Se cuenta con variada bibliografía sobre el lenguaje, y es posible obtenerla también en Internet.
Ambiente de desarrollo	Buena	Se cuenta con ambientes amigables de desarrollo.
Portabilidad	Muy buena	El hecho de que la ejecución sea interpretada hace que la portabilidad de los programas sea mayor.

Tabla A4 - 5: Características del lenguaje Java.

4.3.2. Elección del Lenguaje de Programación

Las características consideradas tienen calificaciones similares para ambos lenguajes, con excepción de la rapidez y la portabilidad. Se determina la utilización de Java dando prioridad a la portabilidad sobre la rapidez, teniendo en cuenta que:

- ◆ el proceso de asignación que es el único en el que la rapidez es crítica es de baja frecuencia
- ◆ aunque actualmente en bedelía se cuenta con Windows en todas las estaciones de trabajo, es probable que en el futuro se migre a Linux, con lo

Taller V - 2002 <i>Asignación de horarios y Salones a Cursos</i>	Tipo de Documento: <i>Informe del Proyecto</i>	Finalizado:
	Versión: <i>1.0</i>	Aprobado:
Preparado por: <i>IC, JP, YU</i>		Aprobado por:

cual la portabilidad pasa a tener mayor relevancia; esto se relaciona con lo especificado anteriormente respecto al costo de las herramientas.

Taller V - 2002 <i>Asignación de horarios y Salones a Cursos</i>	Tipo de Documento: <i>Informe del Proyecto</i>	Finalizado:
	Versión: 1.0	Aprobado:
Preparado por: IC, JP, YU		Aprobado por:

Anexo 5. Calibración y Resultados Obtenidos

5.1 Calibración del Tabú Search

Para cada parámetro a determinar, se tomaron dos valores candidatos que surgieron de pruebas informales y de la experiencia adquirida de observar el Tabú Search durante la etapa de implementación del mismo. De esta manera llegamos a que los valores candidatos para cada parámetro son los siguientes:

Parámetro	Valor Candidato 1	Valor Candidato 2
Cantidad de Soluciones Vecinas por Iteración (V)	10	25
Largo de Tabú List de clases (k1)	2	5
Largo de Tabú List de clase/salón/horario (k2)	10	20
Cantidad de iteraciones sin mejora la solución (Tmax)	8000	12000

Tabla A5 - 1: Valores candidato para los parámetros.

Las ejecuciones de todas las pruebas para determinar los parámetros tuvieron algunas características similares en cuanto a su comportamiento. Por lo general, durante los primeros minutos, se baja de forma considerable el costo de la solución inicial y luego la disminución del costo se estabiliza y empieza a bajar de forma más lenta, esto se puede apreciar en cada gráfica que acompaña a las siguientes pruebas.

5.1.1. Valor óptimo para (V) – Cantidad de Soluciones Vecinas

Para la determinación del valor óptimo de (V) se efectuaron cuatro ejecuciones del Tabú Search, dos de ellas con cada uno de los valores candidatos tomando como solución inicial la asignación del segundo semestre del año 2002 y una solución generada en forma aleatoria.

La cantidad de soluciones vecinas representa la cantidad de opciones que tiene el Tabú Search para elegir como próxima solución en una determinada iteración, por lo que en principio puede parecer que un valor mayor es mejor, pero en realidad implica más tiempo de ejecución y puede llevar a la generación de soluciones vecinas no relevantes para encontrar una solución. Esto justifica la ejecución de estas pruebas para encontrar el valor óptimo para este parámetro.

Taller V - 2002 <i>Asignación de horarios y Salones a Cursos</i>	Tipo de Documento: <i>Informe del Proyecto</i>	Finalizado:
	Versión: 1.0	Aprobado:
Preparado por: IC, JP, YU		Aprobado por:

5.1.1.1. Prueba 1 – Determinación de (V)

Datos de Entrada de la prueba

Entrada	Valor
Solución Inicial	2do. Semestre 2002
Cantidad de Soluciones Vecinas	10
Largo Tabú List Clases	2
Largo Tabú List Clase/Salón/Horario	10
Cantidad de Iteraciones sin Mejora	8000

Tabla A5 - 2: Determinación de (V). Datos de entrada de la prueba 1.

Visualización Gráfica de la Evolución del Costo de la Solución

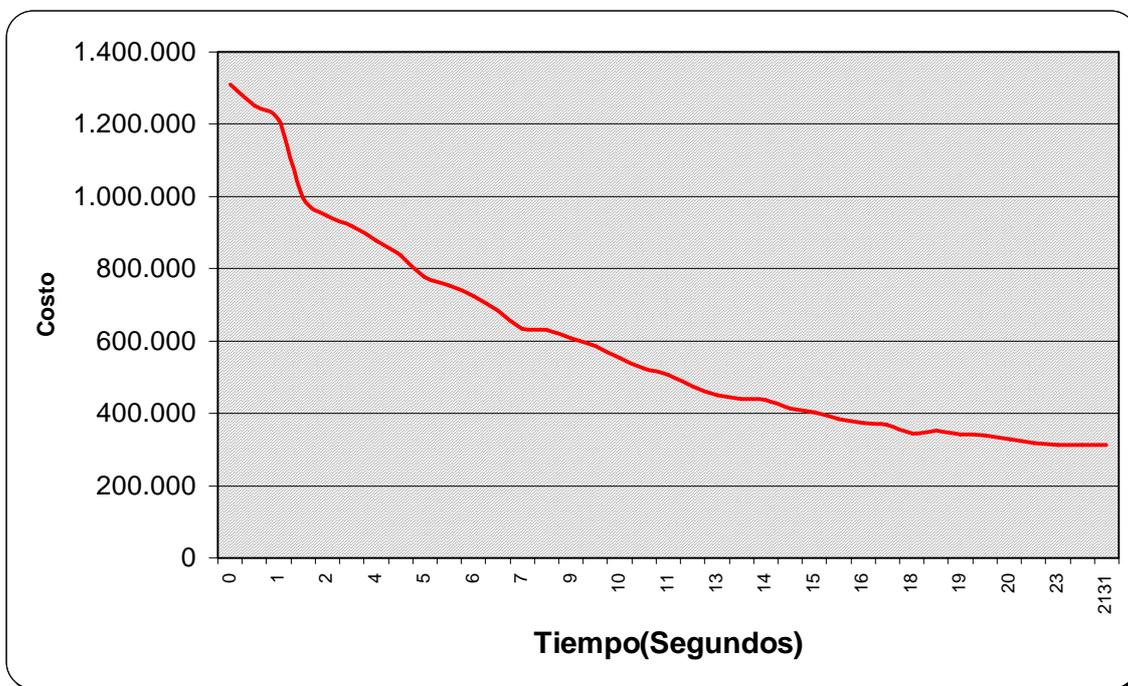


Figura A5 - 1: Determinación de (V). Visualización gráfica de la prueba 1.

Podemos observar que se llega en forma muy rápida (20 a 30 segundos) a la estabilización de los valores de la función objetivo, debido a que una cantidad baja de soluciones vecinas por iteración hace que el método funcione más rápidamente.

Taller V - 2002 <i>Asignación de horarios y Salones a Cursos</i>	Tipo de Documento: <i>Informe del Proyecto</i>	Finalizado:
	Versión: 1.0	Aprobado:
Preparado por: IC, JP, YU		Aprobado por:

5.1.1.2. Prueba 2 – Determinación de (V)

Datos de Entrada de la prueba

Entrada	Valor
Solución Inicial	2do. Semestre 2002
Cantidad de Soluciones Vecinas	25
Largo Tabú List Clases	2
Largo Tabú List Clase/Salón/Horario	10
Cantidad de Iteraciones sin Mejora	8000

Tabla A5 - 3: Determinación de (V). Datos de entrada de la prueba 2.

Visualización Gráfica de la Evolución del Costo de la Solución

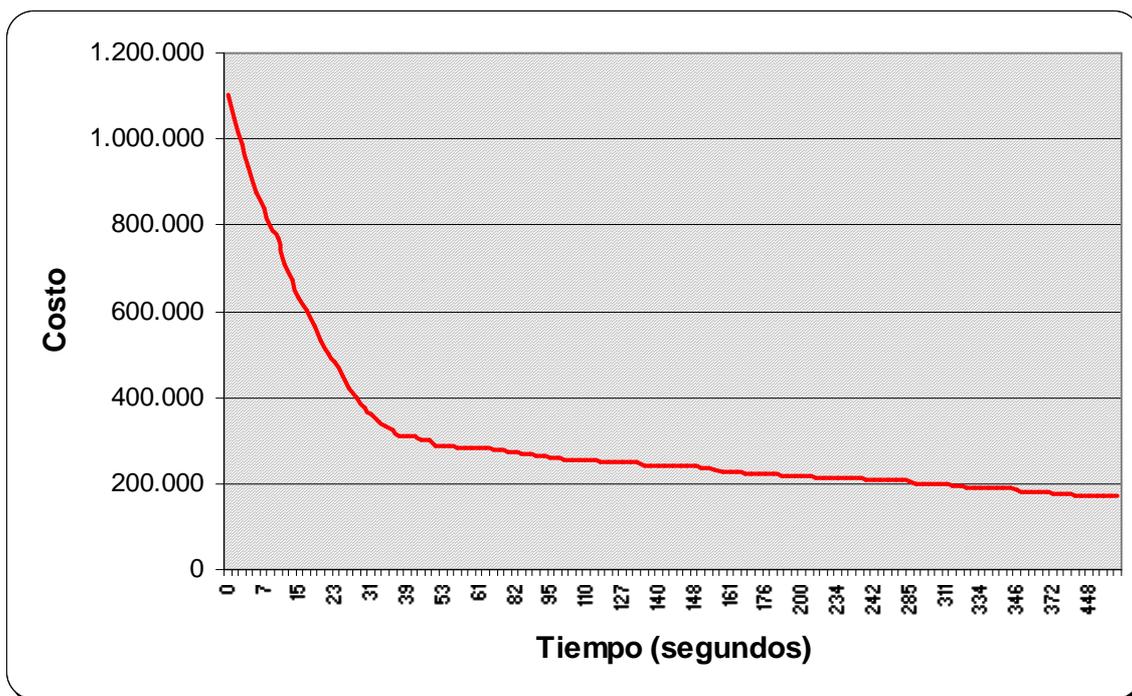


Figura A5 - 2: Determinación de (V). Visualización gráfica de la prueba 2.

La estabilización de los valores se logra entre los 30 y 50 segundos.

Taller V - 2002 <i>Asignación de horarios y Salones a Cursos</i>	Tipo de Documento: <i>Informe del Proyecto</i>	Finalizado:
	Versión: 1.0	Aprobado:
Preparado por: IC, JP, YU		Aprobado por:

5.1.1.3. Prueba 3 – Determinación de (V)

Datos de Entrada de la prueba

Entrada	Valor
Solución Inicial	Aleatoria
Cantidad de Soluciones Vecinas	10
Largo Tabú List Clases	2
Largo Tabú List Clase/Salón/Horario	10
Cantidad de Iteraciones sin Mejora	8000

Tabla A5 - 4: Determinación de (V). Datos de entrada de la prueba 3.

Visualización Gráfica de la Evolución del Costo de la Solución

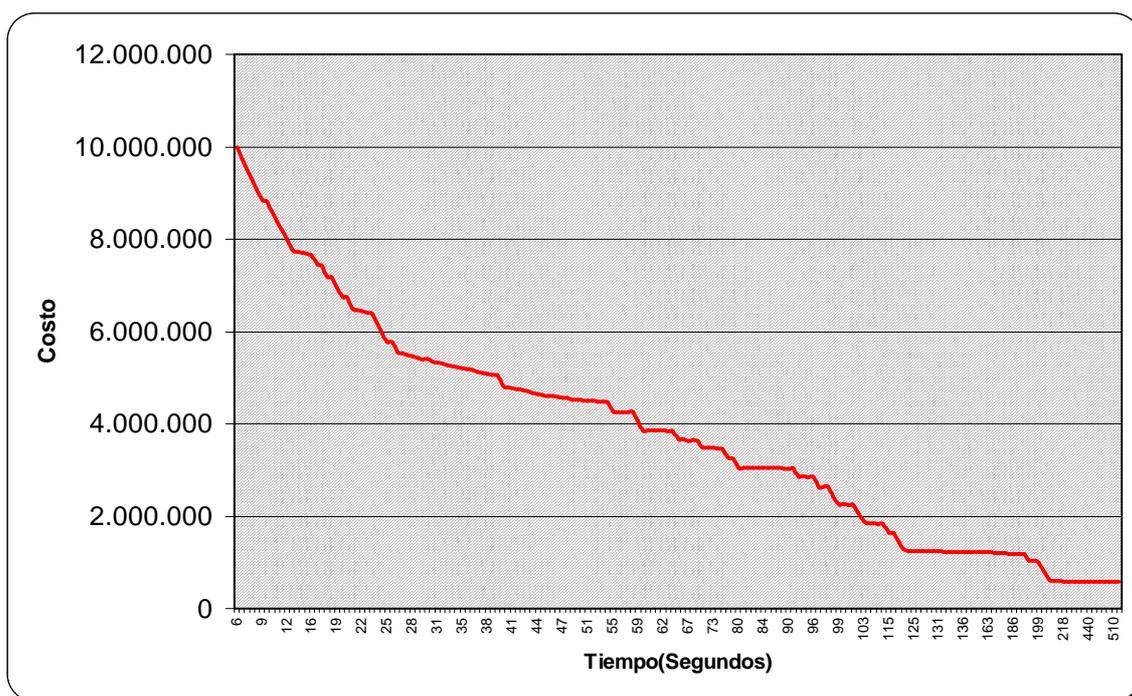


Figura A5 - 3: Determinación de (V). Visualización gráfica de la prueba 3.

En este caso, al ser una solución generada en forma aleatoria, los costos iniciales son muy altos, por lo que cuesta más llegar al período de estabilización, el mismo se logra recién a los 122 segundos.

Taller V - 2002 <i>Asignación de horarios y Salones a Cursos</i>	Tipo de Documento: <i>Informe del Proyecto</i>	Finalizado:
	Versión: 1.0	Aprobado:
Preparado por: IC, JP, YU		Aprobado por:

5.1.1.4. Prueba 4 – Determinación de (V)

Datos de Entrada de la prueba

Entrada	Valor
Solución Inicial	Aleatoria
Cantidad de Soluciones Vecinas	25
Largo Tabú List Clases	2
Largo Tabú List Clase/Salón/Horario	10
Cantidad de Iteraciones sin Mejora	8000

Tabla A5 - 5: Determinación de (V). Datos de entrada de la prueba 4.

Visualización Gráfica de la Evolución del Costo de la Solución

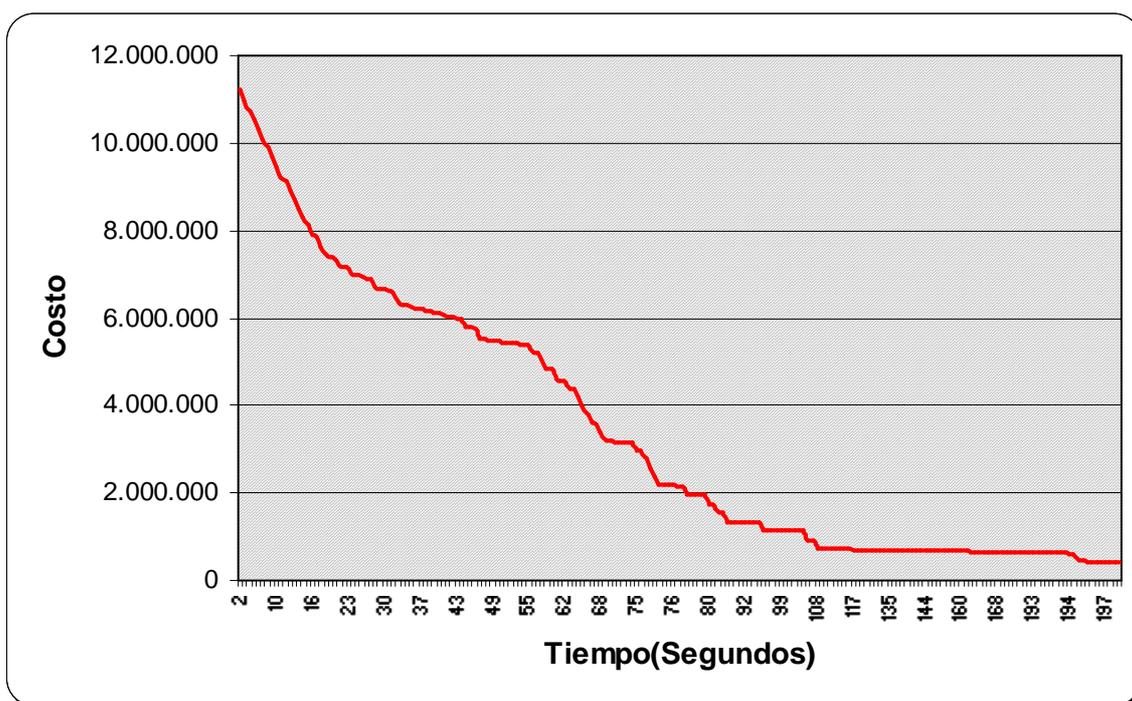


Figura A5 - 4: Determinación de (V). Visualización gráfica de la prueba 4.

Al igual que el caso anterior, como la solución inicial es aleatoria, los altos costos iniciales hacen que se demore en alcanzar la estabilización, esta se logra alrededor de los 110 segundos, por lo que se deduce que cuando se parte de soluciones con altos costos, la cantidad de vecinos tiene menos relevancia para lograr la estabilización de los valores.

Taller V - 2002 <i>Asignación de horarios y Salones a Cursos</i>	Tipo de Documento: <i>Informe del Proyecto</i>	Finalizado:
	Versión: 1.0	Aprobado:
Preparado por: IC, JP, YU		Aprobado por:

5.1.1.5. Elección de (V)

Las conclusiones que podemos obtener analizando los datos obtenidos de estas cuatro pruebas son las siguientes:

- ◆ Cuando se tienen 10 soluciones vecinas por iteración los costos bajan más rápidamente que con 25, hasta alcanzar la estabilización de los mismos. (En las pruebas 1 y 2 se demora 20 y 50 segundos respectivamente). Esto se debe a que al tener menos vecinos entre los cuales elegir el algoritmo funciona más rápidamente.
- ◆ Con 10 soluciones vecinas, luego de alcanzada la estabilización, al algoritmo le cuesta lograr bajas significativas de costos, pensamos que esto se debe a que como tiene menos opciones (vecinos) entre los cuales elegir puede quedar más tiempo “atrapado” en óptimos locales, mientras que con 25 soluciones vecinas esto no sucede y puede seguir bajando los costos de forma regular, navegando hacia regiones del espacio de soluciones de mejores costos.

Pensamos que es más importante para el algoritmo evitar el estancamiento cuando logra la estabilización que la velocidad con que se logra la misma, por lo tanto determinamos que el valor óptimo para la cantidad de vecinos es 25.

Taller V - 2002 <i>Asignación de horarios y Salones a Cursos</i>	Tipo de Documento: <i>Informe del Proyecto</i>	Finalizado:
	Versión: 1.0	Aprobado:
Preparado por: IC, JP, YU		Aprobado por:

5.1.2. Valor óptimo para (k1) y (k2) – Largos de las Tabú List

Para la determinación de los valores óptimos de (k1) y (k2) se efectuaron cuatro ejecuciones del Tabú Search, dos con cada uno de los valores candidatos tomando como solución inicial la asignación del segundo semestre del año 2002 y dos con una solución generada en forma aleatoria.

Si bien se trata de dos parámetros, estos se determinan de forma conjunta ya que las dos listas en realidad representan una única Tabú List que es la que figura en la forma genérica del Tabú Search, simplemente es una adaptación de la misma para lograr una mayor prevención a que se formen ciclos en las iteraciones del algoritmo.

5.1.2.1. Prueba 1 – Determinación de (k1) y (k2)

Datos de Entrada de la prueba

Entrada	VALOR
Solución Inicial	2do. Semestre 2002
Cantidad de Soluciones Vecinas	25
Largo Tabú List Clases	2
Largo Tabú List Clase/Salón/Horario	10
Cantidad de Iteraciones sin Mejora	12000

Tabla A5 - 6: Determinación de (k1) y (k2). Datos de entrada de la prueba 1.

Visualización Gráfica de la Evolución del Costo de la Solución

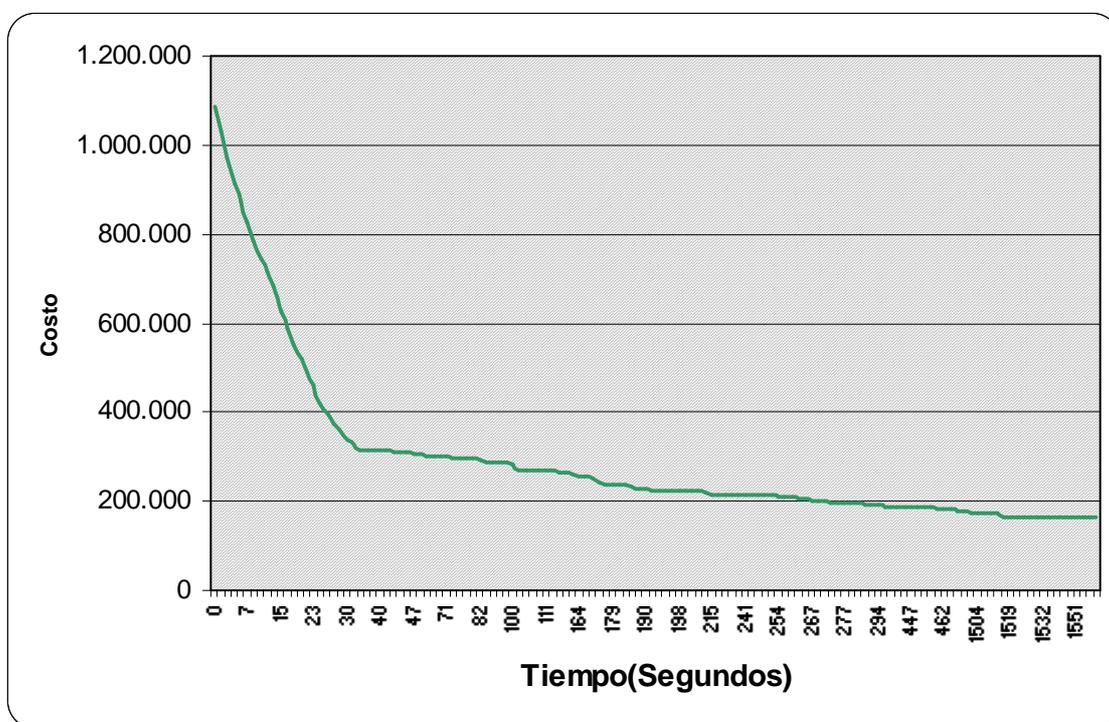


Figura A5 - 5: Determinación de (k1) y (k2). Visualización gráfica de la prueba 1.

La estabilización de los valores se logra aproximadamente a los 30 segundos de ejecución, luego los valores siguen bajando en forma regular.

Taller V - 2002 <i>Asignación de horarios y Salones a Cursos</i>	Tipo de Documento: <i>Informe del Proyecto</i>	Finalizado:
	Versión: 1.0	Aprobado:
Preparado por: IC, JP, YU		Aprobado por:

5.1.2.2. Prueba 2 – Determinación de (k1) y (k2)

Datos de Entrada de la prueba

Entrada	Valor
Solución Inicial	2do. Semestre 2002
Cantidad de Soluciones Vecinas	25
Largo Tabú List Clases	5
Largo Tabú List Clase/Salón/Horario	20
Cantidad de Iteraciones sin Mejora	12000

Tabla A5 - 7: Determinación de (k1) y (k2). Datos de entrada de la prueba 2.

Visualización Gráfica de la Evolución del Costo de la Solución

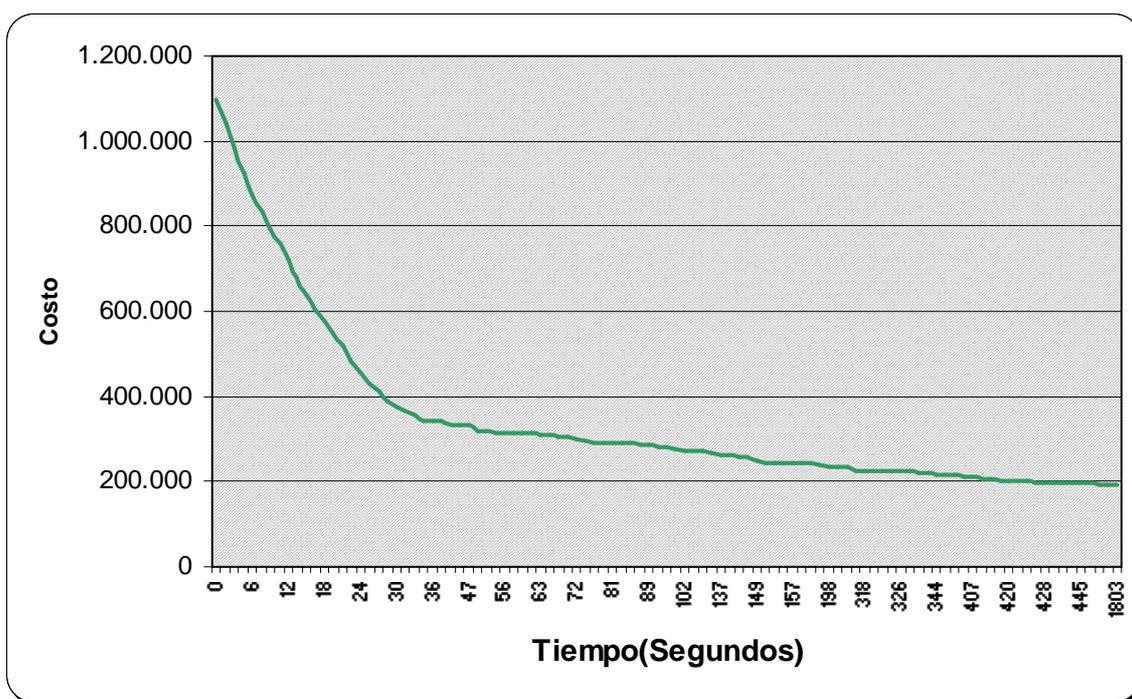


Figura A5 - 6: Determinación de (k1) y (k2). Visualización gráfica de la prueba 2.

Al igual que la prueba anterior, la estabilización de los valores se logra aproximadamente a los 30 segundos de ejecución, sin embargo luego parece no es tan sostenido el ritmo de bajar los valores como en la prueba 1.

Taller V - 2002 <i>Asignación de horarios y Salones a Cursos</i>	Tipo de Documento: <i>Informe del Proyecto</i>	Finalizado:
	Versión: 1.0	Aprobado:
Preparado por: IC, JP, YU		Aprobado por:

5.1.2.3. Prueba 3 – Determinación de (k1) y (k2)

Datos de Entrada de la prueba

Entrada	Valor
Solución Inicial	Aleatoria
Cantidad de Soluciones Vecinas	25
Largo Tabú List Clases	2
Largo Tabú List Clase/Salón/Horario	10
Cantidad de Iteraciones sin Mejora	12000

Tabla A5 - 8: Determinación de (k1) y (k2). Datos de entrada de la prueba 3.

Visualización Gráfica de la Evolución del Costo de la Solución

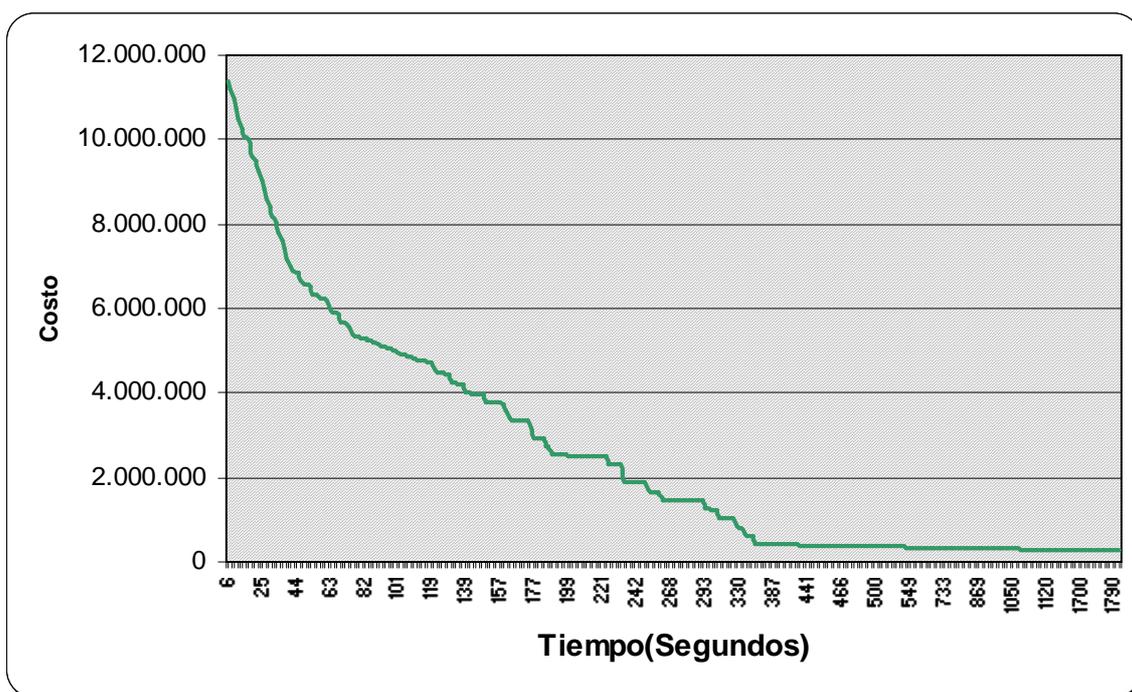


Figura A5 - 7: Determinación de (k1) y (k2). Visualización gráfica de la prueba 3.

La estabilización de los valores se logra aproximadamente a los 350 segundos de ejecución.

Taller V - 2002 <i>Asignación de horarios y Salones a Cursos</i>	Tipo de Documento: <i>Informe del Proyecto</i>	Finalizado:
	Versión: 1.0	Aprobado:
Preparado por: IC, JP, YU		Aprobado por:

5.1.2.4. Prueba 4 – Determinación de (k1) y (k2)

Datos de Entrada de la prueba

Entrada	Valor
Solución Inicial	Aleatoria
Cantidad de Soluciones Vecinas	25
Largo Tabú List Clases	5
Largo Tabú List Clase/Salón/Horario	20
Cantidad de Iteraciones sin Mejora	12000

Tabla A5 - 9: Determinación de (k1) y (k2). Datos de entrada de la prueba 4.

Visualización Gráfica de la Evolución del Costo de la Solución

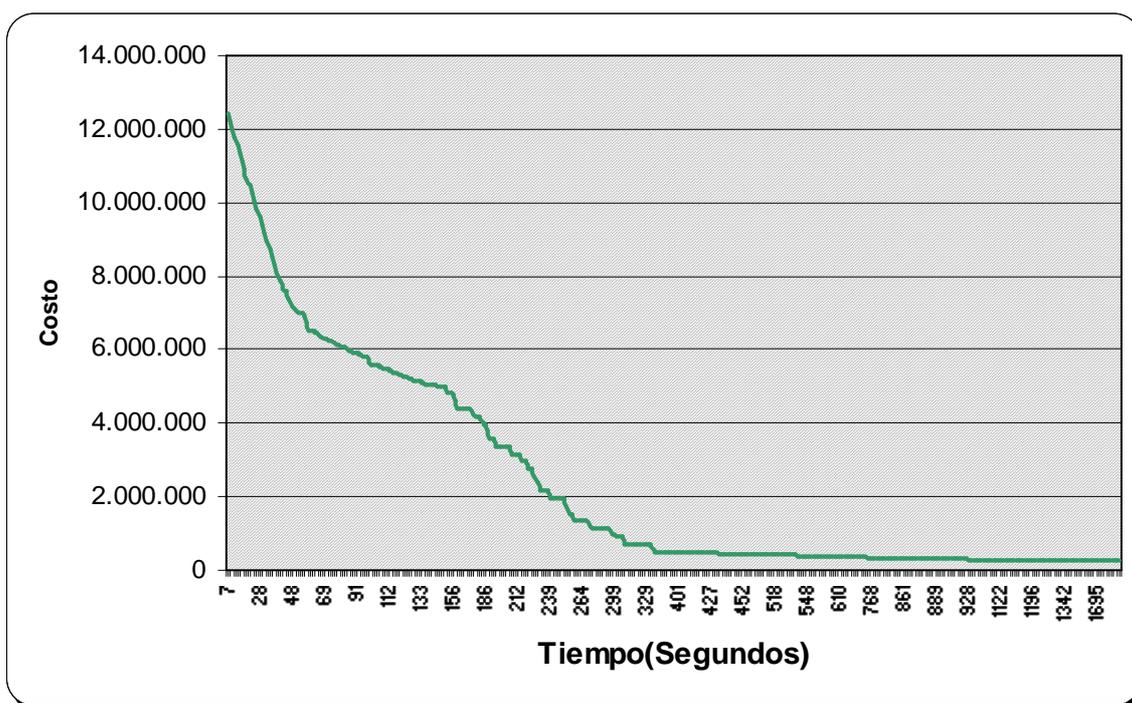


Figura A5 - 8: Determinación de (k1) y (k2). Visualización gráfica de la prueba 4.

La estabilización de los valores se logra aproximadamente a los 350 segundos de ejecución, al igual que en la prueba anterior. Los comportamientos han sido similares en las pruebas 3 y 4.

Taller V - 2002 <i>Asignación de horarios y Salones a Cursos</i>	Tipo de Documento: <i>Informe del Proyecto</i>	Finalizado:
	Versión: 1.0	Aprobado:
Preparado por: IC, JP, YU		Aprobado por:

5.1.2.5. Elección de (k1) y (k2)

Las conclusiones que podemos obtener analizando los datos obtenidos de estas cuatro pruebas son las siguientes:

- ◆ Los pares de pruebas ((1,2) y (3,4)) en los que variamos los largos de las listas partiendo de el mismo tipo de solución (ya sea 2do. semestre 2002 y aleatoria) tuvieron comportamientos muy similares, sobre todo las pruebas que tomaron una solución aleatoria como solución inicial.
- ◆ En las pruebas 1 y 2, o sea las que parten de la asignación del segundo semestre del año 2002 se puede observar una muy leve ventaja de la prueba 1 sobre la prueba 2, en el sentido de que luego de lograr la estabilización de los valores, sigue bajando los mismos en forma más regular y consistente que la prueba 2. Pensamos que esto puede deberse a que como los largos de las Tabú List en la prueba 2 son mayores, esto puede significar que tenemos más clases en estado de "Tabú", o sea más clases que no pueden ser reutilizadas mientras permanecen en estas listas; esto parece tener algo de influencia en la capacidad del algoritmo de escapar de los óptimos locales. Al parecer el algoritmo funciona un poco mejor siendo más permisivo.

Debido a que preferimos privilegiar la elección de los valores óptimos basados en las pruebas que parten de una asignación anterior ya que este seguramente sea el uso más común que se le de al mismo, seleccionamos como valores óptimos para estos parámetros los utilizados en la prueba 1, basándonos en el punto anterior, o sea que determinamos $k1 = 2$ y $k2 = 10$.

Taller V - 2002 <i>Asignación de horarios y Salones a Cursos</i>	Tipo de Documento: <i>Informe del Proyecto</i>	Finalizado:
	Versión: 1.0	Aprobado:
Preparado por: IC, JP, YU		Aprobado por:

5.1.3. Valor óptimo para (Tmax) - Cantidad de Iteraciones

El parámetro Tmax representa la cantidad de iteraciones que realiza el algoritmo sin mejorar la solución actual, o sea que cuando se alcance este valor el proceso debe terminar.

Para determinar este parámetro, se utilizaron pruebas análogas a las realizadas para los parámetros anteriores (partiendo de la solución del segundo semestre del 2002 y de una solución aleatoria), la diferencia radica en que para estas pruebas se utilizaron los valores óptimos determinados anteriormente para los demás parámetros (cantidad de soluciones vecinas y largos de las Tabú List).

5.1.3.1. Prueba 1 – Determinación de (Tmax)

Datos de Entrada de la prueba

Entrada	Valor
Solución Inicial	2do. semestre 2002
Cantidad de Soluciones Vecinas	25
Largo Tabú List Clases	5
Largo Tabú List Clase/Salón/Horario	10
Cantidad de Iteraciones sin Mejora	12000

Tabla A5 - 10: Determinación de (Tmax). Datos de entrada de la prueba 1.

Resultados Obtenidos

Resultado	Valor
Costo Inicial	1.184.984
Costo Final	177.185
% de Mejora	85.05 %
Tiempo insumido	5:31

Tabla A5 - 11: Determinación de (Tmax). Resultados obtenidos en la prueba 1.

5.1.3.2. Prueba 2 – Determinación de (Tmax)

Datos de Entrada de la prueba

Entrada	Valor
Solución Inicial	2do. semestre 2002
Cantidad de Soluciones Vecinas	25
Largo Tabú List Clases	5
Largo Tabú List Clase/Salón/Horario	10
Cantidad de Iteraciones sin Mejora	8000

Tabla A5 - 12: Determinación de (Tmax). Datos de entrada de la prueba 2.

Resultados Obtenidos

Resultado	Valor
Costo Inicial	1.184.984
Costo Final	161.944
% de Mejora	86.34 %
Tiempo insumido	2:10

Tabla A5 - 13: Determinación de (Tmax). Resultados obtenidos en la prueba 2.

Taller V - 2002 <i>Asignación de horarios y Salones a Cursos</i>	Tipo de Documento: <i>Informe del Proyecto</i>	Finalizado:
	Versión: 1.0	Aprobado:
Preparado por: IC, JP, YU		Aprobado por:

5.1.3.3. Prueba 3 – Determinación de (Tmax)

Datos de Entrada de la prueba

Entrada	Valor
Solución Inicial	Aleatoria
Cantidad de Soluciones Vecinas	25
Largo Tabú List Clases	5
Largo Tabú List Clase/Salón/Horario	10
Cantidad de Iteraciones sin Mejora	12000

Tabla A5 - 14: Determinación de (Tmax). Datos de entrada de la prueba 3.

Resultados Obtenidos

Resultado	Valor
Costo Inicial	10.708.352
Costo Final	198.795
% de Mejora	98.15 %
Tiempo insumido	12:50

Tabla A5 - 15: Determinación de (Tmax). Resultados obtenidos en la prueba 3.

5.1.3.4. Prueba 4 – Determinación de (Tmax)

Datos de Entrada de la prueba

Entrada	Valor
Solución Inicial	Aleatoria
Cantidad de Soluciones Vecinas	25
Largo Tabú List Clases	5
Largo Tabú List Clase/Salón/Horario	10
Cantidad de Iteraciones sin Mejora	8000

Tabla A5 - 16: Determinación de (Tmax). Datos de entrada de la prueba 4.

Resultados Obtenidos

Resultado	Valor
Costo Inicial	11.371.913
Costo Final	208.325
% de Mejora	98.17 %
Tiempo insumido	5:52

Tabla A5 - 17: Determinación de (Tmax). Resultados obtenidos en la prueba 4.

5.1.3.5. Elección de (Tmax)

Las conclusiones que podemos obtener analizando los datos obtenidos de estas cuatro pruebas son las siguientes:

- ◆ Analizando los resultados de los pares de pruebas ((1,2) y (3,4)), los cuales parten de distintas soluciones iniciales (segundo semestre del 2002 y aleatoria) observamos que en ambos casos la prueba con 8.000 iteraciones superó levemente el porcentaje de mejora obtenido por la prueba con 12.000

Taller V - 2002 <i>Asignación de horarios y Salones a Cursos</i>	Tipo de Documento: <i>Informe del Proyecto</i>	Finalizado:
	Versión: 1.0	Aprobado:
Preparado por: IC, JP, YU		Aprobado por:

- ♦ Como era lógico de esperar, las pruebas en las que se utilizaron 12.000 iteraciones sin mejorar la solución actual demoraron considerablemente más que las pruebas con 8.000 iteraciones, aunque el factor tiempo en realidad no tendría demasiada importancia debido a la baja frecuencia de uso del algoritmo de optimización.
- ♦ Como podemos observar en las gráficas de las pruebas para determinar los anteriores parámetros, las mejoras más significativas en costos de la función objetivo se obtienen en los primeros 5 minutos de ejecución del Tabú Search, luego los valores se estabilizan y las mejoras son escasas con respecto a lo logrado hasta ese momento.

Teniendo en cuenta los puntos anteriores, creemos que no es necesario que el algoritmo se ejecute durante un período largo de tiempo, ya que las mejoras son inmediatas. Inclusive el usuario podría detener el proceso luego de los primeros minutos de ejecución, hacer modificaciones en forma manual y volver a ejecutar a partir de ese punto el Tabú Search. Asimismo, el usuario puede ejecutar el algoritmo todas las veces que desee, lo cual implica que si pretende ejecutar el algoritmo durante más tiempo puede volver a ejecutarlo. Por todo lo expresado, determinamos que el valor óptimo para la cantidad de iteraciones sin mejorar la solución actual es 8.000.

5.2 Pesos utilizados para Restricciones y Preferencias

Como expresamos anteriormente, los valores óptimos de los pesos de las restricciones y preferencias no dependen del volumen de datos utilizado ni del tiempo de proceso de asignación del algoritmo. Estos valores se determinaron en conjunto con los funcionarios de bedelía dando prioridad a aquellas restricciones y preferencias que son fundamentales para la asignación de clases.

Los valores utilizados pueden ser cambiados si cambian las prioridades, o se pueden agregar o quitar restricciones y/o preferencias. Esto último implica un esfuerzo mayor a cambiar los valores, ya que hay que incorporarlas a la función objetivo del algoritmo de optimización.

En las siguientes tablas vemos los valores actuales:

Restricciones:

Restricción	Descripción	Peso
3	Una pareja (horario, salón) debe asignarse a una única clase.	5000
5	Las clases de un mismo semestre, carrera, y grupo no pueden superponerse.	400
8	Los grupos que tienen un bloque especificado solo pueden asignarse dentro de la franja horaria correspondiente al bloque.	5000
9	No pueden dictarse dos clases de un mismo grupo y asignatura un mismo día.	100000

Tabla A5 - 18: Pesos utilizados para las restricciones.

Nota: Las restricciones no incluidas en la tabla son las que se cumplen implícitamente, como se especificó en Decisiones de Diseño - Implementación del Tabú Search.

Taller V - 2002 <i>Asignación de horarios y Salones a Cursos</i>	Tipo de Documento: <i>Informe del Proyecto</i>	Finalizado:
	Versión: 1.0	Aprobado:
Preparado por: IC, JP, YU		Aprobado por:

Preferencias:

Preferencia	Descripción	Peso
2	Las clases deberían asignarse dentro del rango horario de preferencia especificado.	4000
3	Una clase no debería asignarse a un salón con menor capacidad de la requerida.	5
4	Es deseable que el salón asignado a una clase cuente con los recursos necesarios para el dictado de la misma.	50
6	No sería bueno que se utilizaran salones de gran capacidad para clases de pocos alumnos.	1
7	Es conveniente que las clases teóricas de un grupo se repartan en forma uniforme durante la semana.	150
8	Similitud de horarios en las distintas clases teóricas de un grupo.	10

Tabla A5 - 19: Pesos utilizados para las preferencias.

Notas:

1. La preferencia 1 no está incluida en la tabla porque se cumple implícitamente, como se especificó en Decisiones de Diseño - Implementación del Tabú Search.
2. La preferencia 5 no está incluida en la tabla porque no fue implementada, como se especificó en Decisiones de Diseño - Implementación del Tabú Search
3. En la implementación, los pesos de las restricciones 3 y 6 se multiplican por la cantidad de lugares faltantes o sobrantes en el salón
En la implementación, el peso de la restricción 8 se multiplica por la cantidad de clases del grupo cuyos horarios no sean similares.

Taller V - 2002 <i>Asignación de horarios y Salones a Cursos</i>	Tipo de Documento: <i>Informe del Proyecto</i>	Finalizado:
	Versión: <i>1.0</i>	Aprobado:
Preparado por: <i>IC, JP, YU</i>		Aprobado por:

Apéndices

Apéndice A. Documento de Análisis

Apéndice B. Documento de Diseño

Apéndice C. Manual de Usuario