



## MODELOS DE PROGRAMACIÓN ENTERA PARA UN PROBLEMA DE PROGRAMACIÓN DE HORARIOS PARA UNIVERSIDADES

### MODELS OF INTEGER PROGRAMMING FOR AN UNIVERSITY TIMETABLING PROBLEM

Andrés Saldaña Crovo<sup>1</sup>    Cristian Oliva San Martín<sup>2</sup>    Lorena Pradenas Rojas<sup>3</sup>

Recibido 13 de marzo de 2007, aceptado 28 de agosto de 2007

Received: March 13, 2007    Accepted: August 28, 2007

#### RESUMEN

En esta investigación se formulan dos modelos de Programación Lineal Entera para un problema de *Programación de Horarios para Universidades* y se presentan dos estrategias de solución para cada uno de ellos. El problema consiste en programar las asignaturas a ser dictadas, considerando los profesores, días, horarios, aulas y la necesidad de dictar las asignaturas en periodos consecutivos determinados. El objetivo es minimizar la asignación en periodos no deseados, balanceando la carga de trabajo diaria para cada grupo de alumnos. Las estrategias de solución combinan modelos de asignación directa a aulas o asignación a tipos de aulas. Las estrategias de solución que consideran relajación de restricciones, permiten resolver problemas de gran tamaño, a niveles de calidad razonables y utilizando pequeños tiempos computacionales. Los enfoques fueron aplicados a instancias de la Facultad de Ingeniería de la Universidad de Concepción, Chile. Los modelos utilizados en esta investigación pueden ser aplicados a una gran cantidad de problemas de *Programación de Horarios en Universidades*, proporcionando una gran flexibilidad de resolución.

Palabras clave: Problemas de programación de horarios para universidades, programación lineal entera, optimización combinatoria.

#### ABSTRACT

*In this research, two models of Integer Programming for a University Timetabling Problem are formulated and two solution strategies for each model are presented. The problem consists of programming the courses to be taught, considering teaching faculty, days, periods, classrooms and requirements for courses that are taught in consecutive periods. The objective is to minimize the allocation of undesired time slots as well as balancing the daily workload for each group of students. The solution strategies are based on either combining models of direct allocation to classrooms or types of classrooms. The solution strategies that include reducing constraints, allow solving big problems with reasonable levels of efficiency, using the computer system for a short time. The approaches were used with real data obtained from the Facultad de Ingeniería of the Universidad de Concepción, Chile. The models used in this research can be used in more general University Timetabling Problems, providing a great flexibility of resolution.*

Keywords: University timetabling problems, integer programming, combinatorial optimization.

#### INTRODUCCIÓN

Los Problemas de Programación de Horarios consisten, básicamente, en generar horarios para tareas definidas, buscando cumplir de la mejor manera con condiciones

y requerimientos específicos. Estos problemas son muy comunes y se encuentran en distintos tipos de actividades tales como: *Actividades Educativas*, Universidades, Colegios, Institutos, Facultades, Departamentos, *Actividades Deportivas*, *Actividades de Transporte*

<sup>1</sup> Departamento de Planeamiento de la Producción. Compañía Siderúrgica Huachipato S.A. Avenida Gran Bretaña 2910. Talcahuano, Chile. E-mail: andres.saldana@csh.cl

<sup>2</sup> Departamento de Ingeniería Industrial. Facultad de Ingeniería. Universidad de Concepción. Casilla 160-C. Correo 3. Concepción, Chile. E-mail: oliva@udec.cl

<sup>3</sup> Departamento de Ingeniería Industrial. Facultad de Ingeniería. Universidad de Concepción. Casilla 160-C. Correo 3. Concepción, Chile. E-mail: lpradena@udec.cl



y *Actividades* que involucren *personas o equipos de trabajo*.

El problema de Programación de Horarios en Universidades consiste en programar en un Horizonte de Planificación (generalmente una semana) las asignaturas que se dictan en un Periodo Académico determinado (año, trimestre o semestre), para las distintas asignaturas que las requieren, considerando los profesores necesarios en cada asignatura, los grupos de alumnos que toman un conjunto de asignaturas, los días o periodos disponibles, las aulas requeridas de tal manera de satisfacer un conjunto de restricciones relacionadas con la organización del sistema educacional.

No existe una sola clasificación para los Problemas de *Timetabling Educativos* [4, 9 y 20]. En general, cada una de las clasificaciones propuestas define distintas decisiones que se deben tomar para generar una correcta gestión de la enseñanza dentro de una *Unidad Académica*. A continuación se definen dos subproblemas de programación que se encuentran en las *Unidades Académicas* y que tienen directa relación con el presente trabajo [20]:

**Programación de Clases-Profesores (Class-Teacher Timetabling):** Este problema considera un conjunto de clases y un conjunto de profesores. El problema consiste en asignar periodos de tiempo para cada clase y profesor. Para cada par clase-profesor ya asignado, ningún profesor y ninguna clase pueden estar asignados a más de una asignatura en un mismo periodo.

**Asignación de Aulas (Rooms Assignment):** Dada una Programación de Class-Teacher Timetabling, cada asignatura debe ser asignada a aulas considerando las factibilidades de capacidad de alumnos o requerimientos de equipamiento en cada una. Ningún aula debe ser usada en un mismo periodo por más de una asignatura.

Los problemas de Programación de Horarios en Universidades han sido ampliamente estudiados en la Literatura, entre ellos [4, 6, 7, 9 y 13].

Problemas resueltos mediante Programación Entera se presentan en [3, 10, 11, 16, 18, 19 y 23].

Muchos problemas de Timetabling Educativos han sido tratados mediante la aplicación de Heurísticas y Metaheurísticas. Lectores que desean revisar estos métodos, entre ellos [8, 15, 21 y 24]. Por último, el enfoque de programación por restricciones, para este problema, se presenta en [1, 14, 17 y 25], entre otros.

En este trabajo se resuelven problemas reales de Programación de Horarios mediante Técnicas de Programación Lineal Entera. El problema se puede definir como un problema de **Class-Teacher Timetabling con Asignación de Aulas**. Se realizan dos formulaciones distintas y se generan dos enfoques de solución, obteniendo un total de 4 métodos de resolución. Una primera formulación asigna directamente las programaciones a salas de clases específicas, la segunda formulación genera una programación que, en una primera etapa, realiza las asignaciones a Tipos de Salas, con similares o iguales condiciones técnicas y/o de capacidad de alumnos, para en una segunda etapa, realizar las asignaciones de Tipos de Salas, a Salas específicas que pertenecen a cada Tipo. De igual manera, un primer enfoque considera todos los conjuntos de restricciones relacionados con los modelos generados y un segundo enfoque relaja conjuntos de restricciones, obteniendo un modelo inicial de mayor facilidad de resolución y, en una segunda etapa, fijar algunas variables obtenidas, para resolver para cada día que se debe programar un problema de menor tamaño<sup>4</sup>.

## DESCRIPCIÓN DEL PROBLEMA

El *Problema* a modelar consiste en obtener una *Programación de Horarios* para cada una de las asignaturas que intervienen, considerando cada una de las restricciones que tiene el problema y buscando minimizar el número total de periodos asignados en espacios de tiempo no deseados<sup>5</sup>. Las asignaturas se encuentran dentro de grupos o clases y los profesores ya tienen asignada cada una de sus asignaturas a dictar. Las asignaturas se deben asignar a periodos y aulas permitidas, en que los periodos pertenecen a los días en que se puede dictar cada asignatura. Cada asignatura tiene preestablecido el/los profesor/es que la/s dicta/n, una cantidad de periodo a asignar y el tamaño o cantidad de periodos consecutivos (bloques de periodos) en que debe estar dividida la asignatura. El *Modelo* también considera que pueden existir asignaciones previas para que algunas asignaturas sean dictadas en periodos y/o aulas

<sup>4</sup> Es importante aclarar que este trabajo se enmarca en el proyecto DIUC N° 205.093.010-1.0 que tiene por objetivo enfrentar el problema utilizando diversas herramientas provenientes de la investigación de operaciones y de la inteligencia artificial. Por lo anterior, este artículo ilustra específicamente los resultados obtenidos vía programación lineal entera.

<sup>5</sup> Los periodos no deseados se consideran como los periodos de almuerzo y los últimos periodos del día, aunque también los periodos no deseados pueden estar determinados por profesores, alumnos integrantes de asignaturas, grupos o *Unidades Académicas*.

específicas. Por último, se considera que todas las aulas están disponibles en todos los periodos del *Horizonte de Planificación*. A continuación se explica cada una de las restricciones que tiene el modelo:

#### Restricciones fuertes:

- No puede existir más de una asignación en un mismo periodo para: una misma asignatura o un mismo grupo de asignaturas (clases), un mismo profesor y una misma aula.
- La cantidad total de periodos asignados para una asignatura debe ser igual a la cantidad programada.
- Algunas asignaturas tienen que ser programadas en periodos consecutivos y divididas en periodos determinados (bloques).
- Pueden existir asignaciones previas de asignaturas, a un determinado periodo y aula.
- Cada asignatura y profesor puede ser asignado sólo en algunos periodos.
- Cada profesor ya tiene establecidas las asignaturas que dicta.
- Las asignaturas están divididas en periodos consecutivos determinados. Por ejemplo, la asignatura de *Control de la Producción* consta de 5 periodos (horas) que están divididos en 2 multiperiodos: el primero de 3 periodos consecutivos y el segundo de 2 periodos consecutivos.
- Las asignaturas sólo pueden ser realizadas en algunas aulas o laboratorios. Estas razones son debidas a características de capacidad de las aulas y a necesidades de requerimientos técnicos (proyectores, computadores, etc.).
- Para las asignaturas que consideran periodos consecutivos cada periodo consecutivo debe estar asignado en la misma aula. Por ejemplo, si la asignatura de *Control de la Producción* en su primer periodo de 3 multiperiodos consecutivos es asignada al periodo N° 4 al aula N° 3, los siguientes 2 periodos (5 y 6) deben estar asignados también en el aula N° 3.
- En un mismo día, cada asignatura puede ser asignada a lo más un multiperiodo. Por ejemplo, la asignatura de *Simulación* tiene 8 periodos, divididos en 2 multiperiodos de 3 horas consecutivas y 1 multiperiodo de 2 horas consecutivas. Si para esta asignatura se asignan para el día lunes 3 horas consecutivas, en el mismo día lunes no se pueden asignar los otros multiperiodos de 3 o 2 horas consecutivas. Esta restricción es para asegurar que una misma asignatura no se va a dictar de manera completa (todas sus divisiones de periodos o multiperiodos) en un mismo día.

#### Restricciones débiles:

- Una *Programación* es de mejor calidad si la carga de trabajo diaria para alumnos es equilibrada. Por lo tanto, para un mismo grupo y un día determinado, existe un número máximo de periodos a asignar<sup>6</sup>.
- Pueden existir prioridades o penalidades sobre asignaturas para que sean dictadas en días y periodos determinados. Es decir, una *Programación* será de mayor calidad mientras se cumplan de mejor manera las prioridades o se tenga la menor cantidad de asignaciones a periodos penalizados.

En este trabajo el objetivo es encontrar una *Programación* que *satisfaga* todas las *Restricciones Fuertes* y *minimice* la violación de las *Restricciones Débiles*.

### NOTACIÓN Y DEFINICIÓN

A continuación, se definen los conjuntos y parámetros necesarios para formular el problema:

- Sea  $C = \{1, \dots, \bar{c}\}$  el conjunto de asignaturas a programar.
- Sea  $G = \{1, \dots, \bar{g}\}$  el conjunto de clases o de grupos de asignaturas. Cada grupo  $g \in G$  contiene un conjunto de asignaturas  $C_g \subseteq C$ , que no pueden ser programadas en un mismo periodo de clases. Dos clases distintas pueden tener asignaturas iguales, por ejemplo, sea  $C_{g_1}$  y  $C_{g_2}$  dos asignaturas pertenecientes a dos grupos  $G_1$  y  $G_2$ , con  $G_1 \neq G_2$  puede ocurrir que  $C_{g_1} \cap C_{g_2} \neq \emptyset$ .
- Sea  $T = \{1, \dots, \bar{t}\}$  el conjunto de periodos de tiempo en que se deben programar las asignaturas en el *Horizonte de Planificación*.
- Sea  $D = \{1, \dots, \bar{d}\}$  el conjunto de días pertenecientes al *Horizonte de Planificación*.
- Sea  $TD_d$  el conjunto de los periodos  $t \in T$  que pertenecen al día  $d \in D$ .
- Sea  $S = \{1, \dots, \bar{s}\}$  el conjunto de profesores que deben dictar alguna de las asignaturas  $c \in C$  a programar.
- Sea  $R = \{1, \dots, \bar{r}\}$  el conjunto de aulas en las que se puede programar alguna asignatura  $c \in C$ .
- Sea  $SC_c$  el conjunto de los profesores  $s \in S$  que dictan la asignatura  $c \in C$ .
- Sea  $RC_c$  el conjuntos de aulas  $r \in R$  disponibles para dictar la asignatura  $c \in C$ .

<sup>6</sup> El número máximo de periodos a asignar para un mismo día se considera como un parámetro. Esta restricción es considerada como débil porque el parámetro puede ser variado, haciendo aumentar o disminuir la calidad de la solución.

- Sea  $TC_c$  el conjunto de periodos  $t \in T$  en los que se puede dictar la asignatura  $c \in C$ .
- Sea  $TS_s$  el conjunto de periodos en que el profesor  $s \in S$  puede dictar alguna asignatura.
- Sea  $AP = (c,r,t) \in C \times RC_c \times TC_c / (c,r,t)$  el conjunto de asignaciones previas para cada asignatura  $c \in C$  preasignada al aula  $r \in RC_c$  en el periodo  $t \in TC_c$ .

**Parámetros**

- Para cada asignatura  $c \in C$  sea  $n_c$  el número de horas a ser programadas por semana para la asignatura  $c$ .
- Sea  $HC_c = \{h_1, \dots, h_c\}$  el conjunto que indica el tamaño de cada periodo o multiperiodo en que debe dictarse la asignatura  $c \in C$ . Por ejemplo, sea  $c = 1$  y sea  $H_1 = (3,2,1)$ . Esto significa que el curso 1 contiene 6 periodos a asignar dividido en 3 multiperiodos, el primero de 3 periodos consecutivos, el segundo de 2 y el tercero de 1.
- Sea  $Lmax_{gd}$  el máximo número de periodos a dictar en el día  $d \in D$  para el grupo  $g \in G^7$ .
- Sea  $p_{ct}$  una penalidad para la asignatura  $c \in C$ , si es programada en el periodo  $t \in TC_c$ .

A partir de los conjuntos de entrada definidos anteriormente, es posible, mediante algoritmos simples, obtener otros conjuntos necesarios para generar el modelo:

- Sea  $CRT_{rt}$  el conjunto de las asignaturas que se pueden dictar en el aula  $r \in R$  y en el periodo  $t \in T$ . Este conjunto se obtiene a partir de  $RC_c$  y  $TC_c$ , para toda asignatura  $c \in C$ .
- Sea  $DC_c$  el conjunto de días en que se puede dictar la asignatura  $c \in C$ . Este conjunto se obtiene a partir de  $TD_d$  y  $TC_c$ , para todo día  $d \in D$ .
- Sea  $CGD_{gd}$  el conjunto de asignaturas que se pueden dictar para el grupo  $g \in G$  en el día  $d \in D$ . Este conjunto se obtiene a partir de  $CG_g$ ,  $TD_d$  y  $TC_c$  para toda asignatura  $c \in C$ .
- Sea  $CST_{st}$  el conjunto de asignaturas  $c \in C$  que dicta el profesor  $s \in S$  y factibles de ser realizadas en el periodo  $t \in T$ . Este conjunto se obtiene a partir de  $SC_s$ ,  $TC_c$  y  $TS_s$ .
- Sea  $TCD_{cd}$  el conjunto de los periodos en que se puede dictar la asignatura  $c \in C$  el día  $d \in D$ . Este conjunto se obtiene a partir de  $TD_d$  y  $TC_c$ .

- Sea  $CGT_{gt}$  el conjunto de asignaturas de  $c \in CG_g$  que se pueden dictar en el periodo  $t \in T$ . Este conjunto se obtiene a partir de  $CG_g$  y  $TC_c$ .
- Sea  $MCD_{cd}$  el conjunto de todos los intervalos en que se puede dictar la asignatura  $c \in C$  el día  $d \in DC_c$ . Cada elemento (intervalo)  $m \in MCD_{cd}$  tiene un conjunto  $TMCD_{mcd}$  de todos los periodos que pertenecen al intervalo  $m$ . Este conjunto se obtiene a partir del conjunto generado  $TCD_{cd}$ . Si dos periodos  $t_1, t_2 \in TCD_{cd}$  son consecutivos, entonces  $t_1$  y  $t_2$  pertenecen al mismo intervalo.

**PROGRAMACIÓN LINEAL ENTERA**

Para definir el Modelo de Programación Lineal Entera, se definen las siguientes variables Binarias:

- $x_{crt} = 1$ , si la asignatura  $c \in C$  es programada en el aula  $r \in RC_c$ , en el periodo  $t \in TC_c$ ,  $x_{crt} = 0$ , e.o.c.
- $\mu_{cmh} = 1$ , si para la asignatura  $c \in C$  se asignan  $h$  periodos consecutivos en el intervalo  $m \in MCD_{cd}$ ,  $h \in HC_c$ ,  $d \in DC_c$ ,  $\mu_{cmh} = 0$ , e.o.c.

Con la definición de estas variables, la formulación es la siguiente:

$$\min \sum_{c \in C} \sum_{t \in TC_c} p_{ct} \sum_{r \in RC_c} x_{crt} \tag{1}$$

$$\text{s.a.} \sum_{r \in RC_c} \sum_{t \in TC_c} x_{crt} = n_c, \quad c \in C \tag{2}$$

$$\sum_{c \in CGT_{gt}} \sum_{r \in RC_c} x_{crt} \leq 1, \quad g \in G, t \in T \tag{3}$$

$$\sum_{c \in CST_{st}} \sum_{r \in RC_c} x_{crt} \leq 1, \quad s \in S, t \in TS_s \tag{4}$$

$$\sum_{c \in CRT_{rt}} x_{crt} \leq 1, \quad r \in R, t \in T \tag{5}$$

$$\sum_{r \in RC_c} \sum_{t \in TMCD_{mcd}} x_{crt} - \sum_{\substack{h \in HC_c \\ h \leq |TMCD_{mcd}|}} (\mu_{cmh} * h) = 0, \tag{6}$$

$c \in C, d \in DC_c, m \in MCD_{cd}$

El modelo está orientado a minimizar la suma de las penalidades.

Los conjuntos de restricciones (2), (3), (4) y (5) imponen que cada asignatura debe ser asignada al número de periodos requeridos, cada grupo debe tener asignado a lo

<sup>7</sup> El Parámetro  $Lmax_{gd}$  es determinado realizando consideraciones ergonómicas y de acuerdo a la *Calidad* deseada de la Programación.

más una asignatura, cada profesor puede dictar a lo más una asignatura en cada periodo y en cada sala se programa a lo más una asignatura por periodo, respectivamente.

El conjunto de restricciones (6) impone para cada asignatura  $c \in C$  que es asignada en un día  $d \in DC_c$  en el intervalo  $m \in MCD_{cd}$  se deben asignar  $h \in HC_c$  periodos consecutivos.

Los siguientes conjuntos de restricciones aseguran que si más de un periodo es programado el mismo día, estos sean en periodos consecutivos (7) y en la misma aula (8).

$$\sum_{r \in R_c} (x_{cr_1t_1} - x_{cr_1t_2} + x_{cr_1t_3}) \leq 1, \quad (7)$$

$$c \in C, \exists h \in HC_c > 1, d \in DC_c,$$

$$m \in MCD_{cd}, |TMCD_{mcd}| \geq 3,$$

$$t_1, t_2, t_3 \in TMCD_{mcd}, t_1 < t_2 < t_3,$$

$$|TMCD_{mcd}| > h$$

$$x_{cr_1t_1} + x_{cr_2t_2} \leq 1, \quad c \in C, |RC_c| > 1, \quad (8)$$

$$r_1, r_2 \in RC_c, r_1 \neq r_2, d \in DC_c,$$

$$m \in MCD_{cd}, t_1, t_2 \in TMCD_{mcd},$$

$$t_1 < t_2, t_2 - t_1 < h, h = \max(h),$$

$$|TMCD_{mcd}| \geq h, h > 1, h \in HC_c$$

$$\sum_{c \in CGD_{gd}} \sum_{r \in RC_c} \sum_{t \in TC_{cd}} x_{crt} \leq l \max_{gd}, \quad (9)$$

$$g \in G, d \in D$$

$$\sum_{m \in MCD_{cd}} \sum_{\substack{h \in HC_c \\ h \leq |TMCD_{mcd}|}} \mu_{cmh} \leq 1, \quad (10)$$

$$c \in C, d \in DC_c$$

$$\sum_{d \in DC_c} \sum_{\substack{m \in MCD_{cd} \\ h \leq |TMCD_{mcd}|}} \mu_{cmh} = 1, \quad c \in C, h \in HC_c \quad (11)$$

$$x_{crt} = 1, \quad \forall (c, r, t) \in AP$$

$$x_{crt} \in \{0, 1\}, \quad c \in C, r \in RC_c, t \in TC_c \quad (12)$$

$$\mu_{cmh} \in \{0, 1\},$$

$$c \in C, d \in DC_c, m \in MCD_{cd}, h \in HC_c$$

El conjunto de restricciones (9) impone un número máximo de periodos diarios que se pueden asignar para cada grupo.

El conjunto de restricciones (10) asegura que para cada asignatura en un mismo día, el número máximo de multiperiodos a asignar es 1.

El conjunto de restricciones (11) impone que para cada asignatura, cada multiperiodo debe ser asignado una sola vez.

El conjunto de restricciones (12) permite tener asignaturas previamente asignadas a un periodo y aula, determinada.

## MODIFICACIONES AL MODELO Y MÉTODOS DE RESOLUCIÓN

Al Modelo formulado anteriormente, es posible realizarle modificaciones con el objetivo de obtener una solución en un menor *Tiempo Computacional*. A continuación se mencionan modificaciones al Modelo y a los Métodos de resolución:

### Asignación a Tipos de Salas

- De manera similar a la formulación propuesta en [22], la siguiente modificación al modelo consiste en definir conjuntos de *Tipos de Salas*  $R'$ . Un *Tipo de Sala*  $r' \in R'$  tiene asociado un conjunto de aulas  $RR'_{r'} = \{r_1, r_2, \dots, r_{N_{r'}}\}$  con características similares y generalmente de igual capacidad. Todos los conjuntos, parámetros, variables del modelo definido anteriormente que estén relacionado con cada aula  $r \in R$  deben reemplazarse por  $r' \in R'$ . De igual manera se redefine la variable  $x_{crt} = 1$ , si la asignatura  $c \in C$  es programada en el aula  $r' \in R'_{r'}$ , en el periodo  $t \in TC_c$ ,  $x_{crt} = 0$ , e.o.c.

De esta manera, existe una menor combinatoriedad de las variables  $x_{crt}$ . La principal modificación a realizar al modelo es en el conjunto de restricciones (5) relacionada con las salas:

$$\sum_{c \in CR'_{T_r, t}} x_{crt} \leq N_{r'}, \quad r' \in R', t \in T \quad (13)$$

Con  $r' \in R'$ , el Tipo de Patrón de salas  $r'$  y  $N_{r'}$ , el número de salas de Tipo Patrón  $r'$ . Esta restricción quiere decir que para todas las asignaturas que se pueden dictar en el periodo  $t \in T$  y en alguna sala perteneciente a la Sala *Patrón* tipo

$\mathbf{r}' \in \mathbf{R}'$ , se pueden realizar simultáneamente, a lo más,  $N_r$  asignaciones. Posterior a resolver el problema, se generó un algoritmo que asigna cada variable  $x_{cr't}$  igual a 1 ( $x_{cr't} = 1$ ), a una sala específica  $\mathbf{r} \in \mathbf{RR}'_r$ , respetando que para cada asignatura programada a más de un multiperiodo consecutivo se deba realizar en la misma sala.

**Relajación de Restricciones**

Esta modificación está basada en el método de relajación de conjuntos de restricciones propuesto en [17].

Los conjuntos (7) y (8) involucran una gran cantidad de restricciones y hacen que el problema sea difícil de resolver e implica un alto *Tiempo Computacional* para obtener una solución.

Al relajar estos dos conjuntos de restricciones, la solución obtenida consiste en asignaciones de cursos en distintos periodos en un mismo intervalo, en un mismo día y en distintas aulas. Al realizar esta relajación se espera que la solución obtenida se obtenga en un menor *Tiempo Computacional*.

Posterior a la obtención de un resultado del problema relajado se puede resolver un problema para cada día. Del problema relajado, se obtienen los valores de  $\mu_{cmh}$  (iguales a 0 o 1). Estos valores en una segunda etapa se consideran para fijar, para cada curso  $c \in C$  y cada día  $d \in DC_c$ , los multiperiodos  $h \in HC_c$  que han sido asignados, independiente del periodo  $m \in MCD_{cd}$ , que se asignó previamente<sup>8</sup>. Por lo tanto, usando la misma definición de variables, para cada día  $d \in D$ , se definen los siguientes conjuntos y variables:

$CD_d$ : Conjunto de las asignaturas  $c \in C$  que son asignadas al día  $d$

$GD_d$ : Conjunto de los grupos  $g \in G$  que son asignados en el día  $d$

$SD_d$ : Conjunto de los profesores  $s \in S$  que son asignados el día  $d \in D$

$TSD_{sd}$ : Conjunto de los periodos en que el profesor  $s \in S$  puede dictar asignaturas para el día  $d \in D$

<sup>8</sup> En la siguiente etapa se pueden asignar los  $h$  periodos consecutivos en un intervalo  $m \in MCD_{cd}$  distinto al asignado en la primera etapa, pero dentro del mismo día  $d$ , siempre que los periodos consecutivos estén contenidos dentro del otro intervalo.

$AP_d = (c_d, r, t_d) \in C_d \times R_c \times T_{cd} / (c, r, t)$ : Es una asignación previa cuando la asignatura  $c \in CD_d$  se realiza en el aula  $r \in RC_c$  en el periodo  $t \in TCD_{cd}$ .

Con la definición de estos nuevos conjuntos, más los conjuntos definidos en el modelo inicial se tienen las siguientes variables:

$y_{cr't} = 1$ , si la asignatura  $c \in CD_d$  es programada en el aula  $r \in RC_c$ , en el periodo  $t \in TCD_{cd}$ ,  $x_{cr't} = 0$ , e.o.c.

$z_{cmh} = 1$ , si para la asignatura  $c \in CD_d$  se asignan  $h$  periodos consecutivos en el intervalo  $m \in MCD_{cd}$ ,  $h \in HC_c$ ,  $z_{cmh} = 0$ , e.o.c.

Los modelos respectivos para cada día  $d \in D$  son los siguientes:

**MÉTODOS DE RESOLUCIÓN**

Considerando el Modelo Básico, el Modelo con Tipos de Aulas e incluyendo o no estrategia de relajación, es posible generar 4 métodos distintos de resolución:

1. **Timetabling (TT)**: Modelo inicial
2. **Timetabling con Tipos de Aulas (TTA)**: El Modelo con Tipos de Aulas
3. **Timetabling con Estrategia de Relajación (TTR)**: Modelo con Relajación de Restricciones
4. **Timetabling con Tipos de Aulas y Estrategia de Relajación (TTAR)**: El Modelo con Tipos de Aulas y Relajación previa de Restricciones.

**IMPLEMENTACIÓN**

Cada uno de estos métodos fue programado en **Microsoft Visual C++ 6.0** y usando las librerías de resolución para problemas de **Programación Lineal Entera ILOG CPLEX 9.0** [26].

$$MIN \sum_{c \in C_d} \sum_{t \in T_{cd}} p_{ct} \sum_{r \in R_c} y_{cr't} \tag{14}$$

$$s.a. \sum_{r \in RC_c} \sum_{t \in TCD_{cd}} y_{cr't} = \sum_{m \in MCD_{cd}} \sum_{h \in HC_c} \mu_{cmh} h, \quad c \in CD_d \tag{15}$$

$$\sum_{c \in CGT_{gt}} \sum_{r \in RC_c} y_{cr} \leq 1, \quad g \in GD_d, \quad (16)$$

$$t \in TD_d$$

$$\sum_{c \in CST_{st}} \sum_{r \in RC_c} y_{cr} \leq 1, \quad s \in SD_d, \quad (17)$$

$$t \in TSD_{sd}$$

$$\sum_{c \in CRT_{rt}} y_{cr} \leq 1, \quad r \in R, t \in TD_d, \quad (18)$$

$$\sum_{r \in RC_c} \sum_{t \in TMCD_{mcd}} y_{crt} - \sum_{h \in HC_c, h \leq |TMCD_{mcd}|} (z_{cmh} \cdot h) = 0, \quad c \in CD_d, \quad (19)$$

$$\sum_{m \in MCD_{cd}} m_{cmh} \neq 0$$

$$m \in \hat{MCD}_{cd}$$

$$\sum_{r \in RC_c} (y_{cr_1} - y_{cr_2} + y_{cr_3}) \leq 1, \quad (20)$$

$$c \in CD_d \mid \exists h > 1 \wedge \sum_{m \in MCD_{cd}} (\mu_{cmh}) \neq 0,$$

$$m \in MCD_{cd}, \quad |TMCD_{mcd}| \geq 3,$$

$$t_1, t_2, t_3 \in TMCD_{mcd}, \quad t_1 < t_2 < t_3,$$

$$|TMCD_{mcd}| > h$$

$$y_{cr_1} + y_{cr_2} \leq 1, \quad c \in CD_d,$$

$$|RC_c| > 1, r_1, r_2 \in RC_c, r_1 \neq r_2,$$

$$m \in MCD_{cd}, t_1, t_2 \in TMCD_{mcd}, \quad (21)$$

$$t_1 < t_2, t_2 - t_1 < h, h > 1,$$

$$m \in \sum_{MCD_{cd}} (\mu_{cmh}) \neq 0$$

$$y_{cr} = 1, \quad \forall (c_d, r, t_d) \in APD_d$$

$$x_{cr} \in \{0, 1\}, \quad c \in CD_d, r \in RC_c,$$

$$t \in TCD_{cd} \quad (22)$$

$$z_{cmh} \in \{0, 1\}, \quad c \in CD_d,$$

$$m \in MCD_{cd}, h \in HC_c$$

### Penalizaciones $P_{ct}$ relacionadas con la Función Objetivo

En la implementación de los modelos y en relación a las características del problema de la *Facultad de Ingeniería de la Universidad de Concepción*, se decidió dar una penalidad  $p_{ct}$  igual a 1 para todos los periodos en los cuales no es deseado dictar cualquier asignatura y una penalidad igual a 0 para todos los periodos en los cuales es indiferente dictar cualquier asignatura. Los periodos penalizados (iguales a 1) fueron aquellos que corresponden a las horas de almuerzo, desde las 13:00 hasta las 15:00 horas y las últimas horas de cada día, desde las 18:00 hasta las 21:00 horas. Cada periodo considerado tiene una duración de 1 hora, por lo tanto para 5 días de programación (de lunes a viernes) existen 25 periodos penalizados. De esta manera, el valor de la *Función Objetivo* (F.O.) se interpreta de manera directa y corresponde al número de asignaciones que se están realizando en periodos no deseados (en horas de almuerzo o en los últimos periodos del día).

### Criterios de Término

El método de *Branch and Bound* de *ILOG CPLEX 9.0* finaliza por defecto cuando el valor del **GAP relativo** es menor o igual al 0,01%. Este valor se puede modificar fijando distintos criterios de término como un **GAP absoluto** (que el valor absoluto entre la diferencia de la cota superior e inferior sea menor que un cierto valor) o un término en relación al *Tiempo Computacional* transcurrido, entre otros.

Se observa mediante pruebas que los *Modelos y Métodos* encuentran buenas soluciones pero a medida que el tamaño de cada problema aumenta, se ocupa un alto *Tiempo Computacional* en encontrar una solución con un **GAP relativo** pequeño. Por lo tanto, se concluye que se debe generar un criterio de término en relación al tamaño de cada problema, obteniendo una solución de *Calidad y Tiempo Computacional* razonable. Como las penalidades por dictar una asignatura  $c \in C$  en un periodo  $t \in CT_t$  pueden ser 0 o 1, mientras más asignaturas y periodos existan el problema es de mayor tamaño y, por lo tanto, generalmente se debe ocupar un mayor *Tiempo Computacional* en encontrar una solución satisfactoria. Un buen indicador (**IND**) que relacione la complejidad del problema y su tamaño es la suma de todas las penalidades para todas las asignaturas  $c$  y periodos  $t$ :

$$IND = \sum_{c \in C} \sum_{t \in TC_c} p_{ct}$$

Mediante este indicador se define que el algoritmo finalice o acepte una solución como óptima cuando el **GAP absoluto** sea menor o igual al entero superior del 1% de este indicador:

$$\text{GAP}_{\text{absoluto}} = \lceil 0.01 * \text{IND} \rceil$$

También se fija como criterio de término un **GAP relativo** igual al 5%. Por lo tanto el algoritmo se termina cuando el **GAP absoluto** es menor o igual al techo del 1% del indicador **IND** o cuando el **GAP relativo** es menor o igual al 5%.

Estos criterios de término son válidos para los modelos y métodos de **Timetabling**, **Timetabling con Tipos de Aulas** y para las primeras etapas de los métodos de **Timetabling con Estrategia de Relajación** y **Timetabling con Tipos de Aulas y Estrategia de Relajación**. Para la segunda etapa de estos dos últimos métodos se fijan como criterios de término un **GAP absoluto** menor o igual a 2 y un **GAP relativo** del 5%.

## RESULTADOS

### Generación de Instancias

Para poder evaluar la calidad de cada uno de los modelos se crearon 10 instancias a partir de datos reales de la *Facultad de Ingeniería de la Universidad de Concepción* para el Primer Semestre del año 2006. En general, las 9 primeras instancias creadas son un subconjunto del problema completo que involucra un total de 177 asignaturas, 72 grupos, 72 profesores, 13 periodos por día, 5 días, 38 aulas, 113 profesores y 26 asignaciones previas. Este problema incluye todas las asignaturas de las carreras de la *Facultad de Ingeniería* que se deben programar. Cada problema contiene 5 días, de lunes a viernes; las instancias 7, 8 y 10 tienen 65 periodos (T) disponibles; el resto de las instancias tiene 60 periodos. Igualmente, las instancias 1 a 6, tienen 3 asignaciones previas, la instancia 7 tiene 21 y las instancias 8 a 10 tienen 26 asignaciones previas. En la tabla 1 se observan las principales características de cada uno de los problemas implementados, el número de asignaturas o cursos (**|C|**), número de periodos totales a programar ( $\sum_{c \in C} n_c$ ), número de grupos (**|G|**), número de aulas (**|R|**), número de Tipos de Aulas (**|R'|**) y el número de profesores (**|S|**):

Tabla 1. Principales Características de los Problemas Implementados.

Instancia	C	$\sum_{c \in C} n_c$	G	R	R'	S
1	17	72	4	21	14	17
2	18	77	5	21	14	17
3	22	94	6	21	14	19
4	26	109	7	21	14	22
5	31	134	8	21	14	26
6	35	154	9	21	14	28
7	68	290	19	24	17	46
8	130	543	61	30	23	80
9	177	700	72	38	31	113
10	177	700	72	38	31	113

### Resultados Obtenidos

Cada instancia fue probada mediante los 4 Métodos de resolución explicados anteriormente. Los valores específicos del indicador **IND** ( $\text{IND} = \sum_{c \in C} \sum_{t \in T_c} p_{ct}$ ) y el **GAP absoluto** para cada instancia, son iguales para cada Método<sup>9</sup> y se presentan en la tabla 2.

Tabla 2. Indicador **IND** y **GAP absoluto** para cada Instancia.

Instancia	IND	GAP <sub>absoluto</sub>
1	252	3
2	272	3
3	358	4
4	433	5
5	533	6
6	605	7
7	1.128	12
8	2.284	23
9	3.147	32
10	3.166	32

Es importante mencionar que todos los problemas resueltos obtienen soluciones que respetan el máximo número de asignaciones por día para cada Grupo de alumnos  $L_{\text{max}_{\text{gd}}}$

<sup>9</sup> Los valores de **GAP absoluto** presentados en la tabla 2 para las estrategias de resolución que incluyen relajación previa de restricciones son los valores de la etapa de relajación. Los valores del **GAP absoluto** de la segunda etapa son iguales a 2 para estas dos estrategias.

igual a 8, cumpliéndose esta *Medida de Calidad*. Los Métodos de resolución y los problemas fueron probados en un computador *Athlon 64 Doble Núcleo* con un *Procesador* de 2200 MHz y 2 MB en RAM.

#### Resultados Obtenidos para el Enfoque de Timetabling

En la tabla 3 se presentan los resultados de los problemas al resolverlos con el método de *Timetabling*. En la primera columna se tiene cada instancia (N°) al que fue aplicado el modelo, en la segunda y tercera respectivamente, se tiene el número de filas (restricciones, N° Fil) y columnas (variables, N° Col) que tiene cada modelo. En la cuarta columna de la tabla, se presentan el número de no ceros (N° No Ceros) que tiene la matriz de restricciones. En la quinta y sexta columna respectivamente, se presenta la *Función Objetivo* (F.O.) y el porcentaje de periodos asignados en periodos penalizados en relación al número total de asignaciones de cada problema. Por último, en la séptima columna se obtiene el *Tiempo Computacional* en segundos (T): De los resultados de la tabla 3 se puede observar que la *Calidad* de la *Función Objetivo* o el número de periodos

asignados en momentos no deseados en relación al número total de periodos a programar es a lo más de un **4,55%**. En relación al *Tiempo Computacional* se puede observar que la instancia que se demora menos en resolverse es el número **4**, obteniendo la solución en **4,47** segundos. La instancia que se demoró más en resolverse fue la número **6**, obteniéndose el resultado en **2.081,16** segundos (**≈34 minutos**). Cabe señalar que las instancias **7, 8, 9** y **10** se dejaron ejecutando más de **cuatro días**, sin obtener una solución factible.

#### Resultados Obtenidos para el Enfoque de Timetabling con Tipos de Aulas:

En la tabla 4 se presentan los resultados de los problemas al resolverlos con el método de *Timetabling con Tipos de Aulas*. La tabla contiene los mismos ítems que la tabla anterior.

A partir de la tabla 4 se observa que cada instancia contiene una menor cantidad de variables (columnas) y restricciones (filas) que el método de *Timetabling*,

Tabla 3, Resultados Obtenidos para el Método de Timetabling,

N°	N° Filas	N° Columnas	N° No Ceros	F. O.	$\left(\frac{F.O.}{\sum_{c \in C} n_c}\right) * 100$	T
1	57.649	6.319	284.072	1	1,39	17,53
2	70.973	7.221	352.952	3	3,90	19,86
3	98.811	9.499	505.865	4	4,26	4,47
4	117.327	11.122	572.528	4	3,67	174,16
5	174.307	15.322	889.628	5	3,73	505,22
6	212.813	18.398	1.077.116	7	4,55	2.081,16
7-8-9-10	-----	-----	-----	-----	-----	-----

Tabla 4, Resultados Obtenidos para el Método de Timetabling con Tipos de Aulas,

N°	N° Filas	N° Columnas	N° No Ceros	F. O.	$\left(\frac{F.O.}{\sum_{c \in C} n_c}\right) * 100$	T
1	22.977	3.807	140.267	2	2,78	7,89
2	28.849	4.397	178.507	3	3,90	12,83
3	42.667	5.951	270.055	4	4,26	2,72
4	50.503	6.983	303.644	5	4,59	32,67
5	77.483	9.808	490.119	5	3,73	73,89
6	95.349	11.884	597.987	6	3,90	135,69
7	157.936	19.973	1.016.512	29	10,00	93,17
8-9-10	—	—	—	—	—	—

pareciendo razonable obtener las soluciones en un *Tiempo Computacional* menor.

Aunque se obtuvieron resultados para la instancia número 7 (a diferencia de método de **Timetabling**), para las instancias 8, 9 y 10 no se obtuvo resultado factible en menos de cuatro días de *Tiempo Computacional*. En relación a los valores de la *Función Objetivo*, se puede concluir que la *Calidad de las Soluciones* en relación al número de periodos asignados en periodos no deseados es similar en instancias de menor tamaño (de a lo más del tamaño de la instancia número 6), obteniendo un porcentaje entre el 2,78% y 4,59% de periodos penalizados sobre el total en estas instancias. Para la instancia número 7 el valor del indicador aumenta a un 10%.

**Resultados Obtenidos para el Método de Timetabling con Estrategia de Relajación**

Como se explicó anteriormente, el método de resolución del enfoque de **Timetabling con Estrategia de Relajación** resuelve primero un problema relajado y posteriormente se resuelve un problema para cada día, obteniendo resultados para cada una de las instancias resueltas por este enfoque y un resultado para la solución global de la instancia. En la tabla 5 se obtienen los resultados para la etapa de relajación. En la primera columna se tienen las instancias que se resolvieron (N°), en la segunda y tercera columna, respectivamente, se presentan el número de filas (restricciones, N° Fil) y el número de columnas (variables, N° Col) que tiene cada instancia. En la cuarta columna de la tabla se presentan el número de no ceros (N° No Ceros) que tiene la matriz de restricciones. En la quinta y sexta columna, respectivamente, se presenta la *Función Objetivo* obtenida (F.O.) y el *Tiempo computacional* en segundos (T) para cada instancia.

De la tabla 5 se observa que cada instancia a resolver tiene un número menor de variables (columnas) y restricciones (variables) que los enfoques propuestos anteriormente, por lo que parece razonable que el *Tiempo Computacional* necesario para resolver cada problema sea también significativamente menor. En la segunda etapa se genera un problema para cada día, existiendo en total 5 problemas por cada instancia. Cada uno de estos problemas a resolver genera un menor número de variables y restricciones. Por ejemplo, para la instancia 10 cada día tiene entre 83.783 y 105.524 restricciones y entre 17.093 y 18.110 variables ocupando entre 132,12 y 580,64 segundos en resolver cada problema. Se observa que cada problema tiene en general un mayor número de restricciones y un menor número de variables, en comparación con la etapa de relajación del enfoque de Timetabling.

Tabla 5. Resultados Obtenidos para la Etapa de Relajación del Enfoque de Timetabling con Estrategia de Relajación.

Etapa de Relajación					
N°	N° Filas	N° Columnas	N° No Ceros	F.O.	T
1	2.587	6.319	39.541	0	0,25
2	2.719	7.221	48.717	1	0,31
3	2.970	9.499	63.131	0	0,44
4	3.233	11.122	73.289	4	0,49
5	3.588	15.322	99.764	2	0,61
6	3.842	18.398	119.072	0	3,64
7	6.214	31.333	203.486	20	2,39
8	12.154	60.388	506.197	52	176,69
9	15.112	89.951	705.180	64	720,63
10	15.701	90.170	706.758	65	554,69

En la tabla 6 se presenta el resultado global obtenido para cada problema mediante el enfoque de **Timetabling con Estrategia de Relajación**. En la primera columna se tiene cada instancia que fue resuelta (N°), en la segunda columna, se presenta la *Función Objetivo* (F.O.) o el número de periodos asignados en periodos no deseados, en la tercera columna, el porcentaje de periodos asignados en periodos penalizados en relación al número total de asignaciones de cada problema y en la cuarta columna se presenta el *Tiempo Computacional* en segundos ocupado para resolver cada instancia.

Tabla 6. Resultado Global obtenido mediante el Enfoque de Timetabling con Estrategia de Relajación.

N°	F.O.	$\left( \frac{F.O.}{\sum_{c \in C} n_c} \right) * 100$	T
1	1	1,39	1,36
2	1	1,30	1,56
3	2	2,13	2,31
4	5	4,59	2,33
5	8	5,97	3,84
6	6	3,90	7,73
7	27	9,31	10,86
8	60	11,05	418,89
9	69	9,86	2346,41
10	73	10,43	1839,41

A partir de la tabla 6 se observa que el porcentaje de periodos asignados en periodos no deseados es menor que un 5,97% para instancias de un tamaño menor o igual que la número 6, pero para instancias de mayor tamaño el porcentaje aumenta llegando a obtenerse un porcentaje del 11,05% para la instancia número 8. En relación al *Tiempo Computacional*, se observa que los resultados se obtienen en *Tiempos Computacionales* en general significativamente menores que en los otros dos enfoques expuestos con anterioridad, siendo más significativo para instancias de mayor tamaño. La instancia que se demoró más fue la número 9, obteniéndose el resultado en aproximadamente 39 minutos.

### Resultados Obtenidos para el Método de Timetabling con Tipos de Aulas y Estrategia de Relajación

En la tabla 7 se presentan los resultados de las instancias al resolverlas con el método de **Timetabling con Tipos de Aulas y Estrategia de Relajación**. Esta tabla contiene los mismos ítems que los expuestos en el enfoque anterior (tabla 5).

De la tabla 7 y comparando los resultados con los anteriores se puede observar que cada instancia tiene una menor cantidad de restricciones (filas) y variables (columnas) en comparación con la primera etapa del modelo que incluye *Relajación de Restricciones*.

Tabla 7. Resultados Obtenidos para la Etapa de Relajación del Enfoque de Timetabling con Tipos de Aulas y Estrategia de Relajación

Etapa de Relajación					
Nº	Nº Filas	Nº Columnas	Nº No Ceros	F.O.	T
1	2.167	3.807	23.436	0	0,22
2	2.299	4.397	29.116	0	0,20
3	2.550	5.951	38.900	1	0,27
4	2.813	6.983	45.267	0	0,39
5	3.168	9.808	62.992	1	0,45
6	3.422	11.884	75.940	0	1,52
7	5.759	19.973	127.992	22	1,73
8	11.699	39.534	336.528	49	331,09
9	14.692	59.311	469.002	63	964,63
10	15.246	59.458	470.027	58	1346,56

Para instancias pequeñas se observa que el *Tiempo Computacional* utilizado en resolver cada instancia es similar al *Tiempo Computacional* utilizado al resolver

la instancia mediante el método anterior para instancias pequeñas. Para instancias de mayor tamaño (de la instancia 8 en adelante), esta etapa de este enfoque entrega los resultados en *Tiempos Computacionales* más altos que en comparación con el enfoque anterior.

En la tabla 8 se presenta el resultado global obtenido para cada instancia mediante el método de **Timetabling con Tipos de Aulas y Estrategia de Relajación**. La tabla contiene los mismos ítems e instancias que el método anterior (tabla 6).

Tabla 8. Resultado Global Obtenido Mediante el Enfoque de Timetabling con Tipos de Aulas y Estrategia de Relajación.

Nº	F.O.	$\left(\frac{F.O.}{\sum_{c \in C} n_c}\right) * 100$	T
1	3	4,17	0,83
2	1	1,30	0,91
3	4	4,26	1,34
4	4	3,67	1,55
5	5	3,73	2,16
6	3	1,95	3,78
7	30	10,34	10,59
8	58	10,68	411,00
9	71	10,14	2241,91
10	67	9,57	2389,92

### Comparación de los Métodos de Resolución

La comparación de cada uno de los *Métodos de Resolución* puede realizarse en relación a la calidad de la *Función Objetivo* y en relación al *Tiempo Computacional* utilizado en resolver cada instancia. A continuación se evalúan los enfoques en relación a las mediadas de calidad mencionadas anteriormente:

#### Función Objetivo

En la tabla 9 se comparan los resultados obtenidos por cada Método y para cada instancia, en relación a la *Calidad de la Función Objetivo*. Cada Método **Timetabling (TT)**, **Timetabling con tipos de Aulas (TTA)**, **Timetabling con Relajación de Restricciones (TTR)** y **Timetabling con tipos de Aulas y Relajación de Restricciones (TTAR)** y cada instancia, contiene la *Función Objetivo* en la primera columna y la *Calidad* ella, en la segunda, obteniendo el porcentaje de periodos asignados en periodos no deseados (F.O.) en relación al total de periodos asignados en cada problema  $\left(\sum_{c \in C} n_c\right)$ . A partir de la tabla 9 se puede observar

que no existe un enfoque que obtenga la mejor *Función Objetivo* en todas las instancias. Las instancias más pequeñas tienen una *Calidad de Solución* mejor debido a que éstas tienen un menor número de asignaturas, aulas y en algunos casos periodos, existiendo soluciones de mejor calidad, que en instancias más grandes, no existen. Cabe señalar que hasta la instancia 6 cada problema tiene como cota inferior o solución de relajación de las restricciones de integralidad de las variables, igual a 0, y la mejor solución obtenida es para el enfoque de **Timetabling con Relajación de Restricciones**, para la instancia 10.

Para esta instancia y método, la solución es igual a 73, por lo tanto, la diferencia entre la solución obtenida y la posible mejor solución factible (57) es 16, que significa una disminución máxima de la calidad de un 8,14% a un 10,43%, es decir, de una diferencia de un 2,29%, lo que sigue siendo una solución de calidad de la *Función Objetivo* razonable.

**Tiempo Computacional**

En la tabla 10 se presentan los *Tiempos Computacionales* (en segundos) obtenidos para cada una de las 10 instancias propuestas y para cada uno de los *Métodos de Solución* propuestos, **Timetabling (TT)**, **Timetabling con tipos de Aulas (TTA)**, **Timetabling con Relajación de Restricciones (TTR)** y **Timetabling con Tipos de Aulas y relajación de Restricciones (TTAR)**.

A partir de la tabla 10 se puede observar que los Métodos que no incluyen *Relajación de Restricciones (TT y TTA)* ocupan un *Tiempo Computacional* más alto que los Métodos que sí incluyen *Relajación de Restricciones (TTR y TTAR)*, observándose una mayor diferencia a medida que el tamaño de las instancias aumenta. También se observa

que el Método de **Timetabling con Tipos de Aulas** ocupa un *Tiempo Computacional* menor en resolver cada instancia en comparación con el Método de **Timetabling**.

Para instancias de gran tamaño (a partir de la instancia 7 para el Método de **Timetabling** y a partir de la instancia 8 para el Método de **Timetabling con Tipos de Aulas**), los Métodos que no incluyen *Relajación de Restricciones* no encontraron una solución factible en menos de 4 días de *Tiempo Computacional*. En relación a los Métodos que sí incluyen *Relajación de Restricciones* se observa que el Método de **Timetabling con Tipos de Aulas y Relajación de Restricciones (TTAR)** ocupa un menor *Tiempo Computacional*, salvo en la instancia 10, en comparación con el Método de **Timetabling con Relajación de Restricciones**. Las instancias de mayor tamaño (instancias 9 y 10) se resolvieron en menos de aproximadamente 39 minutos.

Tabla 10. Comparación de los Enfoques utilizados en relación al *Tiempo de Resolución*, para cada instancia.

N°	Tiempo (s)			
	TT	TTA	TTR	TTAR
1	17,53	7,89	1,36	0,83
2	19,86	12,83	1,56	0,91
3	4,47	2,72	2,31	1,34
4	174,16	32,67	2,33	1,55
5	505,22	73,89	3,84	2,16
6	2.081,16	135,69	7,73	3,78
7	-	93,17	10,86	10,59
8	-	-	418,89	411,00
9	-	-	2.346,41	2.241,91
10	-	-	1.839,41	2.389,92

Tabla 9. Comparación de los Enfoques utilizados en relación a la *Función Objetivo* y su *Nivel de Calidad*.

N°	TT		TTA		TTR		TTAR	
	F.O.	$\left(\frac{F.O.}{\sum_{c \in C} n_c}\right) * 100$	F.O.	$\left(\frac{F.O.}{\sum_{c \in C} n_c}\right) * 100$	F.O.	$\left(\frac{F.O.}{\sum_{c \in C} n_c}\right) * 100$	F.O.	$\left(\frac{F.O.}{\sum_{c \in C} n_c}\right) * 100$
1	1	1,39	2	2,78	1	1,39	3	4,17
2	3	3,90	3	3,90	1	1,30	1	1,30
3	4	4,26	4	4,26	2	2,13	4	4,26
4	4	3,67	5	4,59	5	4,59	4	3,67
5	5	3,73	5	3,73	8	5,97	5	3,73
6	7	4,55	6	3,90	6	3,90	3	1,95
7	-	-	29	10,00	27	9,31	30	10,34
8	-	-	-	-	60	11,05	58	10,68
9	-	-	-	-	69	9,86	71	10,14
10	-	-	-	-	73	10,43	67	9,57

## CONCLUSIONES Y RECOMENDACIONES

En esta investigación se ha caracterizado, modelado y resuelto un problema de *Programación de Horarios en Universidades* a través de *Programación Lineal Entera*, obteniéndose Modelos y Métodos que permiten resolver problemas de gran tamaño en *Tiempos Computacionales* razonables y satisfaciendo *Niveles de Calidad* deseados.

Se propusieron 4 Métodos de resolución basados en *Programación Lineal Entera* para resolver los problemas. Dos Métodos pueden obtener soluciones óptimas para cualquier instancia en que exista solución. El primer Método asigna directamente los cursos a periodos y aulas (**Timetabling**); el segundo asigna los cursos a periodos y *Tipos de Aulas* que posteriormente, mediante un algoritmo, realiza las asignaciones de *Tipos de Aulas* a *Aulas* específicas que pertenecen a cada Tipo (**Timetabling con Tipos de Aulas**). Estos dos Métodos nombrados permiten resolver problemas satisfaciendo un *Nivel de Calidad* deseado pero en un alto *Tiempo Computacional*, existiendo problemas de gran tamaño, en que no fue posible obtener resultados en un *Tiempo Computacional* razonable. Por esta razón, se propusieron dos Métodos que no garantizan obtener soluciones óptimas e incluso encontrar solución, para instancias en que sí puede existir solución. Estos Métodos se basan en relajar en una primera etapa restricciones que aumentan la complejidad del problema, obteniendo soluciones que permiten fijar variables y resolver en una segunda etapa problemas más pequeños para cada día. De igual manera, un Método incluye asignar directamente a cada aula (**Timetabling con Estrategia de Relajación**) y otro, asigna inicialmente a *Tipos de Aulas* (**Timetabling con Tipos de Aulas y Estrategia de Relajación**). Para cada problema propuesto se encontró una solución en un *Tiempo Computacional* razonable y satisfaciendo un *Nivel de Calidad* deseado, para los dos últimos Métodos nombrados anteriormente.

Comparando los Métodos que incluyen una estrategia de **Relajación de Restricciones**, no se puede asegurar que un Método es mejor que otro en cuanto al *Nivel de Calidad* y *Tiempo Computacional*, por lo que se recomienda evaluar cada Método con problemas de mayor tamaño.

Los Modelos generados permiten obtener horarios para cada grupo de alumnos, cada profesor y cada aula. Aunque se modeló y resolvió el problema específicamente para la *Facultad de Ingeniería de la Universidad de Concepción*, los Modelos permiten ajustarse a una gran

cantidad de problemas de *Programación de Horarios en Universidades*, proporcionando una gran flexibilidad de resolución. La información obtenida permite realizar análisis de sensibilidad evaluando cada problema real agregando o quitando recursos (profesores, aulas o periodos).

Los Métodos de solución propuestos fueron implementados fijando variables, usando técnicas de ramificación en algunos enfoques, que producen una disminución en el *Tiempo Computacional* y fijando *Criterios de Término* ( $GAP_{\text{absoluto}}$  y  $GAP_{\text{relativo}}$ ) que también disminuyen el *Tiempo Computacional* pero no garantizan la optimalidad de todos los problemas. Si se dispone de un mayor *Tiempo Computacional* para resolver un problema, se pueden restringir los *Criterios de Término*, con el objetivo de obtener un mayor *Nivel de Calidad* de solución. Por el contrario, si se desea obtener una solución en un bajo *Tiempo Computacional*, se pueden relajar los criterios de término obteniendo generalmente una solución de un menor *Nivel de Calidad* deseado.

## AGRADECIMIENTOS

El presente trabajo ha sido presentado a la Escuela de Graduados de la Universidad de Concepción, en el programa de Magíster en Ingeniería Industrial. Este trabajo es apoyado parcialmente por el proyecto ALFA N° II-0457-FA-FCD-FI-FC, proyecto DIUC 205.093.010-1.0 y por proyecto FUNDACIÓN ANDES, CHILE N° C-13955/18.

## REFERENCIAS

- [1] S. Abdennadher and M. Marte. "University Course Timetabling using Constraint Handling Rules". *Journal of Applied Artificial Intelligence*. Special Issue on Constraint Handling Rules (Holzbaur, C. y Frühwirth, T. Editores). Vol. 14 N° 4, pp. 311- 325. 2000.
- [2] A. Asratian and D. de Werra. "A Generalized Class-Teacher Model for Some Timetabling Problems". *European Journal of Operational Research*. Vol. 143 N° 3, pp. 531-542. 2002.
- [3] P. Avella and I. Vasil'ev. "A Computational Study of a Cutting Plane Algorithm for University Course Timetabling". *Journal of Scheduling*. Vol. 8 N° 6, pp. 497-514. 2005.

- [4] V. Bardadym. "Computer Aided School and University Timetabling. The New Wave". Lecture Notes in Computer Science Series, Vol. 1153, pp. 22-45. 1996.
- [5] P. Brucker and S. Knust. "Resource-Constrained Project Scheduling and Timetabling". Lecture Notes in Computer Science. Vol. 2079, pp. 277-293. 2001.
- [6] E. Burke, D. de Werra and J. Kingston. "Applications to Timetabling". Gross y Yellen Editores and Handbook of Graph Theory, pp. 445-474. 2003.
- [7] E. Burke, K. Jackson, J. Kingston and R. Weare. "Automated University Timetabling: The State of the Art". The Computer Journal. Vol. 40 N° 9, pp. 565-571, 1998.
- [8] M. Carrasco and M. Pato. "A Multiobjective Genetic Algorithm for the Class/Teacher Timetabling Problem". Lecture Notes in Computer Science. Vol. 2079, pp. 3-17. 2001.
- [9] M. Carter and G. Laporte. "Recent Developments in Practical Course Timetabling". Lecture Notes in Computer Science. Vol. 1408, pp. 3-19. 1998.
- [10] S. Daskalaki and T. Birbas. "Efficient Solutions for a University Timetabling Problem Through Integer Programming". European Journal of Operational Research. Vol. 160 N° 1, pp. 106-120. 2005.
- [11] S. Daskalaki, T. Birbas and E. Housos. "An integer Programming Formulation for a Case Study in University Timetabling". European Journal of Operational Research. Vol. 153, pp. 117-135. 2004.
- [12] D. de Werra, A. Arsatian and S. Durand. "Complexity of Some Special Types of Timetabling Problems". Journal of Scheduling. Vol. 5, pp. 171-183. 2002.
- [13] D. de Werra. "An Introduction to Timetabling". European Journal of Operational Research. Vol. 19, pp. 151-162. 1985.
- [14] S. Deris, S. Omatu and H. Ohta. "Timetable Planning Using the Constraint - Based Reasoning". Computers and Operations Research. Vol. 27 N° 9, pp. 819-840. 2000.
- [15] L. di Gaspero and A. Schaerf. "Multi-Neighbourhood Local Search with Application to Course Timetabling". Lecture Notes in Computer Science. Vol. 2740, pp. 263-278. 2003.
- [16] M. Dimopoulou and P. Miliotis. "An Automated University Course Timetabling System Developed in a Distributed Environment: A Case Study". European Journal of Operational Research. Vol. 153, pp. 136-147. 2004.
- [17] W. Legierski. "Search Strategy for Constraint - Based Class-Teacher Timetabling". Lectures Notes in Computer Science. Vol. 2740, pp. 247-261. 2003.
- [18] S. MirHassani. "A Computational Approach to Enhancing Course Timetabling with Integer Programming". Applied Mathematics and Computation. Vol. 175 N° 1, pp. 814-822. 2006.
- [19] A. Qualizza and P. Serafini. "A Column Generation Scheme for Faculty Timetabling". Lecture Notes in Computer Science. Vol. 3616, pp. 161-173. 2005.
- [20] L. Reis and E. Oliveira. "A Language for Specifying Complete Timetabling Problems". Lecture Notes in Computer Science. Vol. 2079, pp. 322-341. 2001.
- [21] H. Santos, L. Ochi and M. Souza. "A Tabu Search Heuristic with Efficient Diversification Strategies for the Class/Teacher Timetabling Problem". Lecture Notes in Computer Science. Vol. 3616, pp. 343-358. 2005.
- [22] K. Socha, M. Sampels and M. Manfrin. "Ant Algorithms for The University Course Timetabling Problem with Regard to The State - of - The-Art". In Applications of Evolutionary Computing: Proceedings of Evo Workshops 2003. Berlin, Germany. Lecture Notes in Computer Science. Vol. 2611, pp. 334-345. 2003.
- [23] A. Tripathy. "School Timetabling - A Case in Large Binary Integer Linear Programming". Management Science. Vol. 30 N° 12, pp. 1473-1489. 1984.
- [24] H. Ueda, D. Ouchi, D. Takahashi and T. Miyahara. "A Co-evolving Timeslot/Room Assignment

- Genetic Algorithm Technique for University Timetabling”. Lecture Notes in Computer Science. Vol. 2079, pp. 48-63. 2001.
- Object - Oriented Constraint - Based Model for University Course Timetabling”. Lecture Notes in Computer Science. Vol. 2079, pp. 28-47. 2001.
- [25] K. Zervoudakis and P. Stamatopoulos. “A Generic  
[26] ILOG. “ILOG CPLEX 9.0 Reference’s Manual”. 2003.