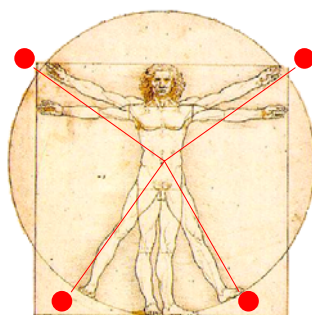


# **TECNOLOGÍ@ y *DESARROLLO***

*Revista de Ciencia, Tecnología y Medio Ambiente*

VOLUMEN V . AÑO 2007

SEPARATA



## **APLICACIÓN DE TÉCNICAS DE ENFRIAMIENTO LENTO AL PROBLEMA DE PLANIFICACIÓN DE HORARIOS**

Pilar Moreno Díaz, Jesús Sánchez Allende, Almudena García Manso



UNIVERSIDAD ALFONSO X EL SABIO  
Escuela Politécnica Superior

Villanueva de la Cañada (Madrid)



© Del texto: Pilar Moreno Díaz, Jesús Sánchez Allende, Almudena García Manso  
Septiembre, 2007

[http://www.uax.es/publicaciones/archivos/TECELS07\\_002.pdf](http://www.uax.es/publicaciones/archivos/TECELS07_002.pdf)

© De la edición: *Revista Tecnol@ y desarrollo*

Escuela Politécnica Superior.

Universidad Alfonso X el Sabio.

28691, Villanueva de la Cañada (Madrid).

ISSN: 1696-8085

Editor: Julio Merino García [tecnologia@uax.es](mailto:tecnologia@uax.es)

No está permitida la reproducción total o parcial de este artículo, ni su almacenamiento o transmisión ya sea electrónico, químico, mecánico, por fotocopia u otros métodos, sin permiso previo por escrito de la revista.

[http://www.uax.es/publicaciones/archivos/TECELS07\\_002.pdf](http://www.uax.es/publicaciones/archivos/TECELS07_002.pdf)



# APLICACIÓN DE LAS TÉCNICAS DE ENFRIAMIENTO LENTO AL PROBLEMA DE PLANIFICACIÓN DE HORARIOS

**Jesús Sánchez Allende<sup>a</sup>, Pilar Moreno Díaz<sup>b</sup>,  
Almudena García Manso<sup>c</sup>**

<sup>a</sup>Dr Telecomunicación, <sup>b</sup>Lda Matemáticas, <sup>c</sup>Dra Sociología

Departamento de Electrónica y Sistemas, Escuela Politécnica Superior, Universidad Alfonso X el Sabio.

Avda. De la Universidad nº1, Villanueva de la Cañada, 28691 Madrid. España. Tlf.:918109237, email: pilar@uax.es

**RESUMEN:** Uno de los problemas comunes en muchos centros docentes es la planificación de los horarios de los profesores para un conjunto grande de grupos o asignaturas que pueden resultar equivalentes. Este problema de planificación es un problema NP-completo cuya resolución no admite una exploración exhaustiva del espacio de soluciones, ni siquiera para problemas de dimensiones relativamente reducidas. En este artículo se presentan las bases de cómo utilizar de forma general la técnica de Simulated Annealing (enfriamiento lento simulado o recocido) y su aplicación real al citado problema.

**PALABRAS CLAVE:** Simulated annealing, problema de asignación de horarios, planificación de recursos.

## 1 Introducción

Existen múltiples aproximaciones para la resolución de problemas de optimización y planificación de recursos. Este tipo de problemas se caracteriza porque el espacio de soluciones es de tal dimensión que no permite analizar todas las posibilidades combinatorias del problema. Por ello se utilizan distintos tipos de técnicas que permitan obtener soluciones lo suficientemente buenas para su aplicación práctica.

Una de las técnicas prometedoras para resolver este tipo de problemas es el enfriamiento lento simulado (Simulated Annealing), también llamados algoritmos de recocido. En este trabajo se exponen los conceptos básicos que se aplican con este tipo de técnica y se dan las pistas para su aplicación en la resolución del problema de planificación de horarios, más específicamente a la planificación de los horarios de un conjunto de profesores teniendo en cuenta un conjunto ya existente de horarios de grupos de asignaturas, acompañándolo con un ejemplo real de uso.

[http://www.uax.es/publicaciones/archivos/TECELS07\\_002.pdf](http://www.uax.es/publicaciones/archivos/TECELS07_002.pdf)

A partir del esquema aquí planteado no debería resultar complicado utilizar el esquema de solución que se presenta para resolver un problema distinto al de planificación de los horarios, manteniendo toda la parte del programa referida al algoritmo de enfriamiento lento simulado. La diferencia estará en el tratamiento que se haga del modelo de sistema a planificar, de la forma de generar las variaciones del problema (generación de vecindad) y de la forma de calcular la función de coste asociada a cada una de las variantes de solución que se vaya obteniendo en la ejecución del algoritmo.

Aunque existen muchas variantes del algoritmo que se presenta en este artículo (adaptive simulated annealing, cooperative simulated annealing, parallel simulated annealing, etc) se ha decidido utilizar su formulación clásica como aproximación al lector de este tipo de técnicas, de forma que pueda conocer sus características y su modelo de uso.

## **2 Enfriamiento lento simulado**

La familia de algoritmos de enfriamiento lento simulado aparece a principios de los años 80 [KIRKPATRICK 1983]. Se trata de un modelo de resolución para la optimización de problemas de tipo combinatorio con mínimos locales. Su aproximación consiste en generar aleatoriamente una solución cercana a la solución actual (o en el entorno de la solución) y la acepta como buena si consigue reducir una determinada función de coste, o con una determinada probabilidad de aceptación.

Esta probabilidad de aceptación se irá reduciendo con el número de iteraciones y está relacionada también con el grado de empeoramiento del coste.

Este tipo de algoritmos se diferencia de los algoritmos de hill climbing o de escalada en que en estos últimos solo se acepta una nueva solución si ésta mejora la anterior de manera que cada solución que se obtenga vaya obteniendo soluciones mejores cada vez. El problema de este tipo de algoritmos es que la solución final suele encontrarse en un mínimo local no explorando con suficiente amplitud el espacio de soluciones. Sin embargo, en la aplicación del algoritmo de enfriamiento lento se pueden aceptar soluciones que empeoran la solución actual, solo que esta aceptación dependerá de una determinada probabilidad que depende de la temperatura a la que se encuentre el sistema.

Estos algoritmos se llaman de esta forma por su parecido en funcionamiento al proceso de enfriamiento metalúrgico: cuando se enfría un metal fundido suficientemente despacio, tiende a solidificarse en una estructura de mínima energía (equilibrio térmico); según va disminuyendo la temperatura, las moléculas del metal tienen menor probabilidad de moverse de su nivel energético. La probabilidad de movimiento de una molécula de su estado viene dada por la función de Boltzmann  $P(DE) = e^{-DE/kT}$ , que depende de la temperatura, de la diferencia de energía y de una constante  $K$  (la constante de Boltzmann).

Los algoritmos de enfriamiento lento simulado tienen algunas ventajas con respecto a otras técnicas de optimización global. De acuerdo con [SALEH ELMOHAMED 1997] entre las ventajas de estos algoritmos se pueden citar:

- Resulta un modelo sobre el que se puede implementar de forma relativamente sencilla este tipo de problemas.
- Su aplicabilidad a la mayoría de los problemas de optimización con una estructura combinatoria.
- Su capacidad para ofrecer soluciones razonablemente buenas a la mayoría de los problemas, aunque hay una cierta dependencia de la planificación del enfriamiento y los movimientos que se realicen.
- La facilidad con la que se puede combinar este tipo de algoritmos con otras técnicas heurísticas como los sistemas expertos, los algoritmos genéticos, las redes neuronales, etc, consiguiendo sistemas híbridos que pueden resultar de gran potencia en la resolución de problemas muy complejos.

Por otra parte, también se pueden citar algunos aspectos que pueden limitar su utilización:

- Se necesita elegir con mucho cuidado los movimientos que se realizan, así como los parámetros que se van a utilizar para tratarlo, como por ejemplo la tasa de enfriamiento.
- Una ejecución del problema puede requerir mucho tiempo de cálculo.
- Puede que sea necesario realizar muchas ejecuciones para encontrar una solución satisfactoria.
- Dependiendo de los parámetros elegidos, las soluciones que se van encontrando pueden ser poco estables, “saltando” mucho de unas a otras sin encontrar una

solución buena con la rapidez suficiente lo que obliga a retocar los parámetros con las distintas ejecuciones.

### **2.1 El algoritmo de enfriamiento lento simulado**

La estructura básica de un algoritmo de enfriamiento lento simulado puede corresponderse con el siguiente pseudocódigo:

```
Generar una solución inicial
Calcular el coste de la solución
Calcular la temperatura inicial
Mientras (no se termine)
  Mientras (no se alcance un equilibrio)
    Generar una vecindad de la solución
    Calcular el coste de la nueva solución
    Si el nuevo coste es menor tomarla
      como nueva solución
    si no, si se acepta el incremento de coste
      para la temperatura actual
      tomarla como nueva solución
  Actualizar la temperatura
```

En él se puede observar que tras una generación inicial de la solución, que puede ser aleatoria u obtenida mediante algún tipo de heurística, se calcula el coste de la misma de acuerdo con la función de coste establecida para el problema, así como la temperatura inicial del sistema.

Dependiendo de las características del problema, principalmente de la complejidad del mismo, suele ser preferible buscar una heurística que “facilite” encontrar una buena solución o dejar que se genere una solución de forma completamente aleatoria a partir de la cual se irán calculando el resto de soluciones. En este último caso el cálculo de la temperatura inicial para el sistema puede convertirse en un elemento de gran importancia.

Para decidir la temperatura inicial con la que se inicia la ejecución del algoritmo hay que tener en cuenta que es la que va a permitir, con una cierta probabilidad, aceptar soluciones cuyo coste empeora la solución de la que ya se dispone. De acuerdo con [WHITE 1984] esta temperatura inicial se puede calcular utilizando la medida de la desviación estándar de los cambios en coste en un cierto número de nuevas soluciones en la vecindad de la inicial.



Una forma práctica de obtener la temperatura inicial es obtener una temperatura en la que se permitan prácticamente todos los cambios posibles en el sistema. Para ello se elige una temperatura cualquiera y se van generando soluciones en la vecindad de la inicial, aumentando la temperatura sucesivamente hasta que se aceptan soluciones de la vecindad con una determinada tasa.

Posteriormente se entra en un bucle que se repite mientras no se termine. Esta condición de terminación puede ser tan sencilla como que ya se haya ejecutado el bucle un determinado número de veces sin que se haya encontrado ninguna nueva solución. Otra condición apropiada que se puede utilizar es que se haya alcanzado un número dado de veces un valor mínimo de coste, lo que presupone que en repetidas simulaciones se llega a dicho coste, una combinación de ambas o cualquier otra.

Dentro del bucle se deja que el sistema evolucione para una determinada temperatura y una vez que se le ha dejado evolucionar lo considerado suficiente se procede a reducir la temperatura. El proceso de actualización de la temperatura puede seguir distintos patrones o algoritmos de enfriamiento que se comentarán más adelante.

En el bucle que se repite mientras no se alcance un equilibrio, se trata de conseguir generar soluciones vecinas de la solución actual e ir evaluándolas en dos sentidos: Si el coste de la nueva solución generada es menor que la anterior se acepta y se toma como nueva solución para evolucionar; también se toma como solución si el incremento de coste que pueda suponer para la temperatura actual es probabilísticamente aceptado. En este segundo caso se genera un número aleatorio y se comprueba si su valor es menor que  $e^{-\Delta C/T}$  siendo  $\Delta C$  el incremento de coste y  $T$  la temperatura actual del sistema.

La determinación de cuando terminar suele tener en cuenta la generación de nuevas soluciones o el tamaño de la vecindad. Una condición típica es considerar que se ha llegado al equilibrio si ya se han aceptado un determinado número de vecinos o se han evaluado un cierto número de soluciones que se determina del orden del tamaño de la vecindad.

## ***2.2 Pasos de desarrollo del problema***

De acuerdo con [MELÍCIO 1996] en el diseño de un algoritmo de enfriamiento lento simulado uno debe considerar cinco elementos:

- El espacio de búsqueda, es decir, el conjunto de todas las soluciones.

- El conjunto de movimiento, es decir, la vecindad de las soluciones.
- La función de coste, es decir, un valor que indique la calidad de una solución.
- La planificación del enfriamiento, es decir, cómo se enfría el sistema.
- Las estructuras de datos, es decir, cómo representar el sistema y las soluciones eficientemente.

Siguiendo estos elementos, y considerando cada uno de ellos se puede encontrar los siguientes aspectos a considerar en cada uno que serían los siguientes:

1. **El espacio de búsqueda:** Es el conjunto de todas las soluciones posibles. En general se tratará de un problema combinatorio, por lo que el espacio de búsqueda será un conjunto de tuplas de todos los elementos de forma que la solución cumpla determinadas restricciones. El interés de este tipo de algoritmos radica en que el espacio de búsqueda de la solución no es abordable por fuerza bruta debido a su enorme tamaño. Por ejemplo, en el caso de asignación de 20 profesores a 20 horarios el espacio de búsqueda es  $20!$ , aproximadamente de  $2,4 \times 10^{18}$ . Y se trataría de un problema relativamente sencillo en número de profesores.
2. **El conjunto de movimientos posibles:** Es el conjunto definido para la vecindad de una solución. Cualquier movimiento en una vecindad debería definirse de manera que el cálculo del nuevo valor de la función de coste de la solución generada fuese sencillo. De esta forma será fácil evaluar un gran número de movimientos lo que nos llevará a evaluar un gran número de soluciones. Un movimiento define la estructura de cambio en la solución que permite generar un nuevo elemento del espacio de soluciones. Este movimiento puede tener en cuenta, o no, las restricciones impuestas sobre la solución final a obtener. En la elección de la forma de generar los movimiento es vital verificar que el sistema elegido es capaz de generar todas las soluciones posibles, sean o no las que se buscan.
3. **La función de coste:** Es la función que representa la energía del sistema. Esta función evalúa la calidad de las distintas soluciones generadas. Lo ideal es que esta función se pueda calcular de manera incremental de forma que el cálculo de la variación de coste debido a un movimiento sea lo más sencilla y eficiente en tiempo de cálculo. Una forma clásica de un movimiento, que se cita en la bibliografía, implica el intercambio de dos elementos. [MELÍCIO 1996]

propone el uso de un movimiento único aunque ello genere soluciones no viables. En el cálculo de esta función de coste se suele valorar muy negativamente el impacto de una solución que incumple restricciones impuestas sobre la solución final. De esta forma se intenta que la solución final cumpla en el mayor grado posible que se pueda encontrar todas las restricciones que debe cumplir una buena solución.

4. **El modelo de enfriamiento:** Este modelo implica la forma en que se controla el parámetro de control, es decir, la temperatura  $t$ , y cómo se consigue reducir. Existen distintos modelos de enfriamiento. En [ABRAMSON 1996] se puede encontrar una lista de los principales modelos, incluyendo algunos donde se realizan procesos de recalentamiento controlado. El modelo de enfriamiento es un elemento fundamental ya que afecta a que la solución se encuentre rápidamente y sea de buena calidad. Comenzar con una temperatura muy baja, terminar con una temperatura alta o enfriar el sistema demasiado rápidamente puede conducir a soluciones subóptimas. La ventaja de este tipo de algoritmos [ABRAMSON 1991] frente a otros algoritmos del tipo Hill Climbing [JACOBSON, 2004] radica en que además de ir buscando soluciones que mejoren la anterior se permiten ciertos incrementos de coste teniendo en cuenta cierta probabilidad que depende de la temperatura del sistema.
5. **Las estructuras de datos:** Se trata de la forma en que el diseñador consigue representar los datos necesarios para realizar el proceso de búsqueda de soluciones. La capacidad de encontrar una buena representación del problema va a influir decisivamente en el tiempo que se tarda en encontrar una solución y por tanto en la capacidad de lograr una solución buena en un tiempo razonable. La elección de las estructuras de datos internas que modelan el sistema y los algoritmo que se emplean para su manejo, que suelen venir asociadas a tipo de estructuras de datos seleccionadas puede hacer que varíe significativamente el tiempo empleado en generar las soluciones y evaluar el coste de las mismas.

### ***2.3 Alternativas en el sistema de enfriamiento lento simulado:***

Dentro del proceso de enfriamiento lento simulado se puede tomar dos posiciones básicas de afrontar el proceso de generación de nuevas soluciones en la vecindad de una dada:

- Buscar siempre sobre soluciones viables que cumplan las restricciones duras del problema. Con este sistema todas las soluciones son soluciones posibles. La

desventaja del mismo puede estar en que el sistema se centre en entornos de soluciones sin ver otros entornos de soluciones que pueden ser mejores de acuerdo con la función de coste, o incluso que no llegue a encontrar ninguna solución factible completa, bien por que no existe o por que no es capaz de generarla.

- Buscar en el espacio de soluciones sin evitar las soluciones no viables pero penalizando convenientemente las soluciones no viables. De esta forma el sistema podría utilizar alguna de estas soluciones no viables para saltar en el espacio de soluciones a otra zona desde la que continuar la exploración del mismo y conseguir llegar a zonas del mismo que de otra forma no se llegarían a evaluar.

### **3 El problema de los horarios**

El problema de los horarios, en su formulación general, consiste en asignar un conjunto de recursos (profesores, aulas, asignaturas, etc.) de manera que se cumplan un cierto conjunto de restricciones de diferente tipo relativo a la combinación de estos recursos.

En [COOPER 1995] se demuestra que este problema es NP-Completo. En particular demuestra que esta característica de NP-Completo ocurre en cuanto los estudiantes pueden elegir asignaturas, las clases varían en duración o se imponen cualquier tipo de condiciones sobre los horarios de las clases, lo que ocurre en cualquier formulación realista del problema. En casos realistas, la asignación de clases a un único profesor (una vez establecidos los horarios) es NP-Completo. Además, aunque se pueden asignar horarios apropiados a todas las clases de un grupo de estudiantes, en cuanto se tengan dos grupos el problema es, también, NP-Completo. Por ello, en general se considera que este problema es siempre NP-Completo.

El hecho de que el problema se considere NP-Completo implica que no se puede aplicar de forma general ningún algoritmo que nos permita obtener la solución óptima al problema de acuerdo con los requisitos establecidos en un tiempo razonable, especialmente si el tamaño del problema crece.

#### ***3.1 Definición del problema***

El problema de asignación de horarios tiene distintas variantes. Entre ellas se pueden encontrar las siguientes:

1. El problema de asignación de profesores. En este problema se trata de encontrar una asignación de profesores a una asignación de grupos de clase ya establecidos de forma que se cumplan ciertas restricciones sobre los profesores, como que un profesor no puede dar clase a dos grupos en la misma hora.
2. El problema de creación del horario de cursos. En este problema se trata de realizar una planificación semanal de todas las clases de un conjunto de cursos de forma que se minimicen los solapamientos de impartición de asignaturas con estudiantes en común.
3. El problema de creación de horarios de exámenes. En este problema se trata de crear un calendario de exámenes de un conjunto de cursos que evite los solapamientos de exámenes con estudiantes en común, espaciando los exámenes de los estudiantes lo máximo posible.

### *3.1.1 Planteamiento del problema a resolver*

Para el presente trabajo se va a tratar el primero de los problemas eligiendo una formulación real. El problema es suficientemente sencillo, pero a la vez completo como para extraer resultados del mismo y poder ponerlo en funcionamiento. Para ello se dispone de los siguientes elementos:

- Un conjunto de grupos de varias asignaturas con un horario ya establecido. Se supone que existe más de un grupo de una o varias de las asignaturas.
- Cada asignatura se impartirá a uno o varios grupos.
- Se define un grupo como un conjunto de intervalos horarios. Para simplificar el problema los intervalos horarios serán siempre de una hora. De esta forma un grupo que se imparta en tres horas tendrá tres intervalos horarios independientemente de que algún día dos de esas horas se imparta más de uno.
- Cada grupo tendrá su propio horario que puede ser coincidente con horarios de otros grupos aunque se trate de la misma asignatura. Esta coincidencia puede ser parcial o total.
- Si un grupo tiene un día más de un intervalo horario, éstos serán consecutivos.

- Para el problema, los grupos vendrán preestablecidos y sus intervalos horarios ya estarán planificados previamente.
- Para impartir las clases se cuenta con un conjunto de profesores. Cada profesor tendrá asignada una determinada carga lectiva a priori, es decir, un número máximo de horas semanales que puede impartir.
- Un profesor tendrá un conjunto de asignaturas de las que puede impartir clase.
- Existirán profesores contratados a tiempo parcial y profesores contratados a tiempo completo. Los profesores a tiempo completo no tienen, básicamente restricciones de horarios, mientras que los profesores a tiempo parcial pueden tener restricciones en cuanto a días en los que pueden impartir clases o en cuanto a determinadas bandas horarias.

### 3.1.2 Restricciones al problema

La asignación de profesores a los grupos debe hacerse siguiendo un conjunto de criterios. Estos criterios los vamos a dividir en tres apartados: restricciones fuertes, que son el tipo de restricciones que se deben cumplir siempre, de forma que el no cumplimiento de una de ellas generaría una solución no factible; restricciones medias, que son restricciones en las que estamos interesados que se cumplan al máximo, ya que puede condicionar el hecho de que realmente se pueda realizar dicha asignación y restricciones débiles, que agrupan un conjunto de condiciones adicionales que permitirían mejorar el horario que resulta para el conjunto del profesorado.

El conjunto básico de restricciones que debe cumplir la solución al problema son las siguientes:

#### **Restricciones fuertes**

- Un profesor no puede impartir clase en dos grupos distintos a la misma hora.
- Hay profesores que solo pueden impartir clase determinados días o en determinados horarios.
- Un grupo no puede tener más de un profesor asignado.

#### **Restricciones medias**

- Un profesor con contrato a tiempo completo no puede tener clases en un intervalo superior a nueve horas.
- Todo grupo debe tener asignado un profesor. (Es de tipo medio por que siempre se puede buscar otro profesor que imparta dicho grupo).

### **Restricciones débiles**

- Un profesor con contrato a tiempo parcial debe tener sus clases agrupadas lo máximo posible eliminando horas muertas entre clases.
- El número de días lectivos de un profesor con contrato a tiempo parcial debe ser el mínimo posible.

#### *3.1.3 Estructura del espacio de soluciones*

Para el problema enunciado una solución tendrá el aspecto de un conjunto de tuplas (Profesor, Grupo). Todas las tuplas cubrirán el conjunto de asignaciones de grupos que pueda haber. En este sentido el espacio de soluciones tiene una estructura muy simple.

Al tener el espacio de soluciones representado de esta forma podemos tener en cuenta ya el cumplimiento de la restricción que indica que todos los grupos deben tener un profesor. Con esta forma de crear el modelo de la solución nos aseguramos que la solución siempre es completa y que, además, los movimientos en la misma pueden resultar sencillos como se presenta en la siguiente sección.

#### *3.1.4 Conjunto de movimientos*

Una vez se ha establecido un patrón de solución en la que los profesores cubran todos los grupos, los movimientos pueden seguir el siguiente pseudocódigo:

```
Se eligen dos tuplas al azar,  
que sean de distinto profesor  
que sean de la misma asignatura  
  
Se intercambian los grupos asignados en dichas  
tuplas  
  
Se recalcula la función de coste
```

Como se puede observar, con este esquema el cálculo de la función de coste se puede hacer fácilmente de forma incremental, pues se puede calcular como el coste ya

calculado de la solución anterior menos el coste de cada uno de los profesores al asignarles el grupo que se les quita, mas el coste de realizar la nueva asignación de coste sólo a esos dos profesores elegidos. Este tema se trata en la siguiente sección.

### 3.1.5 Función de coste

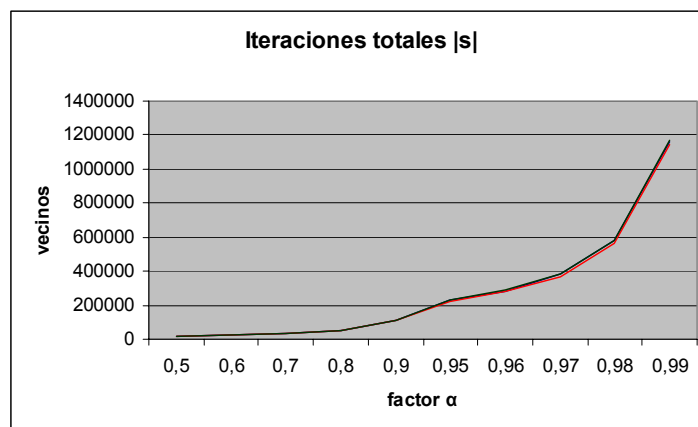
La función de coste es un proceso de evaluación en el que se da mayor peso cuanto más grave es la restricción que se evalúa. Así por ejemplo, se dará mucho más peso a una restricción dura como que un profesor no puede dar dos clases a la misma hora el mismo día, que a una restricción débil como que el número de días debe ser el menor posible.

De esta forma se prima que se busquen soluciones que cumplan las restricciones fuertes y, posiblemente las medias, intentando, además que se lleguen a cumplir al menos en la mayor medida posible las débiles.

La estructura del cálculo de la función de coste es muy sencilla comprobando las condiciones que debe cumplir la solución e incrementando el coste de la misma si la condición evaluada no se cumple. De esta forma la evaluación de la función de coste consta de un conjunto de condiciones como las siguientes:

Para cada solape de horario de un profesor  
`coste = coste + COSTE_SOLAPE;`

En un apartado posterior puede ver el código real utilizado para la función de coste.



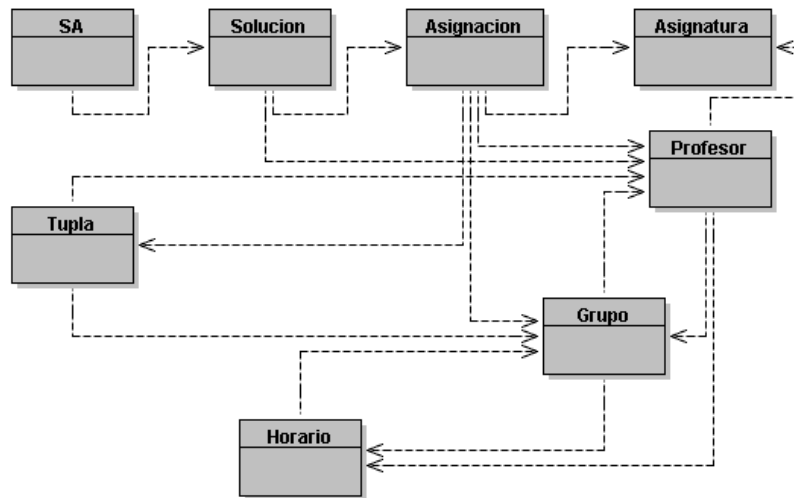


**Figura 1.** Evolución del número de vecinos calculados en función del factor de enfriamiento del algoritmo.

### 3.1.6 Planificación del enfriamiento

En el programa se ha utilizado un enfriamiento exponencial clásico, con un factor de 0,95. Es decir la temperatura se multiplica por dicho factor en cada repetición del bucle, reduciéndose paulatinamente. La elección de esta temperatura supone un compromiso entre velocidad de cálculo y calidad de la solución.

En la Figura 1 se puede ver una curva que representa el número de soluciones vecinas generadas durante la ejecución del algoritmo para ciertos valores del factor de enfriamiento. Como se puede observar, el crecimiento del número de vecinos calculados y, por tanto, del tiempo de ejecución del algoritmo es una exponencial. En la figura de ejemplo se ha utilizado como número de iteraciones de la cadena de Harkov el tamaño de la vecindad  $|S|$  de una solución.



**Figura 2.** Esquema de clases del programa.

## 4 Implementación y análisis de los resultados

El programa se ha escrito en Java con la estructura que se puede observar en el diagrama de la Figura 2. En él se puede observar, por ejemplo, como una asignación (solución

actual) está formada por un conjunto de tuplas, donde cada tupla estará formada por pares profesor y grupo.

Las clases SA, por Simulated Annealing, *Solución y Asignación* son las que llevan el peso de la ejecución, propiamente dicha del algoritmo de enfriamiento lento simulado. El resto de las clases se utiliza para modelar el problema de planificación de horarios, donde se observa que se manejan los conceptos de *Asignatura, Profesor, Grupo, Horario y Tupla*.

La parte más interesante para el artículo es el método de la clase Profesor que se utiliza para evaluar una solución. Este método es el que se muestra a continuación:

```
public int calculaCoste() {
    coste = 0;

    Grupo grupo1, grupo2;

    // Los horarios de los grupos de un profesor
    // no deben de solaparse.
    for(int i = 0; i<gruposAsignados.size(); i++){
        grupo1=(Grupo)gruposAsignados.elementAt(i);
        for(int j=i+1;
            j<gruposAsignados.size(); j++) {
            grupo2 =
                (Grupo)gruposAsignados.elementAt(j);
            if (grupo1.solapanHoras(grupo2)) {
                coste += 1000;
            }
        }
    }

    // El horario de un profesor ha de ser
    // compatible con su horario.
    for(int i = 0; i<gruposAsignados.size(); i++){
        grupo1 =
            (Grupo)gruposAsignados.elementAt(i);
        if (! grupo1.casanHoras(horarioPosible)) {
            coste += 400;
        }
    }
    // Un profesor debe tener clase el mínimo
    // número de días
    int diasConClases;
```

```
if (! esTC){
    diasConClases = diasConClases();
    if (diasConClases > maxDías) {
        coste+=(diasConClases - maxDías) * 30;
    }
}

// El profesor debe de tener el mínimo número
// de huecos
int numHuecos;
if (!esTC) {
    numHuecos = numHuecos();
    coste += numHuecos * 1;
}

// Si es tiempo completo debe tener hasta 9
// horas de clase al dia.
int intervalo = intervaloHorario();
if (intervalo >= 9)
    coste += 10*(intervalo-8);

return coste;
}
```

Esta evaluación se hace para el profesor teniendo en cuenta que gruposAsignados son todos los grupos a los que impartirá clase, horarioPosible es un horario que representa los intervalos en que es imposible que dé clase, esTC indica si el profesor está contratado a tiempo completo y maxDías indica el número máximo de días que se le puede dar clases a un profesor a tiempo parcial.

La configuración que se utiliza para representar a un profesor tiene el siguiente aspecto:

```
[Profesor]
Pilar
L8 L9 L10 L11 L12 L13 L14
TC
ASI 3
AS2 1

[Profesor]
Almudena
-
TP3
AS2 3
```

```
[Profesor]
Jesús
V1 V13 V14 V15 V16
TC
ASI 1
AS2 2
OTR 1
```

En este ejemplo se indica que el profesor de nombre Pilar no puede dar clases los intervalos que empiezan desde los lunes a las 8 hasta las 14 (es decir, entre las 8 y las 15 horas), es una profesora a tiempo completo (TC) e imparte 3 grupos de la asignatura ASI y un grupo de la asignatura AS2.

De forma correspondiente, la profesora de nombre Almudena no tiene restricciones de horarios, es una profesora a tiempo parcial y se le deben intentar asignar los grupos en un máximo de tres días. De forma similar se interpretaría para el profesor Jesús.

Para la representación de los grupos y asignaturas se utiliza una simple lista como la siguiente, que no pretende ser más que un ejemplo significativo:

```
# grupos de la Asignatura 1.
ASI0101 M11 X11 J11
ASI0102 L12 M12 X12
ASI0103 X12 J12 V12
ASI0104 M13 J13 V13

# grupos de la Asignatura 2.
AS20101 L10 M13 J11
AS20102 L12 X11 J11
```

En ella, como ya se puede suponer se indica un código que identifica el grupo de la asignatura y, a continuación, la lista de intervalos horarios. En el ejemplo cada uno de los grupos de asignaturas se imparte en tres horas a la semana. También se puede observar, por referencia al archivo de configuración de los profesores, que las tres primeras letras del grupo indican la asignatura.

#### **4.1 Resultados obtenidos**

El programa actual se ha probado con un conjunto bastante amplio de profesores y asignaturas-grupos. En particular con un grupo de 27 profesores, tanto a tiempo completo como a tiempo parcial y con distintas variaciones de restricciones tanto para unos como para otros. También había un total de 42 asignaturas diferentes, sumando un total de 98 grupos distintos. Entre los grupos más numerosos de una asignatura dada, había una asignatura con 22 grupos diferentes, otra con 10 grupos distintos, varias con 6 y 4 grupos.

Con esta complejidad de espacio de soluciones, y ante un entorno cambiante donde las restricciones de los profesores pueden verse modificadas, o incluso los mismos profesores pueden cambiar, el programa ha resultado de una gran eficacia, sorprendiendo a veces con sus resultados.

Los resultados más interesantes que se han obtenido parten de la posibilidad que da de realizar distintas asignaciones o conjuntos de asignaciones de asignaturas a determinados profesores, de forma que se puede llegar a una estructura de solución de calidad de forma muy cómoda.

Con este principio se pueden pensar en muchas variantes que permitan comprobar la bondad de este método para otro tipo de planificaciones.

## 5 Conclusiones

En este trabajo se ha realizado un estudio del modelo de resolución de problemas mediante técnicas de enfriamiento lento simulado aplicadas a la resolución del problema de los horarios. De este estudio se puede deducir que la aplicación de este tipo de técnica resulta relativamente sencilla de implementar, y es aplicable de forma general a cualquier tipo de problema con una estructura combinatoria.

Sin embargo, no nos es posible extrapolar los resultados de este problema ya que no hemos tenido posibilidad de comparar este mismo problema con otras técnicas de tipo similar.

Asimismo, es difícil considerar la influencia de los distintos parámetros del algoritmo a la resolución. Se han probado cambios en varios de los parámetros, siendo de forma consistente los resultados bastante estables. Para un análisis más detallado habría que comprobar con los distintos parámetros que influyen en la ejecución del mismo. De hecho así lo indica [SALEH ELMOHAMED 1997] afirmando que este tipo de técnica

es dependiente de la elección del modelo de enfriamiento una cuidadosa selección de la generación de vecindad y de la generación de una buena solución inicial.

Algunas de las conclusiones más significativas de los artículos consultados indican que:

1. Las técnicas de enfriamiento lento simulado permiten resolver problemas de planificación muy complejos y de forma muy competitiva en tiempo, pero hay que ser muy cuidadoso en la parte de la implementación dependiente del problema. [MELÍCIO 1997]
2. Existe una mejora muy importante en el rendimiento del algoritmo si se elige una solución inicial buena antes de aplicarle las técnicas de enfriamiento lento simulado. El mejor modelo de enfriamiento se consigue con el enfriamiento adaptativo con recalentamiento. [SALEH ELMOHAMED 1997]
3. El uso de una solución inicial heurística empezando el enfriamiento lento simulado a una menor temperatura es la mejor opción para conseguir una solución cercana a la óptima en el menor tiempo [MELÍCIO 1996]

En este momento estamos planteando la utilización de este tipo de técnicas para otros problemas similares que se suelen resolver en un entorno docente. En particular se está trabajando en la elaboración de calendarios de exámenes, con resultados iniciales muy interesantes.

## 6 Bibliografía

ABRAMSON D, KRSHANAMOORTHY M y DANG H. (1996) “*Simulated Annealing Cooling Schedules for the School Timetabling Problem*”.  
<http://citeseer.nj.nec.com/104097.html>

ABRAMSON D. (1991) “*Constructing School Timetables using Simulated Annealing: Sequential and Parallel Algorithms*”. Management Science Vol 37, 93-113,  
<http://citeseer.nj.nec.com/abramson91constructing.html>

COOPER, TIM B. y KINGSTON, JEFFREY H. (1995) “*The Complexity of Timetable Construction Problems*”. Proceedings of the First International Conference on the Practice and Theory of Automated Timetabling (ICPTAT '95).  
<http://citeseer.nj.nec.com/232089.html>

- JACOBSON, SHELDON H. (2004). *Analyzing the Performance of Generalized Hill Climbing Algorithms*. Journal of Heuristics. Vol 10, pp. 387-405. Kluwer Academic Publishers.
- KIRKPATRICK S., GELATT C.D. y VECCHI M.P. (1983) "Optimization by Simulated Annealing" *Sci-ence*, vol. 220, 45-54.
- MELÍCIO F., CALDEIRA P. y ROSA A. (1996) "Implementation aspects of Simulated Annealing on Timetabling".
- MELÍCIO F., CALDEIRA P. y ROSA A. (1997) "Solving the timetabling problem with simulated annealing".
- SALEH ELMOHAMED M.A. y GEOFFREY FOX, (1997) "A comparison of Annealing Techniques for Academic Course Scheduling". Practice and Theory of Automated Timetabling II, Selected Papers from the 2nd International Conference, PATAT'97. <http://citeseer.nj.nec.com/elmohamed97comparison.html>
- WHITE, S.R. (1984) "Concepts of Scale in Simulated Annealing", Noviembre de 1984.