

Asignación de Aulas aplicando Simulated Annealing

Silvia Rodríguez de Ryan⁽¹⁾
sryan@unas.edu.ar

Cristian Martínez⁽¹⁾
camzeta@hotmail.com

Daniel Morales⁽¹⁾
dmorales@unas.edu.ar

⁽¹⁾Facultad de Ciencias Exactas - Consejo de Investigación -CIDIA
Universidad Nacional de Salta

Buenos Aires 177 (4.400) Salta – Argentina

Resumen

El objetivo de este trabajo es presentar un modelo computacional para “asignar en forma óptima” aulas del Campus Universitario a las asignaturas que las requieran. Se plantea un modelo matemático de asignación equilibrado y un algoritmo heurístico basado en Simulated Annealing codificado en PERL y usando MySQL como motor de base de datos. Para cada día de la semana (d) el horario de clase disponible, se divide en 14 franjas de 1 hora cada una, se determina la franja horaria (k) más requerida, y es ella la que se optimiza, asignando también la misma aula si la comisión (teoría o práctica) tiene el mismo horario otro día de la semana. A partir de esa franja (k) óptima, se continúa optimizando las franjas vecinas (k-1) y (k+1), hasta completar todo el horario requerido para ese día. Se realiza el mismo procedimiento para los otros días de la semana.

Los resultados obtenidos muestran la conveniencia del modelo y del uso de esta técnica heurística para resolver problemas de asignación y en particular el problema en estudio, ya que el porcentaje de sillas vacías o cantidad de alumnos sin bancos fue notoriamente inferior respecto a la asignación manual.

PALABRAS CLAVES: Optimización, Asignación, Simulated Annealing, Simulación, Timetabling.

I.- Introducción

La Universidad Nacional de Salta cuenta con seis Facultades, 36 carreras y un total de 810 asignaturas, las cuales se imparten en el Campus de la Universidad. Cada Facultad posee un número mínimo de aulas las que son destinadas a clases especiales o a asignaturas con pocos alumnos, mientras que el resto de las materias se dictan en el blok de aulas comunes y anfiteatros del Campus (65 en total).

La tarea de asignación de aulas es realizada en la actualidad en forma manual por bedelía, luego de verificar que existen a lo sumo la misma cantidad de asignaturas que solicitan un aula que el número total de aulas disponibles.

El objetivo de este trabajo es presentar un modelo computacional para “asignar en forma óptima” aulas pertenecientes al blok de aulas comunes del Campus a las asignaturas que las requieran. Para cumplir con ello, se plantea un modelo matemático de asignación equilibrado y un algoritmo heurístico basado en Simulated Annealing codificado en PERL (Practical Extraction and Report Language) y MySQL como motor de base de datos.

Los resultados obtenidos muestran la conveniencia del modelo y del uso de esta técnica heurística para resolver problemas de asignación y en particular el problema en estudio, ya que el porcentaje de sillas vacías o cantidad de alumnos sin bancos fue notoriamente inferior respecto a la asignación manual.

II.- Simulated Annealing (SA)

Simulated annealing (SA) es actualmente una de las técnicas heurísticas más usadas para resolver problemas de optimización combinatoria y se adapta muy bien para problemas de asignación, como el que tratamos en este trabajo.

Conceptualmente, SA es un método de búsqueda por entornos, que intenta evitar el quedarse atrapado en óptimos locales relajando la condición de mejora continua impuesta por otros algoritmos. SA permite cambiar la solución actual, bajo criterios probabilísticos, por soluciones factibles peores. Esta permisividad va disminuyendo en función de un parámetro, denominado temperatura, por analogía con el fenómeno físico que da origen a este algoritmo.

La implementación de SA requiere la especificación de :

- Soluciones factibles y la determinación de sus costos.
- Solución factible inicial.
- Temperatura inicial, T_0 .
- Definición del entorno de búsqueda, $N(S)$.
- Estado de equilibrio a cada nivel de temperatura, $nrep$.
- Esquema de enfriamiento, $\alpha(T)$.
- Criterio de término.

Algoritmo

Seleccionar los parámetros T_0 , α , $nrep$, T_f
 $T \leftarrow T_0$;
Generar una solución inicial S_{act}
Repetir
 $I \leftarrow 1$
 Repetir
 Calcular una solución candidato (S_{cand}) $\in N(S_{act})$;
 $\delta \leftarrow C(S_{cand}) - C(S_{act})$
 Si ($U(0,1) < e^{-\delta/T}$) **ó** ($\delta < 0$) **Entonces** $S_{act} \leftarrow S_{cand}$
 $I \leftarrow I+1$
 Hasta $I > nrep$
 $T \leftarrow \alpha(T)$ (según el programa de enfriamiento seleccionado)
Hasta $T \leq T_f$

III.- Procedimiento de optimización

Para cada día de la semana el horario de clase disponible, entre las 8:00 y 22:00 horas, se divide en 14 franjas de 1 hora cada una. Para el día de la semana (d) se determina la franja horaria (k) más requerida para el dictado de clases y es ella la que se optimiza, asignando también la misma aula si la **comisión** de la asignatura solicitante tiene el mismo horario otro día de la semana. A partir de esa franja (k) óptima, se continúa optimizando las franjas vecinas (k-1) y (k+1), hasta completar todo el horario requerido para ese día. Se realiza el mismo procedimiento para los otros días de la semana, considerando que algunas comisiones ya tienen un aula asignada.

IV.- Modelo Matemático

Factores involucrados

- Número de alumnos por comisiones.
- Capacidad del aula.

Nomenclatura: Teniendo en cuenta que una asignatura puede tener varias comisiones de teoría y/o práctica y cada comisión distintos horarios de clases, la clase a asignar un aula la denominamos "comisión". Así, la nomenclatura definida es la siguiente:

- d: día de la semana
- k: franja horaria
- i: comisión de una asignatura
- j: aula
- x_{ij} : comisión i asignada al aula j
- na_i : número de alumnos de la comisión i
- ca_j : capacidad del aula j
- m: número de comisiones a asignar un aula
- n: número de aulas disponibles

Modelo: El objetivo del problema es minimizar el porcentaje de sillas vacías y número de alumnos sin banco en el dictado de clase de una comisión.

$$\min \sum_{j=1}^n \sum_{i=1}^m p_i |na_i - ca_j| x_{ij} \quad \forall k = 1..14, \quad \forall d = 1..6 \quad (1)$$

sa :

$$\sum_{j=1}^n x_{ij} \leq 1 \quad \forall i = 1..m$$

$$\sum_{i=1}^n x_{ij} = 1 \quad \forall j = 1..n$$

$$m \leq n$$

$$x_{ij} \in [0,1] \quad \forall i, j$$

siendo p_i es el factor de ponderación que mide la **calidad** de la asignación, relacionado con el porcentaje de sillas vacías o cantidad de alumnos sin banco, tal que:

$$porc = \frac{ca_j - na_i}{na_i}$$

$$p_i = \begin{cases} 1 & \text{si } |porc| \leq 0.1 \text{ "buena asignacion"} \\ |porc| * 100 & \text{si } porc < -0.1 \text{ "más alumnos que bancos"} \\ porc * 50 & \text{si } porc > 0.1 \text{ "más bancos que alumnos"} \\ ca_j * 500 & \text{si se trata de una comisión ficticia} \end{cases}$$

V.- Implementación del Algoritmo

Los datos de las asignaturas y las aulas fueron almacenados físicamente en una base de datos Mysql para su posterior consulta y procesamiento. La necesidad del uso de una base de datos se debió al constante acceso y modificación de los datos del problema.

Las tablas que se usaron fueron las siguientes:

- *Materia*: código de la materia, nombre de la materia, año, cuatrimestre, facultad, carrera, responsable del dictado, cantidad de alumnos de la materia.
- *Comisión*: código de la materia, código de comisión, cantidad de alumnos, clase, docente.
- *Aula*: código del aula, capacidad, estado.
- *Horarios de aulas*: código del horario, día, franja horaria, asignada o no.
- *Horario de la comisión*: código del horario, día, hora de inicio, hora de fin, asignada o no.

➤ Estructura de datos

Para cada franja horaria de un día de la semana, se definen dos arreglos: uno para las comisiones a asignar un aula (código_comisión, cantidad de alumnos) y el otro para aulas (código_aula, capacidad). Estos dos arreglos nos permiten generar la solución inicial (S_{ini}) y las soluciones candidatas (S_{cand}).

➤ Solución factible

Es un arreglo cuyos elementos son pares ordenados (código_comisión, código_aula) para cada franja horaria de un día de la semana. Si el número de comisiones es menor que el número de aulas, se generan “comisiones ficticias” sin alumnos.

➤ Solución inicial (S_{ini})

Las primeras componentes de los pares ordenados que forman la solución se obtienen al azar mediante una permutación entre los elementos del arreglo de comisiones de la franja horaria a optimizar, mientras que las segundas componentes corresponden a los elementos del arreglo de aulas, el cual se mantiene fijo y ordenado en forma decreciente según la capacidad del aula.

➤ Soluciones candidatas (S_{cand})

Se generan aleatoriamente dos posiciones y se intercambian los elementos correspondientes del arreglo “comisiones”.

$$S_{act} = ((com_1, aula_1); (com_2, aula_2), \dots, (com_n, aula_n))$$

$$Comisión = (com_1, com_2, com_3, \dots, com_n)$$

$$Aula = (aula_1, aula_2, aula_3, \dots, aula_n)$$

$$S_{cand} = ((com_1, aula_1); (com_n, aula_2), \dots, (com_2, aula_n))$$

➤ Función Costo (C)

Esta función, dada por (1), mide la calidad de la asignación de una solución factible.

➤ Aceptación de soluciones candidatas

Utilizamos como función de probabilidad para aceptar soluciones peores la función $e^{-\delta/T}$, donde δ representa la diferencia de las funciones de evaluación (C) de la solución candidata (S_{cand}) y la actual (S_{act}), y T representa el parámetro temperatura. Las soluciones mejores siempre se aceptan.

➤ Temperatura inicial (T₀)

Se fija en función del costo de la solución inicial:

$$T_0 = a * C(S_{ini}) \quad \text{siendo } 0 < a < 1$$

➤ Estado de equilibrio a cada nivel de temperatura (nrep)

Para cada temperatura se realiza un número determinado de iteraciones.

➤ Función de enfriamiento

Se adopta un enfriamiento lento, tal que $T = \alpha(T) = r * T$, donde $r \in [0.8, 0.99]$.

➤ Temperatura final (T_f)

Este parámetro, el cual es próximo a cero, se lo utiliza como criterio de término del algoritmo.

VI.- Resultados Alcanzados

Se realizaron pruebas al modelo tomando como base los requerimientos de aulas solicitados en el primer cuatrimestre del año 2002.

El algoritmo fue codificado en PERL debido a que este lenguaje trata a los datos extraídos de base de datos como arreglos, lo cual lo resultó muy conveniente ya que las asignaciones “comisión-aula” se realiza a través de esta estructura de datos. El uso de MySQL como motor de base de datos se debió a la rapidez del mismo para la gestión de datos. Esto fue muy importante para el trabajo debido al tamaño de los arreglos que se usaron, la cantidad de franjas horarias y por el número de iteraciones realizadas en el Simulated Annealing.

Las primeras pruebas del modelo se realizaron tomando los siguientes parámetros:

$T_0 = 70\%$ del $C(S_{ini})$	$nrep = 10$	$\alpha = 0.85$	$T_f = 5$
-------------------------------	-------------	-----------------	-----------

La siguiente tabla muestra las mejores asignaciones del día martes sobre un total 205 solicitudes de aulas, considerando un $porc \in [-0.1, 0.1]$.

Materia	Comisión	Hora	Alumnos	Aula	Cap. Aula	Porcentaje
EC005	com1	18	200	Anff	200	0.00
EC005	com1	19	200	Anff	200	0.00
EC005	com2	10	200	Anff	200	0.00
EC005	com2	11	200	Anff	200	0.00
EC006	com1	16	200	Anfb	180	-0.10
EC006	com1	17	200	Anfb	180	-0.10
EC007	com11	16	130	Anfc	130	0.00
EC007	com11	17	130	Anfc	130	0.00
EC010	com1	16	400	Anfg	400	0.00
EC010	com1	17	400	Anfg	400	0.00
EC013	com2	20	200	Anfe	200	0.00
EC013	com2	21	200	Anfe	200	0.00
EC016	com1	18	200	Anfe	200	0.00
EC016	com1	19	200	Anfe	200	0.00
EC022	com1	16	130	A113	140	0.08
EC022	com1	17	130	A113	140	0.08
EX015	com1	8	110	A117	110	0.00
EX015	com1	9	110	A117	110	0.00
EX042	com1	16	150	Anfa	160	0.07
EX042	com1	17	150	Anfa	160	0.07
NA013	com5	8	130	Anfc	130	0.00
NA013	com5	9	130	Anfc	130	0.00

Materia	Comisión	Hora	Alumnos	Aula	Cap. Aula	Porcentaje
NA013	com6	16	130	Anfd	130	0.00
NA013	com6	17	130	Anfd	130	0.00
NA017	com1	16	140	A111	140	0.00
NA017	com1	17	140	A111	140	0.00
NA017	com1	18	140	A111	140	0.00
NA022	com1	8	120	Anfd	130	0.08
NA022	com1	9	120	Anfd	130	0.08
NA023	com1	8	200	Anff	200	0.00
NA023	com1	9	200	Anff	200	0.00
NA023	com2	10	170	A20	170	0.00
NA023	com2	11	170	A20	170	0.00
NA023	com2	12	170	A20	170	0.00
NA040	com1	10	200	Anfe	200	0.00
NA040	com1	11	200	Anfe	200	0.00
SAL001	com1	12	400	Anfg	400	0.00
SAL001	com1	13	400	Anfg	400	0.00
SAL003	com1	8	200	Anfb	180	-0.10
SAL003	com1	9	200	Anfb	180	-0.10
SAL003	com1	10	200	Anfb	180	-0.10
SAL003	com1	11	200	Anfb	180	-0.10
SAL004	com4	16	100	A14	100	0.00
SAL004	com4	17	100	A14	100	0.00

SAL007	com1	16	200	Anff	200	0.00
SAL007	com1	17	200	Anff	200	0.00
SAL010	com1	16	100	A23	110	0.10

SAL010	com1	17	100	A23	110	0.10
SAL014	com4	8	100	A23	110	0.10
SAL014	com4	9	100	A23	110	0.10

VII.- Conclusiones y Trabajos Futuros

- El método propuesto para la solución del problema de asignación de aulas, según la calidad de las soluciones obtenidas en la experimentación es superior a la realizada por Bedelía ya que, tomando como testigo algunas aulas, el porcentaje de ocupación de las mismas es notoriamente superior, aún realizando pocas iteraciones en el algoritmo debido a los valores de los parámetros utilizados.
- Se continuará el trabajo modificando los parámetros involucrados en SA y las ponderaciones usadas en la función costo, como así también la valoración del desempeño del algoritmo en cuanto al tiempo de ejecución.
- El objetivo final de este trabajo de investigación es el de desarrollar un sistema web de gestión de aulas que permitirá agilizar la consulta, solicitud y asignación de aulas en la UNSa., es por ello que se decidió codificar el algoritmo en el lenguaje PERL y MySQL.
- Se aplicará la esencia del método a otros problemas similares de asignación.

VIII.- Bibliografía

- [1] Aarts E. and Lenstra J.K. Ed , “Local Search in Combinatorial Optimization” , Ed John Wiley & Sons, (1997)
- [2] Aarts,E.,Korst,J., “Simulated Annealing and Boltzmann machines”, Wiley, 1989.
- [3] Van Laarhoven,P.,Aarts,E. “Simulated Annealing:theory and applications”, Kluwer,1988
- [4] Michelwicz, Z., “Genetic algorithms + Data Structures = Evolutionary programs”, 1996.
- [5] Laguna, M. y Moscato, P., “Optimización Heurística y Redes Neuronales” - Ed. Paraninfo, (1996).
- [6] Williams, H.P, “Model Solving in Mathematical Programminng” John Wiley & Sons (1993).
- [7] Laporte, G., Osman,I.H., “Metaheuristics in Combinatorial Optimization”, Annals of Operations Research, Vol 63, 1996.
- [8] Burke, E.K., Newall, J.P.: “A Multi-Stage Evolutionary Algorithm for the Timetable Problem”, IEEE Trans. Evol. Comput. **3** (1999), 63-74.
- [9] Reeves, Colin R., Ed., “Modern Heuristic Techniques for Combinatorial Problems”, Blackwell, Oxford (1993).

ANEXO

PERL (Practical Extraction and Report Language) fue originalmente un lenguaje para el desarrollo de programas bajo entorno UNIX. En la actualidad, existe la versión de PERL bajo Windows. Posee las siguientes características:

- Posee el poder y la flexibilidad de lenguajes de alto nivel como C.
- Al igual que los lenguajes de scripts, no requiere un compilador y enlazador para devolver código objeto. En vez de esto, lo que se hace es escribir el código e invocar a Perl para que lo ejecute (por ejemplo si se desea correr el archivo "heuristica.pl, debería escribir Perl heuristica.pl)
- Un programa desarrollado en PERL puede correr en un ambiente de red local o en Internet.
- Es de licencia libre.
- Existen en varios portales tales como cpan.org muchas unidades para ser usadas con diferentes aplicaciones: correo, seguridad, multimedia, bases de datos.
- Posee conexión a motores tales como Postgres, Interbase, MySQL como también otros a través de ODBC.

MYSQL: El software MySQL constituye en un servidor de base de datos rápido, multiusuario y multihilo. Actualmente está en la versión 4. MySQL es el motor más popular y usada del tipo OpenSource. Está desarrollado y distribuido por MySQL AB.

Se remarca lo siguiente:

- MySQL es una colección estructurada de datos. Puede usarse en una PyME como en una corporación con sedes en distintos puntos del mundo.
- MySQL es un sistema relacional de administración de base de datos.
- Es Open Source, es decir, de uso libre.
- Es de fácil uso.
- Corre en varios sistemas operativos entre ellos Windows y Linux.
- Es estable.
- Es posible conectarse a la base de datos a través de diferentes lenguajes, principalmente a través de ODBC o bien librerías dispuestas por MySQL.