# Student schedule generation with option selection at a university engineering faculty

Demetrio Rey[*,a], Luis Llave[a], Enrique Flores[a], Víctor Barrios[a], Braulio De Abreu[a], José L. Nazar[b]

[a]*Instituto de Matemáticas y Cálculo Aplicado (IMYCA), Facultad de Ingeniería, Universidad de Carabobo*
[b]*Decanato, Facultad de Ingeniería, Universidad de Carabobo. Valencia, Venezuela*

**Abstract.-**

In this paper we describe the solution algorithm for the real problem of the student schedule generation at the College of Engineering at University of Carabobo. Our approach is to allow the students to pick three complete schedules in preference order. Later, our schedule generator program will try to find a schedule equal to any of the options selected, or otherwise provide the best possible schedule. To this end, a constraint satisfability strategy divided in multiple passes is applied. For datasets considered, we found that our solution yields 100 % efficacy in obtaining schedules for all students, 98.8 % of schedules without overlap, 98.2 % of schedules with all requested courses assigned, and 74 % of satisfactory schedules with 44 % identical to one of the options. We also found that the strategy of student-constructed schedule options partially solves the scheduling problem in relatively short execution time. The program that implements the discussed algorithm has been used successfully since 2013 up to date.

**Keywords:** student schedule generation; option selection; constraint satisfability problem

# Asignación de horarios estudiantiles con selección de opciones en una facultad universitaria de ingeniería

**Resumen.-**

En este trabajo describimos el algoritmo solución al problema real de asignación de horarios estudiantiles en la Facultad de Ingeniería de la Universidad de Carabobo. El mismo consiste en permitir a los estudiantes seleccionar 3 opciones de horarios en orden de preferencia, y posteriormente el programa generador de horarios trata de encontrar un horario igual o aproximado a alguno de los solicitados. Para ello, se aplica una estrategia de satisfacción de restricciones, subdividida en fases sucesivas. En los semestres considerados en este trabajo, el resultado obtenido tiene una eficacia de 100 % de la población estudiantil con horarios asignados, 98,8 % de los horarios sin coincidencias, 98,2 % con todas las materias solicitadas, y 74 % of horarios satisfactorios con 44 % idénticos a una de las opciones solicitadas. Adicionalmente, encontramos que la estrategia de selección de opciones de horario resuelve parcialmente el problema planteado en un corto tiempo de ejecución. La versión definitiva del programa que implementa el algoritmo ha sido utilizada exitosamente en esta institución desde el año 2013 hasta la actualidad.

**Palabras clave:** asignación de horarios estudiantiles; selección de opciones; problema de satisfacción de restricciones

Recibido: noviembre 2016
Aceptado: enero 2017

## 1. Introducción

La asignación óptima de horarios de clase para los alumnos de una institución educativa, es un problema que suscita mucho interés científico, dado su inherente complejidad computacional y la

[*]Autor para correspondencia
*Correo-e:* demetrio.rey@gmail.com (Demetrio Rey)

gran demanda que tiene su aplicación por parte de las diferentes entidades de educación a cualquier nivel.

El tema de la asignación de horarios comprende, entre otros, dos aspectos principales. Primero, el problema del diseño de la oferta académica (en inglés: *Course Timetabling Problem, CTP*); y segundo, la asignación de estudiantes a las secciones de secciones disponibles en la oferta académica, llamado SSP por sus siglas en inglés (*Student Scheduling Problem*) [1].

El CTP consiste en corresponder las materias a impartir en un período académico, con los profesores, horarios y salones de clase disponibles. Este problema en sí mismo, es objeto de intensa investigación, que ha resultado en las mas diversas estrategias de solución [2, 3].

El problema objeto de este trabajo, SSP, es la asignación de alumnos a las materias solicitadas al momento de la inscripción, de acuerdo a la oferta académica existente. En instituciones educativas con gran número de estudiantes, las materias están subdivididas en secciones. El SSP consiste en asignar a los estudiantes a las secciones de las materias que hayan solicitado [4]. Los horarios resultantes no deben tener coincidencias, deben satisfacer las necesidades del alumno en cuando al estudio y tiempo disponible; y al mismo tiempo, satisfacer los requerimientos de las cátedras, en cuanto a disponibilidad de cupo y balanceo de secciones.

Se toma como objeto de estudio la solución dada al SSP en la Facultad de Ingeniería de la Universidad de Carabobo, una facultad de gran matriculación que ofrece 6 carreras de Ingeniería y cuenta con aproximadamente 9500 estudiantes activos.

## 2. El SSP

El SSP es una variación del problema de satisfacción de restricciones (*Constraint Satisfability Problem, CSP*) [4]. El CSP consiste en un conjunto de variables $V = \{v_1, v_2, \ldots, v_n\}$, sus dominios asociados $D_1, D_2, \ldots, D_n$, y un conjunto de restricciones en dichas variables. En el SSP, el conjunto $V$ son las materias solicitadas por el estudiante, y $D_i$ son las diferentes secciones de la oferta académica. El SSP consiste en etiquetar cada materia con una sección disponible. Las restricciones consisten en elegir secciones compatibles (sin coincidencia de horarios), que tengan asientos disponibles (cupo), y otras adicionales que requiera el plan de estudios. El SSP es NP-completo [5].

Existen problemas parecidos al SSP, tal como la asignación de secciones tutoriales y secciones de laboratorio a partir de las secciones de materias teóricas [6], que utilizan estrategias de solución similares al SSP y por lo tanto son consideradas en este trabajo.

Encontrar la solución óptima para el SSP y sus problemas derivados requiere de una búsqueda en un espacio-solución de tamaño combinacional, la cual se repite tantas veces como estudiantes necesiten asignación. Una búsqueda exhaustiva en tal espacio solución sería impráctica. Por lo tanto, las diversas soluciones propuestas utilizan heurísticas para lograr un resultado en corto tiempo que, aunque no óptimo, es suficientemente satisfactorio. Las heurísticas utilizadas pueden ser tanto de bajo nivel, como metaheurísticas e hiperheurísticas.

Entre los métodos utilizados para resolver el SSP encontramos el algoritmo húngaro [7], hiperheurísticas generadoras de heurísticas de bajo nivel [6], tiempo real [8], y el algoritmo de búsqueda iterativa hacia adelante [9, 3].

### *2.1. El SSP y la Oferta Académica*

El SPP está relacionado con la construcción de la oferta académica (*Course Timetabling Problem, CTP*). Una solución satisfactoria del CTP, permite una mejor resultado del SSP. Algunas iniciativas resuelven SSP posteriormente a CTP [7, 6, 10], mientras que otras tratan de resolver SSP y CTP simultáneamente [9, 11, 12].

## 3. Facultad de Ingeniería UC

La Facultad de Ingeniería de la Universidad de Carabobo (FIUC), objeto de estudio en este trabajo, consta de 6 escuelas y un departamento de estudios básicos. Los períodos académicos

son semestrales y se rige por un sistema abierto de prelaciones, donde cada carrera tiene una duración nominal de 10 semestres. En los primeros semestres (1 al 4) se ofrecen materias comunes a más de una carrera, y a partir del semestre 5, se ofrecen las materias obligatorias de la escuelas. En los semestres finales (8, 9, 10) se cursan las materias electivas y el proyecto final de grado.

La matriculación promedio de la FIUC en recientes semestres ha sido de 9432 alumnos [13], lo cual la hace una facultad de gran matriculación. Por tal razón, las materias básicas y obligatorias tienen típicamente dos o mas secciones.

Previo al proceso de inscripción, la Dirección de Asuntos Estudiantiles (DAE) junto con las Comisiones de Horario de cada escuela, elaboran la oferta académica, consistente en un promedio de 270 asignaturas y 1239 secciones [13]. Las asignaturas de los primeros semestres están divididas en decenas de secciones. A medida que se progresa en la carrera, el número de secciones decrece, hasta llegar a secciones únicas en los últimos semestres.

### *3.1. Proceso de Inscripción*

Durante el período de inscripciones (generalmente de 3 a 5 días), los estudiantes proceden a solicitar materias en el sistema Web de la DAE, donde constata el cumplimiento de prelaciones. Los estudiantes pueden hacer su solicitud de inscripción en cualquier momento del período, dado que no se toma en cuenta el momento en que son registradas en el sistema.

Una vez cerrado el período, la oferta académica y las solicitudes son preparadas para servir de insumo al programa de asignación.

### *3.2. Objetivos y Restricciones*

Los requerimientos del proceso de asignación los subdividimos en objetivos generales y restricciones.

*Objetivos Generales.* Son metas que deben ser cumplidas por la solución global, son exigidas por la institución y deben ser obligatoriamente cumplidas

**G**$_1$ Asignar horarios al 100 % de los estudiantes participantes en el proceso inscripción.

**G**$_2$ Uniformar el número de alumnos asignados por sección para cada materia.

*Restricciones.* Son aquellos requerimientos que deben ser cumplidos por cada horario solución. A continuación los requerimientos ordenados de mayor a menor según su prioridad

**R**$_1$ No permitir coincidencias de hora entre dos materias.

**R**$_2$ Asignar sección a todas las materias solicitadas por el estudiante.

**R**$_3$ Respetar una hora libre para el almuerzo en un intervalo $(+1, -1)$ alrededor de las 12m.

**R**$_4$ Número de alumnos $N_{i,j}$ asignados por sección $s_j$ de cada materia $m_i$ no debe exceder cupo máximo $c_i$.

**R**$_5$ Cumplimiento de alguna de las 3 opciones de horario.

Estas restricciones serán cumplidas en lo posible, serán sucesivamente levantadas para llegar a la generación de todos los horarios. Al levantar la restricción $R_1$, en aquellos pocos casos donde esto no sea posible cumplirla, sólo se permitirá una hora de coincidencia por alumno.

## 4. Solución al SSP en la FIUC

La solución para el problema de asignación planteado, se fundamentó en permitir a los estudiantes seleccionar 3 opciones de horario. Más que proveer alternativas a cada una de las materias [9], se obliga al estudiante a construir 3 horarios completos. Estos horarios tienen un orden de preferencia del 1 al 3, no deben tener coincidencias, y deben ser diferentes entre sí en al menos una materia/sección. Para poder implementar esta estrategia, la oferta académica debe estar elaborada al momento del proceso de inscripción, por lo que esta estrategia de solución al SSP es posterior a la solución del CTP.

Se diseñó un algoritmo de asignación que procese las 3 opciones de horario y asigne el 100 % de los alumnos a secciones, maximizando la satisfacción de una de las opciones o en su defecto, introduciendo el mínimo posible de cambios de sección.

El algoritmo consiste en 5 fases sucesivas, cada una asignado el máximo de estudiantes posibles (estrategia voraz o *greedy*). Conforme se avanza cada fase, se va dificultando la asignación de los alumnos, dado que los cupos disponibles de la oferta se irán reduciendo, por lo que cada fase trabaja con la adecuación de los requerimientos de manera de cumplir con los objetivos globales.

### *4.1. Datos de Entrada*

*Oferta Académica (OA).* Conjunto de registros $(m, s, c)$, donde $m$ es el código de la materia, $s$ el número de sección y $c$ el cupo máximo permitido.

*Horarios (H).* Conjunto de registros $(m, s, d, b, a)$ donde $d$ es el día de las semana, $b$ es el bloque de hora, y $a$ el aula, que especifica cuándo y donde se impartirá la materia.

*Estudiantes (E).* Conjunto de registros ordenado $(id, I_a)$, de tamaño $|E| = N$, donde $id$ es la cédula de identificación del estudiante e $I_a$ el índice de asignación. El orden descendente de este conjunto es determinado por $I_a$, que es un valor en el rango $[0 - 1]$. $I_a$ es obtenido a partir del historial académico del estudiante, y favorece a los que tengan de mejor promedio de notas, mejor actividad académica, antigüedad en la carrera, y aspectos particulares de la solicitud [13].

*Solicitudes (S).* Conjunto de registros de tipo $(id, m, s_1, s_2, s_3)$, donde $id$ es la cédula de identificación del estudiante, $m$ es el código de la materia, y $s_1, s_2, s_3$ son las tres opciones deseadas de sección para dicha materia, ordenadas por prioridad.

### *4.2. Resultado*

A partir de los conjuntos $OA$, $E$, $H$ y $S$, el resultado del algoritmo será la asignación $A$, que es un conjunto registros del tipo $(id, m, s)$. Donde $id$ es la cédula de identificación del estudiante, $m$ es el código de la materia, y $s$ la sección asignada.

### *4.3. Fases del Algoritmo*

**AD** Asignación directa de alguna de las opciones, sin cambios de sección. Se rechazan los estudiantes a los cuales no se les puede cumplir ninguna de las opciones.

**AC** Asignación de alguna de las opciones, con cambios de sección, rechazando así la restricción $R_5$. Se rechazan los estudiantes que no cumplen con las demás restricciones.

**AR** Repetición de Fase AC con rechazo de restricciones $R_4, R_3, R_1$ en este orden, hasta llegar a la asignación del 100 % del universo de estudiantes.

**B** (Balanceo) Se efectúan cambios de sección por cada materia para uniformar número de alumnos por sección y cumplir en lo posible con $R_4$

**W** (Retiro) Se efectúa un retiro forzado de materias en aquellos casos que no cumplen $R_1$, cumpliendo así con esta restricción y rechazando la restricción $R_2$.

La Fases AD, AC, AR se encargan de efectuar la asignación de todos los alumnos, guiándose principalmente por la preferencia de las tres opciones seleccionadas. Una vez terminadas estas 3 fases, el universo de asignación es el 100 %. Las siguientes Fases B y W modifican la asignación preliminar obtenida hasta ese momento.

A continuación mostramos un breve resumen de cada una de las fases. Para mayor detalle de los algoritmos puede consultarse nuestro trabajo previo [13].

### *4.4. Fase AD*

La primera fase del proceso sigue una estrategia de asignación secuencial para cada uno de los estudiantes ordenados de mayor a menor según su índice de asignación $I_a$. Para cada estudiante se verifica si su opción 1, 2 ó 3 es viable, En caso afirmativo, le asigna el horario solicitado. En caso negativo, lo deja sin asignar para la próxima fase.

La Fase AD conduce a una asignación *todo o nada*, y la complejidad computacional es $O(N)$, donde de tamaño $|E| = N$.

### *4.5. Fase AC*

La Fase AC tiene una estrategia de asignación ordenada similarmente a la Fase AD, pero rechazando la restricción $R_5$ (cumplir las opciones

de horario). Es decir, se introducen cambios de secciones en las solicitud. Una función calcula el costo de un horario solución de acuerdo a:

$$F(a) = k_c C_c(a) + k_d C_d(a) \quad (1)$$

donde $F(a)$ es la función de costo de un horario solución $a$ para un estudiante, $C_c(a)$ es el costo por el número de cambios de sección respecto a la opción más próxima, y $C_d(a)$ es el costo de dispersión de los bloques horarios. Al seleccionar la solución con menor costo, se busca que se selecciones preferentemente aquellos horarios con menos cambios con respecto a las opciones solicitadas y en su defecto, aquellos que sean más compactos.

Dada una solicitud $s$, conformada por $k$ asignaturas, y para cada asignatura $m_i$ existen $ns_i$ secciones con disponibilidad, tenemos que el tamaño del espacio total solución será la productoria

$$N_h = ns_1 * ns_2 * ns_3 \ldots * ns_k.$$

Dado que la cantidad de soluciones crece según esta productoria, el espacio solución puede llegar a ser muy grande para materias con muchas secciones disponibles.

Para evitar un tiempo de ejecución exageradamente largo, la búsqueda del espacio solución es exhaustiva hasta un tamaño máximo predeterminado. Para mayores espacios se utilizará un recorrido no exhaustivo de acuerdo a un algoritmo de búsqueda y *backtracking*. En todos los casos, se establece un límite máximo de búsqueda de soluciones $N_h max$, a efectos de evitar que el algoritmo se detenga durante periodos muy largos explorando el árbol de soluciones.

La complejidad computacional de la Fase AC es $O(N \times N_h max)$, donde $N$ es el número de estudiantes y $N_h max$ es el número de horarios máximo del espacio solución recorrido.

### *4.6. Fase AR*

Una vez corridas las dos primeras fases, queda un conjunto de alumnos que no se les ha podido asignar sus horarios. Este residuo es producto de las imperfecciones de la oferta institucional, y de las limitaciones de los algoritmos utilizados.

La Fase AR consiste en correr nuevamente la Fase AC, pero levantando las restricciones $R_4$, $R_3$ y $R_1$, una a una, en el orden de menor a mayor prioridad. Primero se libera el límite de cupo en las secciones, luego se libera la hora de almuerzo y por último se permiten coincidencias en las secciones. El motivo de mantener a $R_2$ en esta etapa es que en la última Fase R, se retirarán materias asignadas, reinstaurando el requerimiento $R_1$ y liberando el $R_2$.

Al finalizar este paso, el número de estudiantes que quedan por asignar será cero.

### *4.7. Fase B*

La Fase B, es un balanceo de alumnos por materia. Por requerimiento de las diferentes cátedras académicas, cada profesor de una materia deberá tener en su sección el mismo número de alumnos que las demás secciones. Con esto se asegura una mejor utilización del espacio físico y equidad en la carga académica profesoral. En el caso de aulas de desigual capacidad, este requerimiento significa que cada sección se llenará al mismo porcentaje de capacidad de su aula asignada.

Dado que las Fases AD, AC, AR efectúan la asignación a partir de las solicitudes de cada estudiante, el resultado parcial generalmente presenta un desbalance de alumnos en las secciones de una misma materia. Unas secciones, por preferencia estudiantil, terminan con mayor alumnos que otras. Adicionalmente, dado que en Fase AR fueron relajadas las restricciones de cupo, algunas secciones terminan con mas estudiantes que el cupo máximo.

Para resolver estos problemas, se ejecuta la fase de balance de secciones. Mediante una heurística dinámica de migración entre las secciones de mayor ocupación a las de menor ocupación.

El balance de secciones consiste en encontrar la relación entre el total de cupos y el número de estudiantes en cada materia, y encontrar cuáles secciones están por encima o por debajo de dicha relación. Desde la sección con más ocupación se evalúan los posibles cambios de sección hacia las secciones de menor ocupación. De todos los

posibles cambios, se selecciona el que signifique mayor reducción de costo, de acuerdo a la ecuación:

$$\Delta F(a', a) = F(a') - F(a)$$

donde $F$ es la función de costo de la Ec. (1), $a'$ es la nueva asignación (con el cambio de sección), $a$ es la asignación actual del alumno.

Una vez efectuado el cambio, deben reevaluarse las relaciones de ocupación de las secciones de la materia. El proceso se repite mientras la desviación estándar tienda a cero.

El balanceo de secciones permite el cumplimento de $G_2$, pero va obviamente en detrimento de $R_5$. Hemos verificado mediante análisis empíricos que esta estrategia de balance resulta, en muchos casos, en una reducción del costo de la asignación ($\Delta F < 0$). Esto debido a la compactación de los bloques horarios de la nueva asignación. Entonces, se cumple un intercambio entre una opción solicitada por un horario de menor dispersión.

La complejidad computacional de la Fase B es $O(M \times N)$, donde $M$ es el número de materias y $N$ es el número de alumnos.

### *4.8. Fase W*

Finalmente se aplica la Fase W, que retira materias a los horarios que incumplen la restriccion $R_1$, es decir, aquellas que generan coincidencias. Mediante este procedimiento, se levanta $R_2$ y se restituye el cumplimiento de $R_1$. Esta fase presenta el riesgo de dejar estudiantes sin horario (incumpliendo así con $G_1$). Sin embargo, en los ejemplos estudiados, esto no ha sido el caso.

Para aquellos casos en que un estudiante se quede sin un horario por aplicación de esta Fase, el software activa una señalización apropiada, indicando así la eventualidad.

### *4.9. Resumen de Heurísticas*

La Tabla 1 muestra un resumen de las diferentes heurísticas utilizadas. Clasificamos las heurísticas según si tipo: estáticas y dinámicas. Las estáticas mantienen fijos los criterios de aplicación si importar el cambio en los datos de entrada. Las heurísticas dinámicas son aquellas que exigen constante re-evaluación de las condiciones de aplicación. Por tal motivo, las fases que aplican heurísticas dinámicas requerirán mayor nivel de cómputo para el mismo tamaño de problema de entrada.

### *4.10. Cambios en la Oferta Académica*

Debido a imperfecciones de la oferta, y para considerar aspectos inesperados de las solicitudes estudiantiles, comúnmente es necesario hacer cambios en las secciones y sus horarios luego de cerrado el proceso de inscripción. En ocasiones hay que abrir o cerrar secciones debido a los niveles reales de demanda. En otras ocasiones la ausencia inesperada de un profesor requiere el cierre de una sección.

Mediante el programa desarrollado es posible hacer una primera corrida y con el resultado obtenido evaluar los aspectos de demanda real. A partir del análisis de la demanda y el resultado de la asignación preliminar, pueden hacerse cambios pertinentes en la oferta, para posteriormente volver a efectuar la asignación. En el caso de apertura nueva de secciones, el programa de asignación llenará las nuevas secciones en las fases AC, AR o B. En el caso de cierre de secciones de última hora, el programa hará los retiros pertinentes de sección en la fases AC, AR o B.

El ciclo de modificación de la oferta y asignación se repite hasta encontrar una solución satisfactoria para la institución.

### *4.11. Implementación*

Se desarrolló el programa de asignación para la DAE, que implementa los conceptos descritos en este trabajo. El mismo fue desarrollado lenguaje Python 2.7.6. El programa acepta como entrada los archivos formato CSV generados por el sistema de inscripción y genera el archivo de asignación en formato CSV.

La versión preliminar del programa de asignación fué utilizado por DAE en los semestres 2013–f, 2013–u y 2014–1. Actualmente, la DAE utiliza la versión de producción de este software para las asignaciones de sus horarios.

Tabla 1: Resumen de heurísticas utilizadas

| Fase | Heurísticas | Tipo |
|---|---|---|
| AD | Índice de asignación $I_a$ | Estática |
| | Orden de las opciones de horario | Estática |
| AC, AR | Indice de asignación $I_a$ | Estática |
| | Orden de las opciones de horario | Estática |
| | Búsqueda limitada con backtracking | Dinámica |
| B | Menor $\Delta F$ | Dinámica |
| W | Eliminar materia con mas coincidencias | Dinámica |

## 5. Experimentación

Se tomó una copia la versión definitiva del programa, y se ejecutó en un computador con 4 procesadores AMD Opteron 2.0 GHz de 8 núcleos c/u, 256Gb Memoria RAM, sistema operativo Linux Centos 6. El diseño del programa limita su ejecución a estrictamente secuencial (un solo hilo de ejecución).

Se tomaron 3 conjuntos de datos, provenientes de los semestres 2013–f, 2013–u y 2014–1. Para cada uno de ellos, se generó un índice $I_a$, el programa de asignación fue ejecutado y los resultados fueron evaluados. Consideramos el resultado más conservador de los 3 períodos para la discusión y conclusiones.

## 6. Resultados

La Tabla 2 muestra el cumplimiento de los objetivos y restricciones para cada uno de los datos de entrada, una vez terminado el proceso de asignación. Se observa cómo, para cada uno de los semestres, se verifica empíricamente que el objetivo $G_1$ se cumplió en 100 %, es decir, todos los estudiantes reciben un horario, sin excepción. La asignación sin coincidencias se cumple en un 98,8 %, y horarios con todas las materias asignadas para al menos un 98,8 % de los alumnos.

Se observa también que se logró asignar alguna de las opciones, por lo menos para el 44 % del alumnado. Éste es el indicador que elegimos como grado de satisfacción absoluta en el cumplimiento de $R_5$. En las últimas filas de la tabla, se muestra el porcentaje de satisfacción absoluta discriminado por opción. La primera opción es la mas frecuentemente asignada de las tres opciones.

Mientras que el grado de satisfacción absoluta resultó ser del 44 %, es importante conocer cual es el grado de satisfacción parcial. Es decir, si para el 44 % pudimos satisfacer todas las materias de una opción, necesitamos conocer para los demás casos en que introdujeron cambios de sección, cuántas de las secciones del horario fueron satisfechas. Para ello, comparamos las secciones obtenidas con las secciones seleccionadas en las 3 opciones. La comparación la hacemos en forma porcentual, dado que algunos alumnos piden más materias que otros. Finalmente, contamos los horarios según el porcentaje de secciones satisfechas.

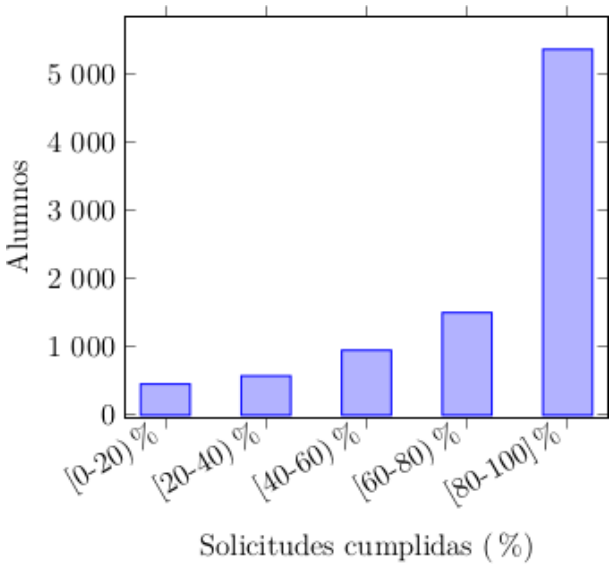


Figura 1: Cumplimiento parcial de solicitudes (2014-1). Cuántas materias dentro de una misma solicitud pudieron asignarse de acuerdo a alguna de las 3 secciones elegidas por el alumno. Se indica el porcentaje de materias respecto al total de la solicitud.

La Figura 1 muestra el histograma de horarios

Tabla 2: Cumplimiento de Objetivos y Restricciones

| | | 2013–f | | 2013–u | | 2014–1 | |
|---|---|---|---|---|---|---|---|
| Asignación | $G_1$ | 10019 | % | 9412 | % | 8864 | % |
| Sin coincidencias | $R_1$ | 9929 | 99,1 | 9358 | 99,4 | 8818 | 99,5 |
| Todas las materias | $R_2$ | 9979 | 99,1 | 9342 | 99,4 | 8818 | 98,5 |
| Cualquier opción | $R_5$ | 4868 | 48,6 | 4179 | 44,4 | 4125 | 46,5 |
| Opción 1 | | 3095 | 30,9 | 2699 | 28,7 | 2604 | 29,4 |
| Opción 2 | | 943 | 9,4 | 814 | 8,6 | 743 | 8,4 |
| Opción 3 | | 830 | 8,3 | 666 | 7,1 | 778 | 8,8 |

de acuerdo al porcentaje de satisfacción para el semestre 2014-1. Se observa la tendencia clara de mayor frecuencia hacia mayores grados de satisfacción. Se observa igualmente que en la clase [80-100] % la frecuencia es de 5372 horarios, y en la clase [60-80) % es 1506 horarios.

Si definimos *asignación satisfactoria* aquel grado de calidad de solución donde se le asigna a un alumno por lo menos el 60 % de su solicitud, entonces podemos concluir que sumando las dos clases más altas, obtenemos 6878 alumnos, que representa un (77 %) del total de horarios.

Se repitieron los cálculos para los semestres 2013-f y 2013-u y se obtuvo un mínimo de (74 %) de asignaciones satisfactorias (2013-f).

### *6.1. Tiempo de Ejecución*

La Tabla 3 muestra el tiempo de ejecución de las distintas fases del proceso de asignación. Sólo se consideraron las Fases AD, AC y B, dado que el tiempo de ejecución para las fases AR y W resultó ser despreciable.

Tabla 3: Tiempo de Ejecución

| | 2013-f | | 2013-u | | 2014-1 | |
|---|---|---|---|---|---|---|
| Fase | Tiempo (s) | % | Tiempo (s) | % | Tiempo (s) | % |
| AD | 16 | 2 | 16 | 2 | 14 | 2 |
| AC | 573 | 73 | 459 | 65 | 543 | 73 |
| B | 201 | 25 | 236 | 33 | 189 | 25 |
| Total | 790 | | 711 | | 746 | |

El tiempo máximo total fue de 790 segundos (poco más de 13 minutos). La fases de mayor tiempo computacional son la Fase AC (72 %), seguida por la Fase B (25 %).

El tiempo total obtenido lo consideramos aceptable para la aplicación, dado que el programa sólo corre en una sesión por semestre y con suficientemente tiempo reservado para obtener la solución (7 horas de jornada laboral). Sin embargo, existe un margen para mejorar estos tiempos de ejecución, utilizando un lenguaje más orientado al cómputo de alto rendimiento y con el empleo de técnicas de multiprocesamiento.

#### *6.1.1. Porcentajes de asignación*

Se contabilizaron las asignaciones por cada fase AD, AC, AR y los resultados se muestran en la Tabla 4. Se observa que por lo menos el 53 % de los estudiantes son asignados en la primera fase (2014-1). En atención a los tiempos de ejecución mostrados en la Tabla 3, se concluye que gracias a la heurística de opciones seleccionadas por los alumnos, es posible obtener una gran parte del resultado en forma directa y en poco tiempo de ejecución (2 % del tiempo total). Esto reviste un mérito significativo para la solución al SSP, dado que en ese momento de ejecución las secciones se encuentran con baja ocupación y una exploración del espacio-solución llevaría mucho tiempo de búsqueda. Esto queda confirmado en la fase AC, que para el período 2014-1 tarda 38 veces más que la fase AD a pesar de trabajar con sólo el 47 % del universo de alumnos.

## 7. Conclusiones

Se diseñó un algoritmo de asignación de horarios estudiantiles para la Facultad de Ingeniería UC, que procesa 3 opciones de horario construidas y solicitadas por los mismos alumnos. Las opciones de horario se construyen a partir de una oferta académica completa con secciones y horarios disponible al momento de la inscripción.

Tabla 4: Porcentajes de Asignación por Fase

| | 2013-f | | 2013-u | | 2014-1 | |
|---|---|---|---|---|---|---|
| Fase | Asignación | % | Asignación | % | Asignación | % |
| AD | 5519 | 55 | 5295 | 56 | 4717 | 53 |
| AC | 4065 | 41 | 3686 | 39 | 3404 | 38 |
| AR | 435 | 4 | 431 | 5 | 743 | 8 |

El algoritmo está compuesto por una secuencia de fases que aplican las heuristicas resumidas en la Tabla 1 que tratan de asignar una de las opciones solicitadas, o en su defecto un horario aproximado con el menor cambio de secciones posible. El algoritmo efectúa una fase balanceo para equiparar los estudiantes asignados a cada sección de una misma materia.

El programa en versión final, desarrollado en lenguaje Python, es utilizado por la Dirección de Asuntos Estudiantiles desde el 2013.

En la experimentación se verificó que los resultado obtenidos presentan un 100 % de efectividad de asignación, con un 98,8 % de horarios sin coincidencias, y un 98,2 % de horarios con todas las materias.

En cuanto a la calidad del resultado, se observó que 44 % de los horarios generados pertecen a una de las 3 opciones seleccionadas por los alumnos, mientras que el porcentaje de asignaciones satisfactorias fue del 77 %.

El programa se ejecuta en poco más de 13 minutos en el equipo utilizado para las pruebas. La fase de asignación directa de opciones (AD) resuelve el 53 % del problema en sólo 14 segundos (2 % del tiempo total), con lo cual concluimos que las opciones dadas por los estudiantes contribuyen grandemente a la solución del problema en forma eficiente. La Fase AC se llevó el 72 % del tiempo y la Fase B el 25 %.

El tiempo de ejecución permite la adecuación de la oferta académica a la demanda real mediante el ciclo iterativo diseño de oferta-asignación, el cual se corre varias veces hasta obtener oferta y asignación final que satisfagan los requerimientos institucionales.

Los resultados obtenidos apuntan a una solución satisfactoria a un problema real de asignación de horarios, que permite la utilización eficiente de los recursos humanos y físicos. En tal sentido, se observa cómo es posible cumplir objetivos dispares mediante la aplicación de algoritmos aproximados fundamentados en principios sencillos de operación. Este grado de satisfacción presentado en este trabajo debe ser tomado en cuenta para cualquier mejora utilizando otros principios algorítmicos de diseño.

## 8. Recomendaciones

Proponemos mejoras en las heurísticas aplicadas por el algoritmo que resulten en aún mejores resultados, es decir, que se incrementen los índices de satisfacción absoluta y/o de asignaciones satisfactorias. Es de especial interés implementar conceptos de meta-optimización y efectuar comparaciones entre resultados tal como ha sido establecido en otros tipos de problemas [14].

También proponemos observar cómo varía la calidad de la respuesta cambiando la oferta académica, de manera que permita establecer estrategias de identificación de errores y factores críticos en la oferta presentada por las cátedras.

Demostrada la validez de los conceptos presentados en este trabajo, y el uso continuado de la versión en Python, se recomienda la implementación de la aplicación utilizando lenguajes de programación compilados como C, Fortran o el reciente lenguaje dinámico de alto nivel Julia [15], y aplicar técnicas de cómputo paralelo para las fases más lentas del algoritmo (Fase AC y Fase B) de manera de reducir el tiempo de ejecución. Una forma interesante de ejecución en paralelo sería explotar el cómputo en cascada realizado por el algoritmo propuesto, mediante bibliotecas especializadas en cómputo *pipeline* con estructuras matriciales [16].

## Referencias


[1] Michael W Carter and Gilbert Laporte. Recent developments in practical course timetabling. In *International Conference on the Practice and Theory of Automated Timetabling*, pages 3–19. Springer, 1997.

[2] Edmund Kieran Burke and Sanja Petrovic. Recent research directions in automated timetabling. *European Journal of Operational Research*, 140(2):266–280, 2002.

[3] T. MÃ¼ller. *Constraint-based Timetabling*. PhD thesis, Faculty of Mathematics and Physics. Charles University in Prague, 2005.

[4] Ronen Feldman and Martin Charles Golumbic. Optimization algorithms for student scheduling via constraint satisfiability. *The Computer Journal*, 33(4):356–364, 1990.

[5] Eddie Cheng, Serge Kruk, and Marc Lipman. Flow formulations for the student scheduling problem. *Practice and Theory of Automated Timetabling IV*, pages 299–309, 2003.

[6] Nelishia Pillay. A study of the practical and tutorial scheduling problem. In *10th International Conference of the Practice and Theory of Automated Timetabling PATAT*, August 2014.

[7] Michael W. Carter. A comprehensive course timetabling and student scheduling system at the University of Waterloo. In *Practice and theory of automated timetabling III*, pages 64–82. Springer, 2001.

[8] Keith Murray and Tomáš Müller. Real-time student sectioning. In *Proceedings of the 3rd Multidisciplinary International Conference on Scheduling: Theory and Application (MISTA 2007), Paris, France*, pages 598–600, 2007.

[9] Tomáš Müller and Keith Murray. Comprehensive approach to student sectioning. *Annals of Operations Research*, 181(1):249–269, 2010.

[10] David Schindl. Student sectioning for minimizing potential conflicts on multi-section courses. In *11th International Confenference on Practice and Theory of Automated Timetabling (PATAT-2016)*, August 23–26 2016.

[11] Gary Lewandowski. Simultaneous construction of student schedules and timetable, 1996.

[12] Jeffrey H Kingston. Integrated student sectioning. In *10th International Confenference on Practice and Theory of Automated Timetabling (PATAT-2014)*, pages 489–492, 2014.

[13] Demetrio Rey, Luis Llave, Enrique Flores, Víctor Barrios, Braulio De Abreu, and José L. Nazar. Asignación de horarios para los alumnos de la Facultad de Ingeniería UC. Technical report, Facultad de Ingeniería Universidad de Carabobo, 2015. http://dx.doi.org/10.13140/RG.2.1.4995.2486.

[14] Habib Youssef, Sadiq M Sait, and Hakim Adiche. Evolutionary algorithms, simulated annealing and tabu search: a comparative study. *Engineering Applications of Artificial Intelligence*, 14(2):167–181, 2001.

[15] Jeff Bezanson, Alan Edelman, Stefan Karpinski, and Viral B. Shah. Julia: A fresh approach to numerical computing. *CoRR*, abs/1411.1607, 2014.

[16] Demetrio Rey and James Canning. Streams: una librería de habilitación de paralelismo mixto de tareas en cascada y datos para el lenguaje de programación zpl. *Revista Ingeniería UC*, 14(2):70–78, 2007.