Un método para asignar estudiantes a asignaturas-grupo

La finalidad de este artículo es presentar una versión mejorada, desarrollada por el autor del método Simplex Stopped propuesto por G. L. Thompson.

El artículo se inicia con un marco general del proceso de registro y sus alternativas. A continuación se presenta el modelo factible para entrar al de holgura minimax y su método de cómputo, algoritmo y heurística incorporada.

Finalmente se presentan los resultados computacionales, los cuales permiten obtener las conclusiones al respecto.

LUIS GERARDO ASTAIZA A.
Ingeniero Mecánico, MIS
Profesor Asociado
Universidad Nacional

Registro en el sentido más amplio normalmente incluye el desarrollo de un curriculum equilibrado, con clases igualmente distribuidas durante el día y la semana, consejería, recolección de datos personales y asignación de los estudiantes a clases y grupos, y matrícula. Las técnicas utilizadas varían de institución a institución.

A pesar de los diferentes procedimientos utilizados en el registro de estudiantes, todos comparten elementos comunes y objetivos similares. Entre los objetivos están:

- Proveer a cada estudiante con un horario para sus asignaturas.
- Mantener la magnitud de los cursos dentro de contornos establecidos.
- Minimizar el tiempo y el esfuerzo requerido por parte de administradores, profesores y estudiantes en el proceso de registro.
- Utilizar las facilidades docentes tan eficientemente como sea posible.
- Preservar la mayor flexibilidad en la programación, de manera que permita ajustes manuales, registro posterior, modificaciones de cursos, etc.
- Recoger el pago de los derechos de matrícula según las políticas establecidas por la institución.

ALTERNATIVAS A LOS SISTEMAS DE REGISTRO

Pre-registro

Las técnicas de pre-registro varían ampliamente pero dentro de lo normal implican la selección de cursos y pago parcial o completo de la matrícula durante el período académico precedente. La metodología incluye sistemas tan simples como recolección de la forma o tarjeta maestra, orientación del profesor consejero y regreso en fecha específica al pago de matrícula y recolección de tarjetas de cursos. Los sistemas de pre-registro generalmente hacen un llamado a los estudiantes, bien sea en orden alfabético o numérico (número de identificación estudiantil).

Sin importar el sistema utilizado, generalmente es imposible pre-registrar a todos los estudiantes. Así es habitual realizar un registro justamente antes de iniciar clases. Este proceso, generalmente, incluye ajustes de horario para estudiantes pre-registrados que han cambiado sus objetivos académicos, o que han sido admitidos en fecha posterior al registro, y estudiantes que no realizaron el pre-registro. La selección del grupo para las asignaturas del pre-registro también varía; la mayoría de las instituciones efectúa esta selección al cierre del período de pre-registro; otras mantienen la información de entrada, para realizar una sola vez la selección de grupos precisamente un poco antes de iniciar clases.

REGISTRO MASIVO

El registro masivo o tipo coliseo se realiza en varios días inmediatamente anteriores al primer día de clases. Normalmente implica la distribución de material, consejería, pago de matrícula y recolección de todas las tarjetas o formas asociadas con el proceso de registro. En algunas instituciones, representantes del profesorado de cada Departamento realizan la asignación de clases, con selección manual del grupo mediante el proceso de recolección de tarjetas por parte del estudiante. En otras instituciones un sistema basado en el computador realiza esta tarea de acuerdo con la solicitud del estudiante. El pago de matrícula se lleva a cabo en otro lugar en fecha posterior. Instituciones con adecuado equipo de computación utilizan el aproximamiento de acceso directo de la información por medio de terminales.

Instrucciones de registro y selección de grupo

Para un adecuado funcionamiento del sistema es necesario elaborar adecuadamente el boletín de asignaturas (horario maestro) y las instrucciones de registro, tanto para estudiantes como profesores, así como planear en avance la producción, ya sea manual o por computador, del material requerido en el proceso.

El paso siguiente es realizado por los estudiantes. El procedimiento se inicia con la recolección del material de registro-instrucciones, formas o tarjetas perforadas que incluyen una tarjeta maestra y una tarjeta de matrícula. Posteriormente viene la selección de asignaturas y grupos.

Precisamente para esta etapa muchas instituciones emplean el computador basado en un conjunto complejo de prioridades. En estos casos, el estudiante generalmente tiene la oportunidad de indicar grupos preferidos y tiempo para realizar otras actividades. Aún más, los programas de computador aceptan solicitudes alternas en caso de que uno o más de los cursos solicitados no estén disponibles. Un conjunto típico de prioridades en la selección de grupo es como sigue:

- Tiempo libre para trabajar o realizar deportes.
- Cursos de grupo único.
- Grupos preferidos de cursos con grupos múltiples cuando se indique.
- Otras solicitudes para cursos de grupos múltiples.
- Asignación de laboratorios.
- Cursos de reemplazo, si uno o más de la solicitud original son negados.

Con agrupación por computador, todas las solicitudes de cursos son procesadas en esta forma hasta que cada estudiante tenga un horario de clases.

EL MODELO FACTIBLE

Mucho esfuerzo se ha gastado en los últimos años en programar en computadores soluciones heurísticas del problema de asignación de estudiantes a asignatura-grupo. Los algoritmos de asignación de estudiantes a asignatura-grupo pueden clasificarse con base en estudiante por estudiante o clase por clase.

Precisamente en este artículo se presenta un método basado en el procedimiento estudiante por estudiante, el cual combina ideas heurísticas y algorítimicas. En primer lugar, usa ideas heurísticas para decidir el orden a considerar para los estudiantes y para formular el modelo matemático que se resuelva. En segundo lugar, emplea las ideas algorítimicas al resolver un problema de programación entera para efectuar la asignación de estudiantes a clases y grupos. El problema de programación entera tiene como objetivo minimizar la suma de las holguras de los grupos considerados. El método propuesto tiene las siguientes características.

- Ofrecimiento de cursos, grupos, horarios y cupos.
- Solicitud del estudiante de acuerdo con el ofrecimiento e incluyendo prioridades.
- Incorporación de reglas heurísticas en la formulación del modelo.
- Aplicación del método Simplex Stopped para asignar el horario a un estudiante, en particular.

A continuación se presenta el modelo que imita la técnica de asignación antes de considerar el modelo mejorado.

Suponga que un estudiante ha indicado las asignaturas que él desea tomar, más sus horas libres. Para definir el modelo del problema de asignación, sea **N** el número del curso, **NG** el número del grupo y **t** el horario. Entonces definimos la variable

$$X(N, NG, t) = \begin{cases} 1 & \text{si el estudiante toma el curso } \mathbf{N}\text{, en el grupo } \mathbf{NG}\text{, con horario } \mathbf{t}. \\ 0 & \text{si no la toma.} \end{cases} \qquad (1)$$

Estas variables presentan las siguientes restricciones:

$$\sum_{NG,\, t} X(N, NG, t) = 1$$

para cada curso N solicitado (2)

Esta restricción garantiza que el estudiante quede registrado solamente en un grupo del curso **N**. Para garantizar que no tenga conflictos en los horarios, se requiere que:

$$\sum_{N,\, NG} X(N, NG, t) \leq 1 \text{ para cada } t \qquad (3)$$

Restricciones de la forma

$$\sum_{N,\, NG} X(N, NG, t_i) = 0$$

para cada $t_i = 1, 2, \dots, K$ (4)

son también agregadas donde **K** es el número de horas libres y **ti** es la i^a hora libre. Adicionalmente si el estudiante solicita un curso de grupo único, se debe agregar una restricción de la forma (4) para ese curso. Esto es conveniente para los últimos períodos académicos de un programa (7º a 10º) semestres).

La variable **B(N, NG, t)** es definida con el número inicial de cupos en el curso **N**, grupo **NG** y horario **t**. Si el estudiante se asigna a ese grupo, el número se reduce en 1. En orden a cerrar un grupo donde no hay más cupo la restricción

$$X(N, NG, t) \leq B(N, NG, t) \qquad (5)$$

es agregada. Cuando **B(N, NG, t)** = 0, esta restricción da $X(N, NG,t) \leq 0$ tal que no permita asignar más estudiantes en esta asignatura-grupo.

Las restricciones (1) — (4) junto con $0 \leq X(N, NG, t) \leq 1$ definen un conjunto convexo de vectores. Un punto en el conjunto que haga todos los valores 0 ó 1 para las variables **X(N, NG, t)** es buscado. El método computacional utilizado es el método Simplex Stopped (con algunas variables de valor fijo) descrito en (5).

De hecho es un problema de programación entera, que requiere una función objetivo. Exactamente, cuál función objetivo se seleccione no tiene importancia para el modelo, pero la función

$$\text{Minimizar } Z = \sum_{N, NG, t} X(N, NG, t) \qquad (6)$$

es seleccionada. Se conoce de antemano que para cualquier solución factible el valor de la función objetivo es **p**, donde **p** es el número de cursos que el estudiante desea cursar. Por tanto la función objetivo no elimina la selección de cualquier solución entera factible.

El modelo definido por (1) — (6) se denomina el "modelo factible". Debido a la forma de la función objetivo, el método S^2 (Simplex Stopped) seleccionará algún horario factible y lo asignará al estudiante. Esto es similar de muchas maneras a otros métodos de asignación manual, excepto que los encargados del proceso pueden hacer el intento para equilibrar los grupos, tal que los "más deseables", no se cierren demasiado temprano en el proceso, dejando únicamente los "menos deseables" para el final.

Exactamente cuál horario seleccionará el método S^2 depende ampliamente del orden en el cual las restricciones para los varios grupos son presentados en el problema. Si, para el caso, los grupos se presentan en el mismo orden, existirá la tendencia para asingar a los grupos listados al comienzo, luego aquellos listados posteriormente, etc.

Debido a estas objeciones al modelo factible, el modelo de holgura minimax fue desarrollado.

MODELO DE HOLGURA MINIMAX

El problema de cualquier método de asignación de horarios es equilibrar los deseos conflictivos del estudiante y el registrador (como representante de la administración de la institución). A la mayoría de los estudiantes le gustaría tener las clases entre 10 a.m. y 4 p.m. y no tener clases los sábados. Naturalmente tal asignación de horarios no puede lograrse para todos los estudiantes simultáneamente. Sin embargo, existe la observación empírica de muchos estudiantes de que, con las técnicas de asignación actuales, la mayoría de sus requerimientos puede cumplirse si ellos se registran más temprano que tarde. Así que una buena técnica debe tratar a los estudiantes sobre una base igual hasta dónde sea posible, sin importar el momento en que les corresponda su asignación.

El objetivo principal del registrador es la elaboración de un horario factible para la institución que tenga la propiedad de que cada estudiante pueda tomar los cursos necesarios sin conflicto en sus horarios, y que no se asignen estudiantes a asignaturas-grupo donde no hay cupo. Además de obtener el objetivo principal, al registrador le gustaría tener flexibilidad, uso eficiente de los salones, etc.

El modelo factible puede lograr estas metas si se considera conveniente el orden de las restricciones, o alternativamente si el problema se resuelve en varias corridas de computador. El método propuesto por G. L. Thompson (4) mejorado en este artículo podrá obtener simultáneamente el objetivo principal de un horario factible, más el objetivo secundario de retener tanta flexibilidad como sea posible para registro extemporáneo y modificaciones.

El modelo de holgura minimax utiliza las expresiones (1) — (4) (pero (5) y (6) no). Nuevas variables X (N) son agregadas, una para cada curso. Entonces las restricciones

$$X(N) \geq B(N, NG, t) - X(N, NG, t) \quad \text{para cada } NG, t \qquad (7)$$

implica que **X(N)** ya sea mayor que (a) **B(N, NG, t)** o (b) **B(N, NG, t) - 1** dependiendo si **X(N, NG, t)** es cero o uno. En otras palabras, **X(N)** es mayor o igual a la holgura máxima de cualquier grupo del curso **N**, donde "holgura" significa el número de cupos en un grupo dado. Luego la función objetiva propuesta es

$$\text{MIN } X = \sum_{N} X(N) \qquad (8)$$

lo cual significa que el objetivo es minimizar la suma de las holguras máximas en todos los cursos. Luego, al seleccionar los grupos para satisfacer la solicitud de un estudiante, el método Simplex tratará de escogerlos con holgura máxima, si es factible hacerlo así.

Además, el método Simplex tendrá incorporada la regla de que si varios vectores pueden entrar a la base, aquél con holgura máxima será seleccionado. Esta regla de prioridad trata de seleccionar la asignatura-grupo con holgura máxima como primer ensayo para asignar el grupo para esa asignatura. Esta regla se considera como parte del modelo de holgura minimax.

Ecuaciones (1) — (4), (7) y (8) junto con la regla de la holgura más grande, conforman el modelo de holgura minimax. Además, puesto que las variables deben tener valores enteros, el modelo es un modelo de programación lineal en enteros.

A continuación presentamos un ejemplo simple del modelo:

Supongamos que un estudiante desea hacer tres cursos, matemáticas, física e historia, cada uno de los cuales tiene los encuentros lunes, miércoles y viernes. Asumiendo que los horarios y cupos son los dados en la Tabla 1.

TABLA 1.

Curso	Matemáticas			Física		Historia	
Horario	9	10	11	10	11	9	11
Cupos	10	10	9	6	7	5	5

Esto significa que matemáticas de las 9 tiene 10 cupos, física de las 10 tiene 6 cupos, etc.

Para efectos de la formulación del problema de programación lineal en enteros, sean **M**, **F** y **H** los nombres de las variables correspondientes a los tres cursos y los subíndices los horarios. Así F_{10} es física a las 10, H_9 es historia a las 9, etc.

Luego el modelo de holgura minimax consta de las expresiones:

$$M_9 + M_{10} + M_{11} = 1 \quad (9)$$
$$F_{10} + F_{11} = 1 \quad (10)$$
$$H_9 + H_{11} = 1 \quad (11)$$
$$M_9 + H_9 \leq 1 \quad (12)$$
$$M_{10} + F_{10} \leq 1 \quad (13)$$
$$M_{11} + F_{11} + H_{11} \leq 1 \quad (14)$$
$$X_M \geq 10 - M_9 \quad (15)$$
$$X_M \geq 10 - M_{10} \quad (16)$$
$$X_M \geq 9 - M_{11} \quad (17)$$
$$X_F \geq 6 - F_{10} \quad (18)$$
$$X_F \geq 7 - F_{11} \quad (19)$$
$$X_H \geq 5 - H_9 \quad (20)$$
$$X_H \geq 5 - H_{11} \quad (21)$$
$$X_M + X_F + X_H \leq X \quad (22)$$
$$\text{Minimizar } X \quad (23)$$

Las expresiones (9), (10) y (11) corresponden a (2); (12), (13) y (14) a (3); y (15) — (21) corresponden a (1). Las expresiones (22) y (23) requieren alguna explicación. En lugar de utilizar la función objetivo en la forma (2), la restricción (22) es considerada, la cual implica una nueva variable **X**, que se desea minimizar en (23). La razón para hacerlo en esta forma es tener el problema en forma preparada para que el método Simplex Stopped pueda trabajar inmediatamente. Es claro que (22) y (23) son equivalentes a (2).

Las ecuaciones (2), (10) y (11) son escritas como desigualdades en el modelo final con el propósito de mantener bajo el número de variables en el problema. Luego el problema resultante tendrá 11 variables y 17 restricciones. Note que los coeficientes de las variables son 1, -1 ó 0.

Podría pensarse que el problema de programación lineal tenga una solución dada por el método Simplex en enteros, pero desafortunadamente este no es el caso, dando lugar a la necesidad de utilizar algún algoritmo de programación de enteros.

En general, se puede demostrar que si un estudiante desea **K** cursos y el ith curso presenta NG_i grupos, con **T** horarios diferentes, entonces el problema de programación lineal resultante tendrá $N + K + 1$ variables y $T + 2K + N$ restricciones, donde:

$$N = \sum_{i=1}^{k} NG_i$$

Así que el tamaño típico de un problema de un estudiante solicitando 5 cursos podrá ser aproximadamente de 30 variables y 40 restricciones y tal vez más.

EL METODO DE COMPUTO

El algoritmo está basado en el trabajo desarrollado por Gerald L. Thompson en [4] y [5].

En orden a describir el problema, consideramos éste de programación lineal que hay que minimizar.

$$\text{Min } Z = CX$$

Sujeto a:

$$AX \leq b$$
$$X \geq 0 \quad (24)$$

donde **C** es $1 \times n$, **X** es $n \times 1$, **A** es $M \times n$ y **b** es $m \times 1$. Además las componentes de **A**, **b** y **C** son enteros. Escribiendo (24) en "forma apropiada" por agregar una variable más X_{n+1} y ampliando **A**, **b** y **C** como sigue:

$$A^* = \begin{pmatrix} A & 0 \\ 0 & -1 \end{pmatrix} \qquad b^* = \begin{pmatrix} b \\ 0 \end{pmatrix} \qquad C^* = (C, -1) \quad (25)$$

Las nuevas dimensiones de A^* son $(m + 1) \times (n + 1)$, b^* es $(m + 1)^* 1$ y C^* es $1 \times (n + 1)$. Entonces la "forma apropiada" de (24) es:

$$\text{MIN } X_{n+1}$$

Sujeto a:

$$A^* X^* \leq b^*$$
$$C^* X^* \leq 0 \quad (26)$$
$$X^* \geq 0$$

Note que $C^*X^* = CX - X_{n+1}$ tal que minimizar X_{n+1} en (26) es enteramente equivalente a minimizar **CX** en (24).

En el método Simplex Stopped algunas variables son mantenidas a valor fijo y el problema de programación lineal resultante es resuelto utilizando las restantes variables. Para establecer una notación para esto dejamos que:

$$X = W^{(K)} + S^{(K)} \quad (27)$$

Donde

$W^{(K)} = (W_1\ W_2 \ldots W_k\ 0 \ldots)$

$$S^{(K)} = (0, 0 \ldots 0, S_{k+1} \ldots S_{N+1}) \qquad (28)$$

Aquí las entradas en $S^{(K)}$ son mantenidas a valor constante para efectos de resolver un problema de programación lineal dado. Específicamente, suponga que $S^{(k)}$ es un vector de valor constante con componentes enteras no negativas. Entonces el problema de programación lineal que hay que resolver es:

$$\text{MIN} \quad W_K$$

Sujeto a:

$$\begin{aligned} A^*\ W^{(K)} &\leq b^* - A^*S^{(K)} \\ C^*\ W^{(K)} &\leq -C^*\ S^{(K)} \\ W^{(K)} &\geq 0 \end{aligned} \qquad (29)$$

Note que el problema (26) es el $P^{(N+1)}$.

Debe anotarse que problemas de la forma (29) presentan degeneración en el sentido usual de la programación y una de las técnicas standard de programación lineal debe utilizarse para evitar cicleo o poca convergencia en el método Simplex.

La notación $<X>$ se establece para 'El entero más pequeño $\geq$ X'.

Para el caso,

$<-1/2> = 0, <1/2> = 1, <-3> = -3$ y $<3> = 3$.

Después de una breve descripción informal del algoritmo Simplex Stopped, una proposición formal es dada. El método empieza por resolver $P^{(N+1)}$. Luego hace $S_{n+1} = <W_{N+1}>$ y resuelve $P^{(N)}$. Luego hace $S_N = <W_N>$ y resuelve $P(N-1)$ etc. Cada vez que se termina un problema $P^{(K)}$ y se fija el valor de la variable correspondiente con base en que $S_K = <X_K>$ se debe revisar que la restricción

$$C^*\ W^{(K)} + C^*\ S^{(K)} \leq 0 \qquad (30)$$

se cumpla. Si esta restricción se incumple haga

$$S_{N+1} = S_{N+1} + 1 \qquad (31)$$

En algún punto digamos en la solución del problema $P^{(K)}$ puede no existir solución. Cuando esto sucede, el método marca en la **(K + 1)th** variable una falla, reemplaza $S_{k+1} + 1$ y resuelve $P^{(K)}$. Cuando una variable se ha marcado dos veces, suspende de su valor constante y **K** se incrementa en 1 y el proceso continúa. El proceso total se termina cuando **K** = **0**. Las bases matemáticas del proceso de investigación anterior se presentan en [5]. Note que los cálculos de cota inferior no son empleados aquí.

ALGORITMO

La descripción formal de la versión especial del método Simplex Stopped para el problema de asignación de horario es como sigue:

P_0 $K = N + 1$

P_1 Resolver $P^{(k)}$. Si hay solución vaya a P_2; si no vaya a P_4.

P_2 Determinar el índice **h** tal que $W_h \ldots, W_K$ son enteros y W_{h-1} no es un entero, o de lo contrario $h = 1$. Si $h = 1$ el algoritmo termina. De lo contrario

P_3 Hacer $S_i = W_i$ para $i = h \ldots k$. Hacer $k = h - 1$, ir a P_1.

P_4 Sentar $K = k + 1$; si $K < N + 1$ ir al paso P_6. De lo contrario ir al paso P_5.

P_5 si $K = N + 1$ hacer $S_{N+1} = S_{N+1} + 1$ y $k = N$. Regresar a P_1.

P_6 Marcar una falla en la **(kth)** variable.

P_7 Si la **kth** variable presenta dos fallas ir a P_9; de lo contrario, continúe.

P_8 Reemplazar S_K por S_{K+1}. Hacer $K = k - 1$. Ir a P_1.

P_9 Hacer $S_k = 0$, reemplazar **K** por $K + 1$. Regresar a P_4.

El algoritmo podrá encontrar una solución al problema de asignación.

En el paso P_1, deben distinguirse los métodos de solución empleados para $k = N + 1$, de aquellos de solución empleados para $K < N + 1$. En el primer caso se puede emplear el método Simplex revisado y en el segundo debe recurrirse a otras técnicas de la programación lineal dado que ya se tiene una solución de partida (tal como método Simplex Dual).

IDEAS HEURISTICAS

Como se ha podido observar, el algoritmo anterior resuelve el problema de asignación de horario a un estudiante en forma óptima. Sin embargo, para resolver el problema completo es necesario recurrir a ideas heurísticas tales como:

1. Llevar a memoria los cursos ofrecidos.
2. Leer la solicitud del estudiante (con su conjunto de prioridades).
3. Formular el modelo de programación entera para un estudiante dado y resolver usando el método S^2. Asignar horario y actualizar las holguras de los cursos en los cuales fue asignado. (En caso de que no exista una solución factible, remueva las restricciones y trate de nuevo). Si todavía no hay solución, provea los mecanismos necesarios para efectuar la asignación manualmente después de consultar con el estudiante.
4. Si todas las solicitudes han sido consideradas, termine. De lo contrario regrese a 2.

De hecho esta heurística es un marco general, el cual puede adaptarse según las características del Departamento o Programa.

Por ejemplo, al formular el modelo, si un curso tiene varios grupos seleccione un número determinado en orden decreciente de la holgura, tratando de no seleccionar más de un grupo a la misma hora.

Adicionalmente se puede permitir que cada estudiante proponga un horario factible que a él le gustaría tener, el cual se puede usar como punto de partida y si es óptimo será el asignado.

RESULTADOS COMPUTACIONALES

Para efectos de comparación del método Simplex Stopped propuesto en [4] y la versión presentada aquí se han considerado los mismos ejemplos propuestos en la referencia anterior, a saber:

Problema correspondiente a la Tabla 1, el cual es un ejemplo muy sencillo y puede representar a un estudiante de últimos semestres, ya que es probable que por lo menos 2 de sus cinco cursos sean de grupo único y los restantes tres podrán ser de grupo múltiple.

Los ejemplos 2 y 3 corresponden a estudiantes que solicitan cinco cursos con posibilidades dadas en la Tabla 2. La suposición de cupos desiguales del caso (a) representa el Ejemplo 2, mientras que cupos iguales del caso (b) representan el Ejemplo 3.

Se espera que (a) sea la situación más típica y de hecho la más fácil para resolver. En cualquiera de los casos la matriz **A*** es de dimensiones 40 × 28

Tabla 2

Física

	LCV9	LCV11	LCV2	MJS8	MJS9	MJS10
(a)	18	17	18	17	17	18
(b)	18	18	18	18	18	18

Matemáticas

	LCV8	LCV9	LCV11	LCV2	MJS9	MJS10
(a)	19	19	19	20	19	19
(b)	20	20	20	20	20	20

Sistemas

	LCV8	LCV10	LCV1	MJS10
(a)	17	17	17	17
(b)	17	17	17	17

Economía

	LCV11	LCV2	MJS9
(a)	12	12	12
(b)	12	12	12

Laboratorio Física

	MJ1-3	MJ2-5	LC1-3
(a)	10	10	11
(b)	11	11	11

Nota: **C** representa miércoles

Los tiempos de computador (IBM 360-44), independientes de la formación del modelo, son dados a continuación para lo cual usaremos la siguiente convención:

1. Solución a cada problema de P.L., en un modelo a partir de una solución inicial.
2. Solución a cada problema de P.L., en un modelo a partir de la solución vigente.
3. Solución para los tres ejemplos.
4. Solución a un ejemplo en particular.
5. Uso de la ecuación (30).
6. Criterio de entrada: cupo máximo.
7. Criterio de entrada: cupo máximo en caso de empate en las variables candidatas a entrar en la base.

— Corrida 1 tiempo: 1 min. 43.35 seg.
Características: Uso de 1, 3 y 6

Ejemplo 1	3 P.L.	40 pivotes
Ejemplo 2	2 P.L.	45 pivotes
Ejemplo 3	7 P.L.	179 pivotes

— Corrida 2 tiempo: 1 min. 38.44 seg.
Características: Uso de 1, 3 y 7

Ejemplo 1	3 P.L.	39 pivotes
Ejemplo 2	2 P.L.	39 pivotes
Ejemplo 3	7 P.L.	174 pivotes

— Corrida 3 tiempo: 1 min. 14.14 seg.
Características: Uso de 1, 3, 5 y 7

Ejemplo 1	3 P.L.	39 pivotes
Ejemplo 2	2 P.L.	39 pivotes
Ejemplo 3	5 P.L.	116 pivotes

— Corrida 4 tiempo: 0 min. 51.02 seg.
Características: Uso de 2, 3 y 7

Ejemplo 1	3 P.L.	35 pivotes
Ejemplo 2	2 P.L.	29 pivotes
Ejemplo 3	7 P.L.	70 pivotes

— Corrida 5 tiempo: 0 min. 44.82 seg.
Características: Uso de 2, 3, 5 y 7

Ejemplo 1	3 P.L.	32 pivotes
Ejemplo 2	2 P.L.	29 pivotes
Ejemplo 3	5 P.L.	58 pivotes

Las corridas 6, 7 y 8 son iguales a la corrida 5, excepto que se considera cada ejemplo por separado de donde se obtiene:

Ejemplo 1	7.15 seg.
Ejemplo 2	17.65 seg.
Ejemplo 3	28.42 seg.

Si estas pruebas se realizan en un IBM 4341, los tiempos de ejecución se reducen por un factor de 6 de donde podemos comparar la solución dada por G. L. Thompson en el IBM 7090 y la obtenida con el algoritmo mejorado en un IBM 4341 así:

IBM 7090

Ejemplo 1	1.2 seg.	3 P.L.	31 pivotes
Ejemplo 2	8.4 seg.	4 P.L.	73 pivotes
Ejemplo 3	22.8 seg.	9 P.L.	210 pivotes

IBM 4341

Ejemplo 1	1.19 seg.	3 P.L.	32 pivotes
Ejemplo 2	2.94 seg.	2 P.L.	29 pivotes
Ejemplo 3	4.73 seg.	5 P.L.	58 pivotes

CONCLUSION

De los resultados anteriores se puede afirmar que es posible asignar el horario a un estudiante en un promedio de 4 seg. o registrar una facultad de

3.600 estudiantes en un período de 4 horas, lo cual es ventajoso en relación con los métodos manuales actuales.

Precisamente se espera obtener resultados definitivos mediante su implementación en alguna facultad, bien sea sobre una base de tiempo real o en una aplicación en línea.

Adicionalmente, de los resultados actuales se puede ver la mejora lograda en el método basado en el uso de la relación (30) y la consideración de la solución dejada por el problema P^k como solución inicial del problema $P^{(k-1)}$.

BIBLIOGRAFIA

Bazaraa, Mokhtar S. y John J. Jarvis. **Linear Programming and Network Flows.** John Wiley & Sons Inc., 1977.

Dantzig, George B. **Linear Programming and Extensions.** Princeton, New Jersey, 1974.

Nemhauser, George L. y Robert S. Garfinkel. **Integer Programming.** John Wiley & Sons Inc., 1972.

Thompson, G. L. **A Method for Scheduling Students to Classes, Recent Advances in Optimization Techniques.** Lavi, Abraham y Thomas P. Vogl. New York. John Wiley & Sons Inc., 1966, págs. 281-296.

Thompson, G. L., The Stopped Simplex Method: Basic Theory for Mixed Integer Programming: Integer Programming. **Revue Française d'Informatique et de Recherche Opérationnelle,** 1964, Vol. 8, págs. 159-182.