



Comparativa de Algoritmo Genético, Memético y SI para el diseño de horarios de una Institución Educativa

Lucero de Montserrat Ortiz Aguilar^a, Juan Martín Carpio Valadez^a, Héctor José Puga Soberanes^a,
Claudia Díaz^a y Carlos Lino Ramirez^a

^aInstituto Tecnológico de León, Maestría en Ciencias en Ciencias de la Computación, León, Gto.,
m09240932@itleon.edu.mx, jmcario61@hotmail.com, pugahector@yahoo.com,
posgrado@itleon.edu.mx, carloslino@itleon.edu.mx

RESUMEN

EL timetabling hace referencia a la calendarización de tareas en organizaciones como lo son Escuelas, hospitales, centros de transporte etc. y una correcta asignación de tareas nos permite optimizar y aprovechar los recursos de la organización. En las universidades, por lo general se busca generar horarios que satisfagan al alumno, al profesor y al plan de estudios. Por ello, tener un buen diseño de horarios nos dará como consecuencia que institución optimice su presupuesto de capital y recursos. Además brinda a los estudiantes, herramientas necesarias para terminar en tiempo y forma su plan de estudios. La calendarización de horarios depende fuertemente de tipo de escuela, universidad y/o plan educativo, por lo que no existe un modelo universal de diseño de horarios que pueda ser aplicado en cualquier caso. El conjunto de restricciones que debe contemplarse en el diseño de horarios involucra a los alumnos, maestros e infraestructura. En este sentido, la investigación en ciencias computacionales ha generado algunas herramientas que permiten optimizar o mejorar el proceso de elaboración de horarios. En este trabajo se propone el uso del Algoritmos Metaheurísticos a partir de la metodología API-CARPIO que consiste en generar horarios que garanticen el no cruce de materias que el alumno tomará permitiendo optimizar los recursos humanos y físicos. Se muestra que mediante la aplicación de algoritmos genéticos, Memético y Sistema Inmune se generan soluciones aceptables, para el problema de la combinación de: Classroom Assignment, Faculty y CourseTimetabling y son aplicados a instancias del Instituto Tecnológico de León logran obtener resultados confiables y comparables con los del experto.

1. INTRODUCCIÓN

Dentro de una institución es importante poder optimizar recursos, tanto de personal como materiales. En el caso de las instituciones educativas, se busca que los estudiantes aprovechen al máximo su estancia dentro de ésta, es decir, que su horario contenga cierto número de materias, para que un alumno regular pueda terminar en tiempo y forma con el plan de estudios. El problema de calendarización depende del tipo de escuela, universidad y sistema de educación, por lo cual no existe un diseño de horarios que pueda ser aplicado donde sea [1].

El University Timetabling es de los problemas más estudiados del timetabling, Adriaen et.al[2] lo clasifica en 5 grupos:

1. **Faculty Timetabling** . Es la asignación de maestros a materias.



2. **Class-Teacher Timetabling.** Es asignación de materias con el menor conflicto temporal posible entre grupos de alumnos.
3. **Course Timetabling.** Es la asignación de materias con el menor conflicto temporal posible entre alumnos individuales.
4. **Examination Timetabling.** Es la asignación de exámenes a los alumnos, de tal forma que el alumno no aplique dos pruebas al mismo tiempo.
5. **Classroom assignment.** Después de asignar las clases a los maestros, se asignan class-teacher a los salones.

Este trabajo se enfoca a generar soluciones aceptables a la combinación de Classroom Assignment, Faculty y Course Timetabling. Las instancias utilizadas pertenecen a datos reales del Instituto Tecnológico de León (ITL), donde la calendarización de horarios la elabora un experto humano y se busca mediante los algoritmos Metaheurísticos generar soluciones aceptables en un tiempo razonable. La comparativa entre algoritmo Genético, Memético y Sistema Inmune, aplicando pruebas estadísticas no paramétricas.

2. TEORÍA

2.1 FACULTY TIMETABLING

Werra y Asratian [1] consideran que el Faculty Timetabling extiende del modelo básico Class Teacher, el cual es comúnmente encontrado en la elaboración de horarios de universidades y escuelas. Además de que ellos muestran que es un problema NP-Completo y dicho modelo se define[1].

2.2 COURSE TIMETABLING

El course Timetabling puede ser definido como el proceso de asignar clases a recursos como lo son de tiempo (time slots), espacio (salones) y maestros (personal), mientras se satisfaga un conjunto restricciones [3]. Cada uno de los diferentes actores (administrador, maestros y estudiantes) tienen cada uno objetivos y estos tienen usualmente conflictos. La complejidad de las relaciones entre time slot, clases, salones y maestros hace que sea difícil de encontrar una solución factible [3].

2.3 TIPOS DE RESTRICCIONES.

Existen dos tipos de restricción [4]:

- **Hard.** Es la restricción que absolutamente no puede ser quebrantada.
- **Soft.** El conjunto de restricciones que se prefieren satisfacer pero no se supone satisfacerlas todas

2.4 Metodología API-Carpio

En este trabajo se usa la metodología API-Carpio[5] la cual describe el proceso de calendarización de horarios educativos como:

$$f(x) = FA(x) + FP(x) + FI(x) \quad (1)$$

Dónde:

$FA(x)$ = Número de estudiantes en conflicto dentro del horario x , (CTT).

$FP(x)$ = Número de profesores en conflicto dentro del horario x , (FTT).

$FI(x)$ = Número de aulas y laboratorios en conflicto dentro del horario x , (CATT).

Con:

$$FA = \sum_{j=1}^k FA_{V_j} \quad (2)$$

$$FP = \sum_{j=1}^k FP_{V_j} \quad (3)$$



$$FI = \sum_{j=1}^k FI_{V_j} \quad (4)$$

En los cuales:

FA_{V_j} = Número de estudiantes en conflicto dentro del vector V_j .

$FP_{V_j}(x)$ = Número de profesores en dentro del vector V_j .

$FI_{V_j}(x)$ = Número de aulas y laboratorios en conflicto dentro del vector V_j .

Donde a su vez:

$$FA_{V_j} = \sum_{s=1}^{(M_{V_j})-1} \sum_{l=1}^{(M_{V_i})-s} (A_{j,s} \wedge A_{j,s+l}) \quad (5)$$

$$FP_{V_j} = \sum_{s=1}^{(M_{V_j})-1} \sum_{l=1}^{(M_{V_i})-s} (P_{j,s} \wedge P_{j,s+l}) \quad (6)$$

$$FI_{V_j} = \sum_{s=1}^{(M_{V_j})-1} \sum_{l=1}^{(M_{V_i})-s} (I_{j,s} \wedge I_{j,s+l}) \quad (7)$$

Para los que:

$A_{j,s} \wedge A_{j,s+l}$ = Número de alumnos que demandan la inscripción simultanea de las materias $M_{j,s} \wedge M_{j,s+1}$.

$P_{j,s} \wedge P_{j,s+l}$ = Número de profesores que demandan la impartición de una misma materia $M_{j,s}$, como la materia $M_{j,s+1}$.

$I_{j,s} \wedge I_{j,s+l}$ = Número de aulas y laboratorios que pueden dar servicio tanto a la materia $M_{j,s}$, como la materia $M_{j,s+1}$.

M_{V_i} = Número de materias diferentes contenidas en un vector determinado, es decir $\eta(V_j) = M_{V_j}$.

2.5 ALGORITMO GENÉTICO

Los algoritmos Genéticos fueron desarrollados por J. Hollan en los 70s [6]. Los algoritmos Genéticos, son algoritmos de búsqueda basados en la mecánica de selección natural y de la genética natural. Además de que combinan la supervivencia del más apto entre estructuras de secuencias con un intercambio de información estructurado, aunque aleatorizado, para que se asemeje a las genialidades de las búsquedas humanas [6].

El proceso de un genético comienza con una población inicial de la cuales se seleccionan a los mejores individuos y pasan a formar parte de la nueva población en la siguiente generación; después se seleccionan a dos individuos para hacer una cruce y generar hijos lo cuales formaran parte de la nueva población; el siguiente paso es de la nueva población hacer una muta a un individuos y se repite este proceso hasta que nuestro criterio se cumpla. En [7] define el algoritmo que corresponde a un Genético Simple.

2.6 ALGORITMO MEMÉTICO

En 1976 Dawkins diseñó un meme, que era parecido a un gen en un algoritmo genético pero diferente. Supuso que existe un progreso como un gen del algoritmo genético que es transferido a la próxima generación, es decir, las características obtenidas se transfieren de una generación anterior a una siguiente, junto con los cambios de población [8]. Los componentes de un Algoritmo Memético son:

- Algoritmo Genético
- Local Search o Búsqueda Local.

El Local Search es una modificación que se puede llegar a hacer toda la población de individuos con la que se trabaja el algoritmo. Esto se hace haciendo una copia de cada individuo, donde esta copia es alterada de alguna forma; si la copia de un individuo en específico es mejor que la original, la copia reemplaza al individuo original.



2.7 SISTEMA INMUNE ARTIFICIAL

El proceso de del algoritmo de sistema Inmune Artificial consiste en generar una población de soluciones candidatas, después seleccionamos un porcentaje de los mejores individuos, los cuales son clonados, luego a estos individuos se les aplica una hipermuta y finalmente continuamos hasta llegar a nuestra solución objetivo, pero para evitar que nuestra población crezca sin medida se pone una poda la cual nos permitirá regresar al tamaño inicial de la población [9].

3. PARTE EXPERIMENTAL

Las instancias usadas para las pruebas con las Metaheurísticas antes mencionadas pertenecen a ITL, estas corresponden a dos planes educativos distintos, pertenecientes al año del 2009 y 2014, cuentan con aproximadamente de 46 a 58 clases (eventos) y una cantidad de 9 a 11 timeslots respectivamente. La configuración utilizada en los algoritmos Genético, Memético y SI se muestran en la tabla 1, donde tenemos que las llamadas a función fueron 1,000,000, la población inicial para cada uno fue la misma. El criterio de paro de los algoritmos fue el de llamadas a función.

Parámetro	Genético	Memético	Sistema Inmune
Población	20	20	20
Llamadas a función	1,000,000	1,000,000	1,000,000
Elitismo	0.1	0.1	NA
Cruza	0.9	0.9	NA
muta	0.15	0.15	NA
Poda	NA	NA	100

Tabla 1. Datos de la configuración Inicial del AG, AM y SI

La tabla 2 muestra los resultados obtenidos (conflictos) de acuerdo a la función objetivo mostrada en 1, donde se puede observar la mediana de las ocho instancias evaluadas con el AG, AM, SI.

Instancia	Mediana		
	AG	AM	SI
-			
1	562	548	600
2	327	282	352
3	511	463	522
4	296	274	350
5	282	262	315
6	172	144	198
7	204	152	272
8	166	158	248

Tabla 2. Medianas de los diferentes Algoritmos

Para aplicar las pruebas estadísticas no paramétricas se aplicó el test de Friedman [10], de donde se tomarán las medianas de los algoritmos genético, Memético y SI.

En la tabla 3 se muestran los rangos y resultados del test ómnibus de Friedman donde la hipótesis nula es $h_0 =$ No existen diferencias en el desempeño de los algoritmos y la hipótesis alternativa es $h_a =$ Existen diferencias entre Algoritmos. Como el valor P es menor que $\alpha = 0.05$ en los tres casos, entonces no tenemos suficiente evidencia para aceptar h_0 . Tomando como



algoritmo de control al Memético, debido a que es el que tiene el menor rango hacemos las pruebas post-hoc, con un valor de $\alpha = 0.05$.

Algoritmos	Friedman	Friedman Alineado	Quade
Genético	2	101	2
Memético	1	79	1
SI	3	119	3
Estadístico	16	8.35	27
Valor P	0.00033	0.01533	1.3706E-06

Tabla 3. Rangos, estadísticos y valor p para AG, AM y SI

En la tabla 4 se muestran los valores z y los valores p con ajuste Bonferroni[10]. Como podemos observar en las pruebas a pares para el caso del AM vs el AG en los tres test el valor p es menor que el α por lo cual no tenemos suficiente evidencia para rechazar h_0 , mientras que para el caso de AM vs SI nuevos valores p son menores que el α , por lo cual nos dice que existe diferencia en el comportamiento de los algoritmos.

Algoritmo	Friedman		Friedman Alineado		Quade	
	z	Bonferroni	z	Bonferroni	z	Bonferroni
-						
AM vs AG	2	0.0910	1.29	0.3897	1.78	0.1494
AM vs SI	4	0.0001	2.35	0.0368	3.56	0.0007

Tabla 4. Pruebas Post-Hoc. Tomando como algoritmo de control al Memético

4. CONCLUSIONES

La elaboración de horarios para instituciones educativas resulta ser una tarea muy exhaustiva y difícil de resolver para el experto humano de forma manual conforme se incrementan el número de eventos o materias y el número de restricciones. Es por eso que en este trabajo se propone el uso de algoritmos Metaheurísticos, los cuales nos generan soluciones aceptables en tiempos razonables.

De igual manera en este trabajo se muestran los resultados obtenidos de aplicar diferentes Metaheurísticas como lo son el Genético, Memético y SI, a las instancias reales del ITL. Como se puede apreciar en la sección de resultados el Algoritmo Memético fue el que tuvo mejor desempeño, más sin embargo la prueba de Friedman nos señala que no existe diferencia entre éste y el Genético y que solo existe diferencia con respecto al Sistema Inmune, esto con un $\alpha = 0.05$.

Como trabajo futuro se propone emplear otro tipo de selección, cruza y muta para el caso del genético y Memético e implementar otras versiones de Sistema Inmune que nos generen mejores soluciones. Además de integrar otros algoritmos Metaheurísticos u otro tipo como enfoques que nos ayuden a resolver y sean aplicables a este tipo de problema.

AGRADECIMIENTOS.

Agradecimientos Al Consejo Nacional de Ciencia y Tecnología (CONACYT) México por el apoyo brindado en esta investigación con el número de beca 308646 y a la División de Estudios de Posgrado del Instituto Tecnológico de León.



BIBLIOGRAFÍA

1. Asratian, A. S., de Werra, D., Luleå University of Technology, Department of Engineering Sciences and Mathematics, Mathematical Science, & Mathematics. (2002). A generalized class–teacher model for some timetabling problems. *European Journal of Operational Research*, 143(3), 531-542.
2. Adriaen, M., Causmaecker, P., Demeester, P.: Tackling the university course timetabling problem with an aggregation approach. In: Burke, K., Rudova, H. (eds.) *Proceedings PATAT 2006*, pp. 330–335 (2006).
3. LAI, L. F., WU, C., HSUEH, N., HUANG, L., & HWANG, S. (2008). an artificial intelligence approach to course timetabling. *International Journal on Artificial Intelligence Tools*, 17(1), 223-240.
4. McCollum, B., McMullan, P., Parkes, A. J., Burke, E. K., & Qu, R. (2012; 2011). A new model for automated examination timetabling. *Annals of Operations Research*, 194(1), 291-315.
5. Carpio-Valadez, J.M.: Integral Model for optimal assignation of academic tasks, encuentro de investigacion en ingenieria electrica. *ENVIE, Zacatecas*, 78–83 (2006).
6. Goldberg, D. E. (1989). *Genetic algorithms in search, optimization, and machine learning*. Reading, Mass: Addison-Wesley Pub. Co.
7. Yang, X.-S. (2010). *Nature-inspired metaheuristic algorithms*. Luniver press.
8. Moscato, P. (1989). "On Evolution, Search, Optimization, Genetic Algorithms and Martial Arts: Towards Memetic Algorithms". *Caltech Concurrent Computation Program* (report 826).
9. Azuaje, F. (2003). *Artificial immune systems: A new computational intelligence approach* Elsevier.
10. Joaquín Derrac, Salvador García, "A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm inte", *Swarm and Evolutionary Computation* (2011).