

## Comparativa entre algoritmos Mono y Multi-objetivo aplicados al problema de calendarización de horarios universitarios

A comparative between algorithms Mono and Multi-Objective applied to Timetabling University

Moises Romo-Franco<sup>1</sup>, Martin Carpio<sup>1</sup>, Lucero Ortiz-Aguilar<sup>1</sup> , Jorge Alberto Soria-Alcaraz<sup>2</sup>, Hector Puga<sup>1</sup>, Carlos Lino , Luis Ernesto Mancilla<sup>1</sup>

<sup>1</sup>Tecnológico Nacional de México-Instituto Tecnológico de León, León, Guanajuato, México

<sup>2</sup>Universidad de Guanajuato, Guanajuato, Guanajuato, México

Correo-e: moises1236@hotmail.es

### PALABRAS CLAVE

NSGA-II, Calendarización de Horarios Universitarios, algoritmo Genético, algoritmo Memético, Sistema inmune.

### RESUMEN

La calendarización de tareas en las instituciones educativas tiene como objetivo que los estudiantes tomen sus asignaturas correspondientes apeándose a un conjunto de restricciones. En el presente trabajo se muestra que, utilizando un enfoque Multi-objetivo junto a la metodología API-CARPIO, se generan soluciones aceptables para el problema de calendarización de horarios. Las instancias de prueba provienen de datos reales del Instituto Tecnológico de León (ITL). Los resultados del algoritmo Multi-objetivo NSGA-II y sus distintas configuraciones son comparados con los resultados de algoritmos Metaheurísticos, además con los resultados de un Experto Humano.

### KEYWORDS

NSGA-II, University Timetabling, Genetic algorithm, Memetic algorithm, Immune System algorithm

### ABSTRACT

The timetabling university aims to schedule a set of school subjects of students taking into consideration restrictions. In this work is showed that it's possible to generate solutions for the timetabling university, using a multiobjective approach together with API-CARPIO methodology. The test instances come from real data of Instituto Tecnológico de León (ITL). The solutions of multiobjective algorithm NSGA-II and their different configurations are compared against the solutions of metaheuristic algorithms, and also with solutions provided by a human expert.

Los problemas de calendarización de tareas o eventos, en general, están sujetos a conjuntos de restricciones tales como: creación de roles de trabajo, funcionamiento de distintos equipos, personal de operación, ..., entre otros. Este tipo de problemas han sido clasificados como clase NP [1].

De particular, el diseño de horarios en las instituciones educativas no ha podido realizarse de manera generalizada, debido a la gran cantidad de variantes que tiene, entre las más importantes se encuentran: alumnos, docentes, plan de estudios, sistema de educación e inmuebles con los que dispone cada institución [2].

De manera general el problema de calendarización de horarios puede ser definido como la asignación de un conjunto de eventos a un conjunto de espacios de tiempo, que está sujeto a un conjunto de restricciones [3].

En el estado del arte se han reportado formas de resolver el problema de Calendarización de Horarios utilizando algoritmos Heurísticos bajo el enfoque Mono-objetivo, tales como, Algoritmos Genéticos [4, 5, 6], Algoritmos Meméticos [7, 8], Búsqueda Local [9], Recocido Simulado [10], optimización por colonia de Hormigas [11], enfoque Hiperheurístico [12] entre otras [2].

Así mismo, el Enfoque Multi-objetivo ha sido utilizado para resolver problemas de Calendarización de Horarios mediante algoritmos heurísticos [13, 14], softwares [15], y el uso de métodos de reducción de objetivos [16]. Los trabajos citados anteriormente mostraron que dentro del enfoque Multi-objetivo existen diferentes metodologías de solución para la Calendarización de horarios universitarios.

Durante este trabajo, el problema de Calendarización de Horarios se abordó mediante el enfoque de Algoritmos Metaheurísticos Multi-objetivo. Sin embargo, dado que no existe Metaheurística que de la mejor solución a todos los problemas de calendarización de horarios universitarios (Teorema de No-Free Lunch [17]), es necesario buscar soluciones aceptables mediante diferentes técnicas Metaheurísticas.

Además, en este trabajo se reportaron los resultados que se obtuvieron en instancias reales del Instituto Tecnológico de León (ITL), donde la calendarización de horarios actualmente se elabora con Algoritmos Metaheurísticos propuestos por Aguilar et al. en [2] y posteriormente un experto humano da los ajustes necesarios para cubrir las

necesidades de la institución. Los Algoritmos Metaheurísticos propuestos por Aguilar et al. en [2], se utilizan para encontrar soluciones donde la prioridad es minimizar la cantidad de alumnos en conflicto en general, sin embargo, no se cuida que la cantidad de alumnos en conflicto sea mínima para cada horario, lo cual trae como consecuencias reubicación de docentes y una demanda mayor de grupos. En el caso planteado en este trabajo, se cuida que la cantidad de alumnos que entren en conflicto en un mismo horario sea mínimo.

En el presente artículo se utilizó el algoritmo NSGA-II, debido a su constante uso dentro del estado del arte en *Timetabling*, para encontrar soluciones aceptables para el problema de calendarización de horarios universitarios con instancias del ITL considerando la minimización de los alumnos en conflicto por cada horario. A continuación se realizó una comparativa de las soluciones obtenidas por las variantes del NSGA-II vs las dadas por Aguilar et al. en [2], con el fin de verificar, utilizando pruebas no paramétricas, si el enfoque Multi-objetivo compete con el utilizado en Aguilar et al. en [2]. Por último, se realizó la prueba de suma de rangos con signo de *Wilcoxon*, entre las soluciones del experto humano y las versiones Multi-objetivo, para verificar si el diseño de horarios bajo el enfoque Multi-objetivo es una herramienta apropiada para dar solución al problema de calendarización de horarios universitarios con instancias del ITL.

## 1. El problema de calendarización de Horarios Universitarios

La calendarización de Horarios Universitarios (*University Timetabling*) puede ser definida como el proceso de acomodar una serie de elementos  $e$  dentro de un conjunto de espacios de tiempo  $s$  de tal manera que no se violen una serie de restricciones  $c$  para cubrir las necesidades de horarios escolares [18].

Otra definición mencionada por [2] es que puede ser definido como el proceso de asignar clases a recursos como lo son de tiempo, espacio (salones) y maestros (personal), mientras se satisfaga un conjunto de restricciones.

Existen dos tipos de restricciones [19].

- Restricciones Duras (*Hard*): Es una restricción cuya violación impide que sea posible implementar la solución en el mundo real.

- Restricciones Blandas (*Soft*): Conjunto de restricciones cuya satisfacción completa es deseable.

### 1.1 Metodología API-Carpio.

La metodología API-Carpio [20] describe el proceso de calendarización de horarios educativos como:

$$f(x) = FA(x) + FP(x) + FI(x)$$

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).

En el presente trabajo abarcará únicamente hasta  $FA(x)$ , misma que es definida como:

$$FA = \sum_{j=1}^k FA_{V_j}$$

Dónde:

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

$FA_{V_j}$  = Número de estudiantes en conflicto con el vector  $V_j$ .

$V_j$  = Es un vector de tiempo que contiene diferentes materias.

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

$j$  = Es el número de vector.

$s$  = Es el índice de las materias contenidas dentro del vector  $j$ .

### 1.2 Metodología del diseño

La metodología del diseño utilizada en el presente trabajo fue la propuesta por Soria et al. [21], y ha sido utilizada en [22] y [2], siendo este último el que contiene los resultados a comparar con este trabajo.

La metodología consiste en la modelación de diferentes restricciones de la calendarización de tareas y listas de restricciones mediante la conversión de las restricciones de tiempo y espacio en un simple tipo de restricción: conflictos de estudiantes. Con este fin en [21] se propone estructuras como la matriz MMA, lista LPH, lista LPA y lista LPS. Las primeras tres representan restricciones duras y la última las restricciones blandas [2]. En el presente trabajo se utilizaron las dos primeras estructuras anteriormente mencionadas. En [21] se definen las estructuras de la siguiente manera:

- **Matriz MMA:** Contiene el número de conflictos (entre alumnos) posibles si dos lecciones son asignadas en el mismo espacio de tiempo.
- **Lista LPH:** Esta lista nos da información acerca de cada lección (clase, evento o materia) en que posible espacio de tiempo puede ser asignada.

## 2. Algoritmos Evolutivos

Los algoritmos evolutivos basan su funcionamiento en la teoría de la evolución de las especies, la supervivencia del más apto y la trasmisión de características de padres a hijos [23]. En los últimos años, han sido desarrollados una gran variedad de algoritmos evolutivos: Algoritmos Genéticos, Estrategias evolutivas, Programación evolutiva, Programación Genética, entre otras. Cada uno con una propuesta diferente, sin embargo, están inspirados en el mismo principio de evolución natural [24].

### 2.1 Algoritmo Genético

Los AGs fueron desarrollados por J. Holland en los 70s, con el fin de entender el proceso adaptativo del sistema natural, para luego aplicarlo en la optimización y Machine Learning en los 80s [25]. Los algoritmos genéticos (AGs) son algoritmos de búsqueda, optimización y máquinas de aprendizaje basados en la teoría de la Selección Natural, la Evolución y la Genética [26].

### 2.2 Algoritmo Memético

En 1976 Dawkins diseñó el concepto de meme, el cual a diferencia del gen puede ser modificado por su portador. 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. [2]. Los componentes de un Algoritmo Memético son: Algoritmo Genético y Búsqueda Local [27].

### 2.3 Sistema Inmune

Estos se basan en imitar el comportamiento del sistema inmunológico humano, el cual se encarga de proteger al cuerpo de los patógenos

externos e internos y su tarea principal es reconocer las células en el cuerpo clasificarlas como propias y no propias. Los algoritmos de Sistema Inmune artificial han sido aplicados con éxito en diversos problemas de optimización [28].

Básicamente el proceso del algoritmo de Sistema Inmune Artificial consiste en generar aleatoriamente 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 [28].

### 3. Enfoque Multi-objetivo

Los problemas de optimización Multi-objetivo (POM) pueden formularse de la siguiente manera: [29]

$$POM = \begin{cases} \min F(x) = (f_1(x), f_2(x), \dots, f_k(x)) \\ \text{s.a. } x \in X \end{cases} \quad (1)$$

Donde  $k \geq 2$  es el número de funciones objetivo.

$x = (x_1, x_2, \dots, x_n)$  es el vector de variables de decisión.

$X$  es el espacio de soluciones factibles.  $X$  es definido habitualmente como el conjunto de restricciones:

$$\{x | g_j(x) \leq 0, j = 1, 2, \dots, m \text{ y } h_i(x) = 0, i = 1, 2, \dots, e\}$$

$F(x)$  es el verdadero objetivo.

$y = (y_1, y_2, \dots, y_k)$ , con  $y_i = f_i(x)$ , es el valor de una solución en la espera de soluciones factibles.

Lo ideal a encontrar en problemas Multi-objetivo es un Vector ideal  $y^*$  que minimice todas las funciones objetivo de manera simultánea, sin embargo, eso usualmente no es posible. [30]. Por lo tanto, en varios problemas de la vida real, los objetivos en cuestión entran en conflicto entre sí. La solución entonces que se puede dar a problemas Multi-objetivo es un conjunto de soluciones, mismas que satisfagan los objetivos en un nivel aceptable y no sean dominadas por alguna otra solución. [30].

#### 3.1 Objetivos

En el presente trabajo se manejan un total de 3 objetivos, los cuales se definen de la siguiente manera:

El primer objetivo es la minimización de los conflictos pertenecientes a los vectores impares:

$$FA = \sum_{j=1}^{k-\lfloor \frac{k}{2} \rfloor} FA_{V_{2j-1}} \quad (2)$$

El segundo objetivo es la minimización de los conflictos pertenecientes a los vectores pares:

$$FA = \sum_{j=1}^{\lfloor \frac{k}{2} \rfloor} FA_{V_{2j}} \quad (3)$$

Donde [31]:

$$\lfloor \frac{k}{2} \rfloor = \begin{cases} \frac{k}{2} & \text{si } k = \text{Número Par } (2, 4, \dots) \\ \frac{k-1}{2} & \text{si } k = \text{Número Impar } (1, 3, \dots) \end{cases}$$

El tercer objetivo es la minimización de la suma de la cantidad de conflictos de alumnos en todos los vectores:

$$FA = \sum_{j=1}^k FA_{V_j} \quad (4)$$

### 4.2 NSGA II

El NSGA (Nondominated Sorting Genetic Algorithm) fue desarrollado en 1998 por Sribvias y Dep [32]. Más tarde, Deb [33] mejoró el NSGA original y lo renombró NSGA-II. La principal diferencia entre ambos algoritmos, es que el NSGA II emplea un archivo para contener las mejores soluciones y una regla de asignación de adaptación que tiene en cuenta, tanto el rango, como la distancia de cada solución [33].

Las características principales del NSGA-II son las siguientes:

- Contiene el esquema general de un Algoritmo Genético Simple.
- El ordenamiento no dominado mediante una técnica de comparación que utiliza una subpoblación auxiliar.
- El uso de una técnica de agrupamiento (Crowding), que no requiere la especificación de parámetros adicionales para mantener la diversidad en la población.

El algoritmo 1 propuesto por [33], corresponde al NSGA-II.

---

**Algoritmo** NSGA-II

**1**

---

**Requiere:** Función Objetivo  $F(x)$ ,  $x = (x_1, \dots,$

- 
- $x_n)^T$
- 1: Generar una población Inicial (Pt)
  - 2: Ct//Cantidad de llamadas a función definidas para el algoritmo
  - 3: **Mientras** (Ct>0) **hacer**
  - 4:     Se realiza un Ordenamiento no Dominado para obtener los niveles de no dominancia de Pt.
  - 5:     Se obtiene la distancia de Agrupamiento de Pt.
  - 6:     Hacer Cruza de 1 punto y bajo una probabilidad una Muta para generar Qt.
  - 7:     Se unen las dos poblaciones Pt y Qt para generar Rt.
  - 8:     Se realiza un Ordenamiento no Dominado para obtener los niveles de no dominancia de Rt.
  - 9:     Se obtiene la distancia de Agrupamiento de Rt.
  - 10:    Se seleccionan los N individuos mejores de Rt, de acuerdo a su nivel de dominancia y al valor de la distancia de Agrupamiento, y son ingresados en Pt.
  - 11:    **Fin del Lazo.**
  - 12:    Se regresan las soluciones no Dominadas.
- 

El Ordenamiento no Dominado tiene como objetivo ordenar la población de tamaño N de acuerdo al nivel de no dominancia, en un ordenamiento sencillo, cada solución debe ser comparada con las demás soluciones en la población a fin de saber si ésta es dominada o no [34].

El ordenamiento es conformado de dos procesos:

- Encontrar los individuos que pertenezcan al Frente 1 y que no sean dominados por otros individuos.
- Encontrar el frente al que pertenecen los individuos restantes. En este proceso los individuos pueden ser dominados por individuos pertenecientes al Frente 1 o a frentes superiores.

La distancia de Agrupamiento (D. A.) Crowding es una medida que indica la distancia promedio de dos puntos vecinos, uno de cada lado de punto de interés, utilizando el valor de sus funciones objetivos. Esta cantidad sirve como un estimador del perímetro del cuboides formado por los vecinos más cercanos a los vértices [34].

La distancia de Agrupamiento Crowding es dada por [35]:

$$d_i = \sum_{j=1}^m \frac{f_i^{(j+1)} - f_i^{(j-1)}}{f_i^{max} - f_i^{min}} \quad (5)$$

Donde:

$f_i^{(k)}$  = la función objetivo  $i$  de la solución  $k$ .

$f_i^{max}$  = el máximo valor del objetivo  $i$  de todas las soluciones.

$f_i^{min}$  = el mínimo valor del objetivo  $i$  de todas las soluciones.

Los algoritmos del ordenamiento no Dominado y la Distancia de Agrupamiento Crowding utilizados en el presente trabajo se encuentran en [34].

El método de selección K-Torneos consiste en seleccionar al azar un número  $k$  ( $k=3$  en este trabajo) de individuos. De entre los individuos seleccionados se selecciona el individuo que se encuentre en un menor frente de Pareto. Si dos o más individuos tienen en común en mismo frente, se selecciona al individuo con la distancia *Crowding* más alta y se pasa a la siguiente generación [34].

La cruza consiste en el intercambio de material genético entre dos cromosomas [36] y la utilizada en el presente trabajo es la cruza a un punto, la cual recibe dos soluciones padres y selecciona aleatoriamente un punto dentro de sus características.

La mutación es una perturbación a un individuo de la población. En el presente trabajo se manejaron 3 Mutas diferentes: *La muta a un gen*, que consiste en la selección de un punto aleatoriamente e intercambiar el valor de esa posición por otro que se encuentre en la LPH [29]. *La muta a dos genes*, que consiste en la selección aleatoria de dos eventos de un mismo semestre e intercambiar los valores de esas posiciones entre sí, siempre y cuando, ambos se encuentren en la LPH [19]. *La muta combinada*, que consiste en dado un número seleccionado aleatoriamente entre 0 y 1, si este es mayor a un umbral con un valor de 0.5, se realiza la muta a un gen, en cambio si él número aleatorio es menor o igual a 0.5, se realiza la muta a dos genes.

#### 4. Experimentación y Resultados.

Durante el presente trabajo se realizaron 4 variantes diferentes del algoritmo NSGA-II, su configuración es mostrada en la tabla 1, donde se tiene que las llamadas a función fueron Ct = 300,000, siendo este último el criterio de paro de los algoritmos. Así mismo, las configuraciones utilizadas para los Algoritmos Genético, Memético y Sistema Inmune se encuentran en [2].

**Tabla 1.** Datos para las configuraciones de las variantes de NSGA-II

Parámetro	NSGA-II50M1	NSGA-II20M1	NSGA-II50M2	NSGA-II50M3
Población	50	20	50	50
Selección de Cruza	K-Torneos			
Probabilidad de Mut	0.15			
Cruza utilizada	1 Punto			

Algoritmos	Test de las Pruebas Estadísticas no Paramétricas			
	Friedman	Friedman Alineado	Quade	
AG	1.125	8.125	1.1111	
AM	1.875	11	1.8889	
SI	5.125	37.5	5.0556	
NSGAI50M1	3.875	28.25	3.9444	
NSGAI20M1	3.5	22.5	3.5278	
NSGAI50M2	6.875	50.25	6.7778	
NSGAI50M3	5.625	41.875	5.6944	
Valor p	0	0.329767829047	0.000000001	229
Muta utilizada	Un gen	Un gen	Dos genes	Combinada
D. A.	Crowding			

Las instancias utilizadas, durante el presente trabajo, para las pruebas con las variantes del NSGA-II, son las mencionadas en [19], estas corresponden a dos planes educativos distintos (2004 y 2010), pertenecen a las carreras LIA (Licenciatura en Informática) del año 2009 e IGE (Ingeniería en Gestión Empresarial) del año 2014 las cuales fueron ofertadas a los alumnos de manera Semestral y en los turnos matutino y vespertino. Las instancias cuentan con aproximadamente de 46 a 58 clases (eventos) y una cantidad de 9 a 11 espacios de tiempo respectivamente.

Se realizaron 33 ejecuciones de cada algoritmo con cada instancia, con el fin de obtener los estadísticos: media, mediana y desviación estándar del conjunto de soluciones otorgadas por los algoritmos. Dichos estadísticos son mostrados en las tablas 2, 3 y 4. Los valores de los algoritmos Genético, Memético y Sistema Inmune son los presentados en [2] y los valores de las variantes del algoritmo NSGA-II en el presente trabajo, fueron resultado de la evaluación en el tercer objetivo mostrado en la sección 4.1 ecuación 4, debido a que es la minimización total de los conflictos en todos los vectores. Las hipótesis utilizadas en las pruebas no paramétricas fueron las siguientes:

$H_0$  (Hipótesis Nula) = El Desempeño de los algoritmos es idéntico.

$H_a$  (Hipótesis Alternativa) = El Desempeño de los algoritmos es diferente.

El nivel de significancia para las pruebas no paramétricas se estableció  $\alpha = 0.05$ .

La tabla 2 muestra los conflictos de las medianas obtenidas con los algoritmos Multi-objetivo. Los valores resaltados en negro son los experimentos en los cuales los valores del primer (Ver ecuación 2) y segundo (Ver ecuación 3) objetivo tienen un valor igual o muy cercano entre sí, lo cual nos muestra que los conflictos se distribuyen de una manera muy cercana entre los vectores pares e impares, siendo este un efecto deseable de aplicar el enfoque Multi-objetivo a las instancias presentadas.

**Tabla 2.** Medianas de los conflictos por objetivo en cada instancia.

Las cantidades remarcadas en negro en la tabla 3, son los mejores resultados obtenidos de cada instancia de todos los algoritmos. Las cantidades en cursiva son los mejores resultados de cada instancia en los algoritmos Multi-objetivo

**Tabla 3.** Medianas de las diferentes Metaheurísticas aplicadas a las instancias

I	AG	AM	SI	NSGAI50M1	NSGAI20M1	NSGAI50M2	NSGAI50M3
1	279	282	304	332	299	390	364
2	162	160	209	193	175	244	219
3	250	262	325	295	296	359	332
4	147	152	189	168	171	213	190
5	131	137	178	146	142	192	161
6	74	77	99	88	80	109	96
7	75	81	190	99	85	140	119
8	89	95	109	113	123	178	145

La tabla 4 se muestra los resultados de la Media y la Desviación Estándar de los algoritmos utilizados, remarcando los algoritmos con menor desviación estándar. Se observa que en 7 de las 8 instancias utilizadas, las variantes del NSGA-II obtuvieron una menor desviación estándar, indicando menor dispersión en sus resultados.

**Tabla 4.** Media y Desviación Estándar de las diferentes Metaheurísticas aplicadas a las instancias

I	AG	AM	SI	NSGAI50M1	NSGAI20M1	NSGAI50M2	NSGAI50M3
1	281.3±16.24	282.6±16.10	295.7±18.70	332.7±14.11	302.0±22.25	389.7±15.28	357.3±17.06
2	161.6±14.73	162.5±14.00	194.2±13.54	193.0±11.18	174.1±15.41	242.3±14.85	216.7±15.65
3	247.6±19.71	265.0±19.84	293.9±20.58	297.8±18.19	292.6±17.35	355.3±14.16	331.1±15.45
4	160.6±11.03	150.9±10.61	177.3±9.07	168.5±10.87	168.4±8.19	210.4±8.27	188.9±7.90
5	129.1±10.34	135.7±8.02	164.7±9.59	147.9±12.86	144.9±12.75	192.0±19.27	162.7±13.36
6	73.9±6.81	77.9±7.19	89.0±7.73	87.7±7.40	81.3±6.52	109.5±6.89	96.5±8.68
7	75.0±7.69	79.5±9.47	180.9±9.92	98.0±9.54	86.7±7.70	137.6±8.86	117.5±11.51
8	91.1±12.80	97.0±16.99	104.7±11.55	110.9±14.56	120.5±11.10	179.2±17.15	147.6±19.61

Con la finalidad de obtener una comparación más justa entre los Algoritmos se realizaron las

pruebas estadísticas no paramétricas *Friedman*, *Friedman Alineado* y *Quade*. Si el valor  $p$  obtenido en las pruebas no paramétricas es menor de  $\alpha$  se rechaza  $H_0$ , de lo contrario la prueba carece de significancia estadística. Para realizar los test, se tomaron los datos de la mediana mostrados en la Tabla 3.

En la tabla 5 se observa que el valor  $p$  es menor a  $\alpha$  en las pruebas de *Friedman* y en la de *Quade*, por lo que se rechaza  $H_0$ . Dicho esto, se procedió a realizar pruebas *post-hoc* en las pruebas de *Friedman* y *Quade*, tomando como algoritmo de control al Genético ya que tuvo menor ranqueo.

**Tabla 5.** Ranqueo y valor  $p$  para los Algoritmos Metaheurísticos

I	NSGAII50M1			NSGAII20M1			NSGAII50M2			NSGAII50M3		
	Ob1	Ob2	Ob3									
1	136	196	332	188	111	299	199	191	390	171	193	364
2	112	81	193	79	96	175	119	125	244	113	106	219
3	185	110	295	152	144	296	231	128	359	160	172	332
4	82	86	168	89	82	171	107	106	213	108	82	190
5	86	60	146	68	74	142	107	85	192	105	56	161
6	31	57	88	43	37	80	51	58	109	47	49	96
7	53	46	99	47	38	85	70	70	140	62	57	119
8	55	58	113	38	85	123	99	79	178	88	57	145

En la tabla 6 se muestran los resultados de las pruebas *post-hoc*, donde se observan los valores  $p$  con ajuste *Bonferroni* [38]. En las pruebas de *Friedman* y *Quade*, se observa que los NSGA-II50M2, NSGA-II50M3 y el Sistema Inmune el valor  $p$  son menores que  $\alpha$ , lo cual indica que existe una diferencia entre estos.

**Tabla 6.** Pruebas *post-hoc*. Tomando como algoritmo de control al Genético

Algoritmo de control	Friedman		Quade	
	Genético vs	Valor p con ajuste Bonferroni	Genético vs	Valor p con ajuste Bonferroni
AM		1		1
SI		0.001277		0.361582
NSGAII50M1		0.065378		1
NSGAII20M1		0.167343		1
NSGAII50M2		0.000001		0.041703
NSGAII50M3		0.000186		0.174135

Para verificar que algoritmo es el que otorga las mejores soluciones se realizaron pruebas de sumas de rangos con signo de *Wilcoxon*. Los pasos para obtener los valores  $W^+$ ,  $W^-$ ,  $W_0$  y  $W$  se muestran en el Apéndice D de [19].

La Tabla 7 muestra la prueba de sumas de rangos con signo *Wilcoxon* entre algoritmo genético vs los algoritmos con los que mostró tener un comportamiento distinto en las pruebas *post-hoc*. En los 3 casos al resultar  $W < W_0$ , entonces se tiene suficiente evidencia estadística para indicar que el algoritmo Genético tiene mejor desempeño que los algoritmos NSGA-II50M2, NSGA-II50M3 y Sistema Inmune.

**Tabla 7.** Prueba se suma de rangos con signo de *Wilcoxon* entre los resultados de conflictos del algoritmo

Genético vs NSGA-II50M2, NSGA-II50M3, Sistema Inmune.

Algoritmos a comparar	Grados de libertad (n)	$W^+$	$W^-$	$W$	$W_0$
Genético vs Sistema Inmune	8	0	34	0	4
Genético vs NSGA-II50M2	8	0	36	0	4
Genético vs NSGA-II50M3	8	0	36	0	4

**Tabla 8.** Prueba se suma de rangos con signo de *Wilcoxon* entre los resultados de conflictos del algoritmo Genético vs NSGA-II20M1, NSGA-II20M1, Memético.

Algoritmos a comparar	Grados de libertad (n)	$W^+$	$W^-$	$W$	$W_0$
Genético vs Memético	8	1	32	1	4
Genético vs NSGA-II50M1	8	0	35	0	4
Genético vs NSGA-II20M1	8	0	36	0	4
Memético vs NSGA-II50M1	8	0	34	0	4
Memético vs NSGA-II20M1	8	0	36	0	4
NSGA-II50M1 vs NSGA-II20M1	8	28	9	9	4

Al aplicar la prueba de suma de rangos con signo de *Wilcoxon* entre los algoritmos que no mostraron diferencia entre su desempeño (Ver Tabla 8), se observó que las soluciones otorgadas por el Algoritmo Genético mejoraron las obtenidas por los algoritmos Memético, NSGA-II50M1, NSGA-II20M1. Además, el algoritmo Memético mejoro los resultados de las variantes Multiobjetivo.

Estadísticamente, el algoritmo Genético tuvo mejor desempeño vs el resto de los algoritmos. Por lo que en [19] se realizó una prueba de suma de rangos con signo de *Wilcoxon* del algoritmo Genético vs las soluciones dadas por un Experto Humano. Concluyendo que el algoritmo Genético mejora los resultados de un Experto Humano.

Para indagar si el enfoque Multi-objetivo supera los resultados de un Experto Humano, se realizaron pruebas de suma de rangos con

signo de *Wilcoxon* entre estos. Dichas pruebas (Ver Tabla 9) afirman que los algoritmos Multi-objetivo mejoraron los resultados del experto humano, por lo tanto, pueden ser utilizadas en la calendarización de horarios.

**Tabla 9.** Prueba se suma de rangos con signo de *Wilcoxon* entre los resultados de conflictos del Experto Humano vs los algoritmos Multi-objetivo

Algoritmos a comparar	Grados de libertad (n)	$W^+$	$W^-$	$W$	$W_0$
NSGA-II50M1 vs Experto	8	0	36	0	4
NSGA-II20M1 vs Experto	8	0	36	0	4
NSGA-II50M2 vs Experto	8	0	36	0	4
NSGA-II20M3 vs Experto	8	0	36	0	4

Para conocer cual de los algoritmos Multi-objetivo presentados ofrece mejores resultados, se realizó la prueba de suma de rangos con signo de *Wilcoxon*. Dicha prueba (Ver tabla 10) indica que los algoritmos NSGA-II50M1 y NSGA-II20M1 son mejores que NSGA-II50M2 y NSGA-II50M3. Sin embargo la prueba realizada (Ver Tabla 8) entre los algoritmos NSGA-II50M1 y NSGA-II20M1 no permite conocer cuál ofrece mejores resultados. Esto afirma, que los algoritmos NSGA-II50M1 y NSGA-II20M1 son las mejores variantes Multi-objetivo mostradas en este trabajo

**Tabla 10.** Prueba se suma de rangos con signo de *Wilcoxon* entre los resultados de conflictos de los algoritmos Multiobjetivo.

Algoritmos a comparar	Grados de libertad (n)	$W^+$	$W^-$	$W$	$W_0$
NSGA-II50M1 vs NSGA-II50M2	8	0	36	0	4
NSGA-II50M1 vs NSGA-II50M3	8	0	35	0	4
NSGA-II20M1 vs NSGA-II50M2	8	0	35	0	4
NSGA-II20M1 vs NSGA-II50M3	8	0	35	0	4

NSGA-II50M2 vs NSGA-II20M3	8	36	0	0	4
----------------------------	---	----	---	---	---

## 5. Conclusiones

En el presente trabajo se mostró evidencia estadística (Ver Tabla 7 y 8) que afirmo que los algoritmos NSGA-II de este trabajo no mejoraron a los dados por Aguilar et al. en [2]. A pesar de lo anterior, los algoritmos NSGA-II demostraron una mayor confiabilidad estadística al tener menor dispersión (Ver tabla 4) en sus soluciones. Además, las soluciones dadas por los NSGA-II contemplaron la minimización de los conflictos de alumnos por horario, lo que permitirá generar un diseño de horarios en el cual un experto humano disminuirá o incluso no necesitará realizar ajustes, tales como la apertura de grupos extras o mayor demanda de docentes y aulas.

Por último, los resultados obtenidos por los algoritmos NSGA-II presentados, mejoraron a los obtenidos por un Experto Humano, lo que se reafirma que los algoritmos NSGA-II puede ser una herramienta de apoyo de gran ayuda para el diseño de horarios.

Como trabajo a futuro se propone la integración de otros algoritmos Multi-objetivo para dar solución al problema de calendarización de horarios. Aguilar et al. en [2] utilizó los algoritmos Memético y Sistema Inmune, por lo que se considera la integración de las versiones Multi-objetivo de los dichos algoritmos. También existe un interés por integrar algoritmos Heurísticos Trayectoriales, tales como, el Recocido Simulado o el Algoritmo de Búsqueda local Iterada, ambos en sus versiones Multi-objetivo. Lo anterior con la finalidad de experimentar con una cantidad mayor de instancias y aplicar contraste estadístico no paramétricos para determinar cuál de los algoritmos logran mejor desempeño.

También en un trabajo a futuro se propondrá una adecuación a la función objetivo utilizado actualmente, que minimice la diferencia entre los conflictos del primer y segundo objetivo, buscando una distribución uniforme de conflictos entre los vectores.

Por último, se tiene contemplado aplicar la comparativa entre algoritmos Mono y Multi-Objetivo bajo el fitness modificado aplicados al problema de calendarización de horarios universitarios.

## Agradecimientos

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

## Referencias

1. Jorge A. S., Martin, C. J., & Hugo, T. Diseño de Horarios Mediante Algoritmos Genético. *Décima Primera Reunión de Otoño de Potencia, Electrónica y Computación del IEEE, SI ROPEC*, Morelia. 2009.
2. Aguilar, L. D. M. O., Valadez, J. M. C., Soberanes, H. J. P., González, C. L. D., Ramírez, C. L., & Alberto Soria-Alcaraz, J. Comparativa de algoritmos bioinspirados aplicados al problema de calendarización de horarios. *Research in Computing Science*. 2015, vol 94, 33-43.
3. Soria-Alcaraz J., A., Martín, C., Héctor, P., & Sotelo-Figueroa, M. A. Comparison of metaheuristic algorithms with a methodology of design for the evaluation of hard con-strains over the course timetabling problem. Berlin, Heidelberg: *Springer Berlin Heidelberg*. 2013, pp. 289–302. Avariable from: doi:10.1007/978-3-642-33021-6\_23.
4. Asratian, A. S., de Werra, D. A generalized class–teacher model for some Timetabling problems. University of Technology, Department of Engineering Sciences and Mathematics, Mathematical Science. *Mathematics, European Journal of Operational Research*. 2002, pp. 531–542. Avariable from: doi:10.1016/S0377-2217(01)00342-3.
5. Deng, X., Zhang, Y., Kang, B., Wu, J., Sun, X., & Deng, Y. An application of genetic algorithm for university course timetabling problem. In *Control and Decision Conference (CCDC)*. 2011, *Chinese*, pp. 2119-2122, Avariable from: doi:10.1109/CCDC.2011.5968555.
6. Mahiba, A.A. & Durai, C.A.D. Genetic algorithm with search bank strategies for university course timetabling problem. *Procedin Engineering*. 2012, vol. 38, pp. 253–263.
7. Soria-Alcaraz, J. A., Carpio, J. M., Puga, Hé., Melin, P., Terashima-Marn, H., Reyes, L. C. & Sotelo-Figueroa, M. A. Castillo, O., Melin, P.; Pedrycz, W. & Kacprzyk, J. Generic Memetic Algorithm for Course Timetabling. In: *ITC2007 Recent Advances on Hybrid Approaches for Designing Intelligent Systems*, Springer. 2014, vol. 547, pp. 481–492.
8. Nguyen, K., Lu, T., Le, T., & Tran, N. Memetic algorithm for a university course timetabling problem. In *Informatics in Control, Automation and Robotics*. 2011, pp. 67–71. Avariable from: doi:10.1007/978-3-642-25899-2\_10.
9. Joudaki, M., Imani, M., & Mazhari, N. Using improved Memetic algorithm and local search to solve University Course Timetabling Problem (UCTTP). *Doroud, Iran: Islamic Azad University*. 2010.
10. Frausto-Solís, J., Alonso-Pecina, F., & Mora-Vargas, J. An efficient simulated annealing algorithm for feasible solutions of course timetabling. In *Mexican International Conference on Artificial Intelligence*. Springer. 2008, pp. 675–685.
11. Thepphakorn, T., Pongcharoen, P., & Hicks, C. An ant colony based timetabling tool. *International Journal of Production Economics*. 2014, vol. 149, pp. 131–144. Avariable from: doi:10.1016/j.ijpe.2013.04.026.
12. Soria-Alcaraz, J., Ochoa, G., Swan, J., Carpio, M., Puga, H., & Burke, E. Effective learning hyper-heuristics for the course timetabling problem. *European Journal of Operational Research*. 2014, pp. 77–86. Avariable from: doi:10.1016/j.ejor.2014.03.046.
13. Carrasco, M. P., & Pato, M. V. A multiobjective genetic algorithm for the class/teacher timetabling problem. In *International Conference on the Practice and Theory of Automated Timetabling*. 2000, pp. 3-17. Springer, Berlin, Heidelberg.
14. Datta, D., Deb, K., & Fonseca, C. Multi-objective evolutionary algorithm for university class timetabling problem. *Evolutionary Scheduling*. 2007, pp.197-236.
15. Azadeh, A., Gholizadeh, H., & Jeihoonian, M. A multi-objective optimisation model for university course timetabling problem using a mixed integer dynamic non-linear programming. *International Journal of Services and Operations Management*. 2013, 15(4), 467-481.
16. Movahhedfar, N., Ranjbar, M., Salari, M., & Rostami, S. Memetic and scatter search metaheuristic algorithms for a multiobjective fortnightly university course timetabling problem: a case study. *Journal of Industrial and Systems Engineering*. 2013, Vol 6.
17. Wolpert, H., Macready, G. No free lunch Theorems for Search. Technical report, *The Santa Fe Institute*. 1996, vol. 1.
18. Conant-Pablos, S. E., Magaña-Lozano, D. J., Terashima-Marín, H. Pipelining memetic algorithms, constraint satisfaction, and local search for course timetabling. *MICAI Mexican international conference on artificial intelligence*. 2009, vol. 1, 408–419.
19. Ortiz, L. Diseño de horarios de Alumnos y Maestros mediante Técnicas de Soft Computing, para una Institución Educativa. *Master's thesis, Instituto Tecnológico de León*. 2015.
20. Carpio-Valadez, J.M.: Integral Model for optimal assignation of academic tasks. In: En-

- cuentro de investigación en ingeniería eléctrica, ENVIE, Zacatecas. 2006, pp. 78–83.
21. Soria-Alcaraz, J. A., Martín, C., Héctor, P., Hugo, T., Laura, C. R., & Sotelo-Figueroa, M. A. Methodology of design: A novel generic approach applied to the course timetabling problem. *In Soft Computing Applications in Optimization, Control, and Recognition*. 2013, pp. 287–319. Available from: doi:10.1007/978-3-642-35323-9-12.
  22. Ortiz-Aguilar, L. D. M., Carpio, M., Soria-Alcaraz, J. A., Puga, H., Díaz, C., Lino, C., & Tapia, V. Training OFF-Line Hyperheuristics For Course Timetabling Using K-Folds Cross Validation. *En la revista programación matemática y software*. 2016.
  23. D. B. Fogel, Evolutionary Computation. The Fossil Record. Selected Readings on the History of Evolutionary Algorithms., *New York*, 1998.
  24. Glover, F., Laguna, M., & Martí, R. Scatter Search, to appear in Theory and Applications of Evolutionary Computation: Recent Trends, A. Ghosh and S. Tsutsui. 1999.
  25. Goldberg, D. E. Genetic Algorithms in Search, Optimization and Machine Learning. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1989, 1st edition.
  26. M. Mitchell, An Introduction To Genetic Algorithms, *London England: The MIT Press*, 1998.
  27. Talbi, E. Metaheuristics: From design to implementation. US: Wiley. 2009.
  28. Azuaje, F.: Review of “Artificial immune systems: A new computational intelligence approach. *Journal Neural Networks*. 2003, vol. 16, no.8, Elsevier, pp. 1229–1229.
  29. Molina, J. Problema de Optimización de rutas de vehículos con aspectos medioambientales. *Master's thesis*, Universidad de Sevilla. 2013.
  30. Safaei, N., Banjevic, D., & Jardine, A. K. Multi-objective simulated annealing for a maintenance workforce scheduling problem: a case study. *In Simulated annealing. InTech*. 2008.
  31. Arfken, G. Métodos matemáticos para físicos (No. 510 A7Y 1970). 1981.
  32. Srinivas, N. and Deb, K.: Multi-Objective function optimization using non-dominated sorting genetic algorithms. *Evolutionary Computation*. 1994, 2(3), pp. 221-248.
  33. Deb, K., Agrawal, S., Pratap, A., Meyarivan, T. : A Fast Elitist Non-dominated Sorting Genetic Algorithm for Multi-objective Optimisation: NSGA-II, *Proceedings of the 6th International Conference on Parallel Problem Solving from Nature*. 2000, pp. 849-858.
  34. Cortés, A. Comparación de dos algoritmos metaheurísticos en la solución Multi-objetivo de un modelo particular de la cadena de suministros. *Master's thesis, Instituto Tecnológico de León*. 2010.
  35. Escobar, A. Optimización Multi-objetivo. *Universidad Tecnológica de Pereira – Colombia*. PDF Presentation. 2014.
  36. Arranz de la Peña, J. Algoritmos genéticos. *Madrid: Universidad Carlos III*. 2009.
  37. Srinivas, M. & Patnaik, L. M. Genetic algorithms: A survey. *Computer*. 1994, 27(6), 17-26.
  38. Derrac, J., García, S.: A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence. *In Swarm and Evolutionary Computation*. 2011

Acerca de los autores



Moisés Emmanuel Romo Franco es Ingeniero en Sistemas Computacionales, egresado en 2016 en el Instituto Tecnológico de León. Actualmente se encuentra cursando la Maestría en Ciencias de la Computación, en su segundo año en el Instituto Tecnológico de León. Actualmente se encuentra investigando y aplicando diferentes algoritmos Multiobjetivo para la generación del diseño de horarios. Sus áreas de interés son: Aplicación de Algoritmos Multiobjetivo y Visión por computadora.



Juan Martín Carpio Valadez, obtuvo el grado de Doctor en Ciencias (Óptica) del CIO en 1995. Su experiencia profesional incluye el CIO, ITESM campus León, Universidad Iberoamericana plantel León, y desde 1994 a la fecha el Instituto Tecnológico de León, en donde ocupó el cargo de Jefe del Depto. de Sistemas y Computación de 1999 a 2004, el cargo de Jefe de la División de Estudios de Posgrado e Investigación de 2004 a 2006. Ha participado como responsable y colaborador de varios proyectos de investigación, apoyados por CONCyTEG, COSNET y DGEST. Colaboró en la formación de recursos humanos a través de la dirección tesis de licenciatura (10), maestría (35) y doctorado (3). Actualmente es miembro del Sistema Nacional de Investigadores nivel I. Es miembro del Consejo de Posgrado y miembro del claustro del Doctorado Interinstitucional en Ciencias en Computación. Sus áreas de interés son: sistemas inteligentes, University Timetabling, Metaheurística, Hiperheurísticas.



Lucero de Montserrat Ortiz Aguilar es Ingeniero en Sistemas Computacionales, egresada en 2009 en el Instituto Tecnológico de León. Actualmente se encuentra cursando la Maestría en Ciencias de la Computación, en su segundo año en el Instituto Tecnológico de León. Actualmente se encuentra investigando y aplicando diferentes técnicas Metaheurísticas e Hiperheurísticas para la generación del diseño de horarios. Ha publicado artículos en el Congreso de la Mujer 2015, organizado por el CIO y en COMIA 2015 (Congreso Mexicano de Inteligencia Artificial), en relación a la aplicación de Metaheurísticas al diseño de horarios. Sus áreas de interés son: Técnicas de optimización combinatoria, Metaheurísticas, Hiperheurísticas y Visión por computadora.



Jorge Alberto Soria Alcaraz es egresado como Ingeniero en sistemas computacionales por el instituto tecnológico de león en 2008, continuó su formación como Maestro en ciencias en Ciencias de la Computación por la misma casa de estudios egresando en 2010. Obtuvo el grado de Doctor en ciencias de la Computación por el Instituto Tecnológico de Tijuana B.C parte del Tecnológico Nacional de México en el 2015. El Dr Soria-Alcaraz cuenta trabajos publicados en el área nacional e internacional sobre los temas de Hiper-heurísticas, Timetabling así como Autonomus Search. Ha asistido a congresos nacionales e Internacionales a presentar trabajos acordes a estas áreas. Actualmente se desempeña como profesor investigador de tiempo completo en la División de Ciencias Económico-Administrativas de la Universidad de Guanajuato campus Guanajuato apoyando la carrera de Licenciatura en Sistemas de Información Administrativa. También pertenece al Sistema Nacional de Investigadores (SNI) con la distinción de Candidato a investigador Nacional.



Héctor J. Puga Soberanes se graduó de Licenciatura en Físico Matemáticas, en el Instituto Politécnico Nacional en 1993. Obtuvo el grado de Maestría en Ciencias (Óptica) en 1995, egresado del Centro de Investigaciones en Óptica, A. C. (CIO), obteniendo el título por parte de la Universidad de Guanajuato. Obtuvo el grado de Doctor en Ciencias (Óptica), egresado del CIO, obteniendo el título por parte de la Universidad de Guanajuato en 2002. Cuenta con publicaciones internacionales, en congresos internacionales y nacionales. Ha participado como responsable y colaborador de varios proyectos de investigación, apoyados por CONCyTEG, COSNET y DGEST. A colaborado en la formación de recursos humanos a través de la dirección tesis de licenciatura (3) y Maestría (4) Actualmente es miembro del Sistema Nacional de Investigadores nivel I. Profesor con perfil deseable (Promep) de Agosto de 2005 a la fecha. Sus áreas de interés son: Metrología, Hiperheurísticas y sistemas inteligentes.



Carlos Lino Ramírez profesor Investigador y jefe de la División de Estudios de Posgrado e Investigación (DEPI) del Instituto Tecnológico de León, en el programa de Maestría en Ciencias en Ciencias de la Computación desde Noviembre del 2010. Doctorado en

Arquitectura y Tecnología de los Sistemas Informáticos por la Universidad Politécnica de Valencia, España (2012). Maestro en Ciencias en Ciencias Computacionales por el Instituto Tecnológico de León (1999). Ingeniero en Sistemas Computacionales por el Instituto Tecnológico de León (1996). Ha sido subdirector del Instituto Tecnológico de León (2006-2007), jefe del departamento de Sistemas y Computación del Instituto Tecnológico de León (2004-2006), jefe del área de redes en el Sistema Avanzado de Bachillerato y Educación Superior (1999-2001). Ha presentado sus trabajos de investigación en diversos congresos internacionales en España, Italia, Alemania y México. Sus áreas de investigación son Inteligencia de Ambiente, Algoritmos de Encaminamiento y Redes de Sensores Inalámbricas.



Luis Ernesto Mancilla Espinosa recibió su BS grado en comunicación e ingeniería electrónica de la Escuela Superior de Ingeniería Eléctrica del Instituto Politécnico Nacional en 1975. El obtuvo el grado de Maestro en ciencias de la Computación del Instituto Tecnológico de León en 1999 y su posgrado en ingeniería especializada en Mecatrónica del Centro de Investigación y desarrollo tecnológico en Querétaro, Qro. Sus líneas de interés cubren la Mecatrónica, Inteligencia Artificial, Software de computadora e Ingeniería en Software.