

## Resolución del problema de balanceo de mallas curriculares mediante algoritmo de luciérnagas

*Solving the balanced academic curriculum problem using firefly algorithm*

José Miguel Rubio<sup>1,\*</sup> Ricardo Soto<sup>2</sup> Hernán Jorquera<sup>2</sup>  
Jorge Aguilera<sup>2</sup> Cristian Vidal<sup>3</sup>

Recibido 27 de junio de 2018, aceptado 06 de agosto de 2018  
*Received: June 27, 2018 Accepted: August 06, 2018*

### RESUMEN

En este artículo se propone resolver el problema de balanceo de mallas curriculares (BACP), que consiste en la asignación de una carga académica balanceada a periodos académicos de un plan de estudios, mediante una técnica de optimización basada en la atracción de las luciérnagas (FA). Se resolverán tanto instancias de prueba como instancias reales con el fin de medir el rendimiento del algoritmo y entregar un sistema que permita simplificar el proceso de diseño de una malla curricular en las Instituciones de Educación Superior. Los resultados obtenidos muestran que el algoritmo FA tiene una convergencia bastante rápida y encuentra el óptimo conocido en la mayoría de las pruebas realizadas.

Palabras clave: Problema de balanceo de mallas curriculares, algoritmo luciérnaga, metaheurísticas.

### ABSTRACT

*In this paper, we propose to solve the balanced academic curriculum problem (BACP), which consists of assigning a balanced academic load to academic periods of a curriculum, through an optimization technique based on the attraction of fireflies (FA). Will be solved test instances and real instances in order to measure the performance of the algorithm and to provide a system that simplifies the process of designing a curricular plan in Higher Education Institutions. The obtained results show that the algorithm FA has a fast convergence and finds the optimum known in most of the tests performed.*

*Keywords: Balanced academic curriculum problem, firefly algorithm, metaheuristic.*

### INTRODUCCIÓN

Cuando se diseñan mallas curriculares para un programa de estudios en educación superior se consideran factores tales como; la cantidad de asignaturas que posee la carrera, el número de periodos en el que se deben asignar estos cursos, la carga académica mínima y máxima aceptable, y el número de cursos

permitido por semestre. Con toda esta información se conforman las restricciones o regulaciones académicas relacionadas al plan de estudios, las cuales son usadas para su construcción mediante un enfoque de prueba y error hasta lograr una malla adecuada.

Si se considera que el grado de esfuerzo que se requiere para aprobar una asignatura se puede

---

<sup>1</sup> Escuela de Ingeniería y Tecnología. Universidad Tecnológica de Chile INACAP. Av. Vitacura 10151. Santiago, Chile. E-mail: jrubiol@inacap.cl

<sup>2</sup> Escuela de Ingeniería Informática. Pontificia Universidad Católica de Valparaíso. Av. Brasil 2950. Valparaíso, Chile. E-mail: ricardo.soto@pucv.cl; jorge.aguilera.m@mail.pucv.cl; hernan.jorquera.m@mail.pucv.cl

<sup>3</sup> Ingeniería Civil Informática. Escuela de Ingeniería. Universidad Viña del Mar. Agua Santa 7055, sector Rodelillo, Viña del Mar, Chile. E-mail: cristian.vidal@uvm.cl

\* Autor de correspondencia.

medir en créditos, el éxito que puedan tener los estudiantes tiene directa relación con la carga académica que enfrenten en cada periodo. La carga académica corresponde al número de créditos por semestre. Es por esta razón que las mallas curriculares deben estar “balanceadas”, es decir, la cantidad de créditos para cada periodo debe ser similar para que la carga que enfrenten los alumnos sea la mínima posible. Por lo tanto, es de interés minimizar este costo diseñando un plan de estudios mediante un algoritmo que realice este esfuerzo de manera automática y sin errores.

Este problema es conocido en la literatura como el problema de balanceo de mallas curriculares (BACP), y es del tipo CSP (Problema de Satisfacción de Restricciones); en este tipo de problemas se busca satisfacer todas las restricciones asociadas y luego optimizar la calidad de la solución encontrada. Se han estudiado diversos modelos que resuelven el BACP, donde generalmente este problema ha sido abordado utilizando el paradigma de programación con restricciones y algoritmos híbridos utilizando algoritmos genéticos, esquemas de colaboración y búsquedas locales, entre otras técnicas.

El presente trabajo se centra en resolver tanto las instancias de prueba, reconocidas por la CSPLib [1], como instancias reales del problema. Las instancias reales consideradas consisten en programas de estudio del área informática de dos universidades chilenas.

Se utilizará la metaheurística de optimización basada en el comportamiento de las luciérnagas o *Firefly Algorithm* (FA), propuesto por Xin-She Yan [2], para resolver el BACP. En la presente investigación, el algoritmo FA se aplica a un conjunto de soluciones encontradas previamente mediante el método de programación lineal entera y representadas en una matriz binaria. Luego de obtener un conjunto de soluciones válidas, considerando que cada una se representa con una luciérnaga, se procede a optimizar el espacio de soluciones iniciales mediante FA para poder encontrar una solución óptima.

El resto del artículo está organizado como sigue: en la sección 2 se presenta el desarrollo del análisis teórico del problema. Luego, los principales resultados computacionales obtenidos y, finalmente las conclusiones del trabajo realizado.

## ANÁLISIS TEÓRICO

### Formulación del problema

El BACP fue originalmente introducido y desarrollado por Castro y Manzano en [3], ellos plantearon un modelo de programación lineal entera que considera las siguientes entidades y restricciones:

**Cursos:** Los cursos poseen créditos asignados y deben ser dictados dentro del plan de estudios.

**Periodos:** La malla curricular compone un plan de estudios que se divide en un número fijo de intervalos de tiempo (periodos académicos). Cada periodo es lectivo lo cual permite que se dicten cursos. Por ejemplo, una malla curricular que se divide en 8 periodos académicos, dura 4 años y cada año se divide en 2 periodos (semestres).

**Máxima carga:** Por cada periodo hay un máximo de carga académica permitida.

**Mínima carga:** Por cada periodo hay un mínimo de carga académica permitida.

**Prerrequisitos:** Dentro de la malla curricular, el plan de estudios contiene algunos cursos que deben ser dictados antes que otros, estos cursos se llaman prerrequisitos. Esto genera que se formen pares ordenados de cursos, en los cuales se debe cumplir la restricción de que el estudiante debe aprobar el primer curso antes de cursar el segundo.

**Distribución balanceada de la carga:** La malla curricular debe quedar balanceada, es decir el número de créditos de cada periodo académico deber ser similar, idealmente igual.

Considerando estas entidades y restricciones, el objetivo del problema de investigación es encontrar una asignación de cursos para cada periodo que satisfaga todas las restricciones mencionadas de forma óptima.

Para validar el algoritmo propuesto se utilizarán las instancias de prueba de la CSPLib (BACP8, BACP10 y BACP12). Por ejemplo, si se considera el BACP8, este plan de estudios está conformado por 46 cursos con un total de 133 créditos a impartir en 8 períodos académicos. En consecuencia, el promedio aritmético simple de créditos por período

es equivalente a  $133/8 = 16,625$ , lo que implica que el límite inferior del número máximo de créditos por período es 17. Por lo tanto, soluciones con valor 17 son óptimas.

**Modelo matemático**

Se propone un modelo de programación lineal entera basado en [3]. Este modelo plantea una variable de decisión de una dimensión para la resolución del problema y considera los siguientes parámetros:

- m: Número de cursos.
- n: Número de periodos académicos.
- $a_i$ : Número de créditos de curso  $i$ , donde  $i=1, \dots, m$
- $\beta$ : Carga académica mínima por periodo.
- $\gamma$ : Carga académica máxima por periodo.
- $\delta$ : Cantidad mínima de cursos por periodo.
- $\varepsilon$ : Cantidad máxima de cursos por periodo.

**Variables de decisión:**

Vector con los periodos que cada curso tiene asignado:

$$x_i = j, \forall i = 1, \dots, m \tag{1}$$

$c$ : es la máxima carga académica para todos los periodos:

$$c = \max\{c_1, \dots, c_n\} \tag{2}$$

La carga académica del periodo  $j$  está definida por:

$$c_j = \sum_{i=1}^m \alpha_i \xi_i, \forall i = 1, \dots, m, \forall j = 1, \dots, n \tag{3}$$

$$\xi_i = \begin{cases} 1 & \text{si } x_i = j \\ 0 & \text{si } x_i \neq j \end{cases}$$

**Función Objetivo:**

$$\min c$$

**Restricciones:**

Todos los cursos  $i$  deben ser asignados a un periodo  $j$ :

$$\sum_{j=1}^n x_{i,j} = 1, \forall i = 1, \dots, m \tag{4}$$

El curso  $b$  tiene un curso  $a$  como prerrequisito:

$$x_{b,j} \leq \sum_{r=1}^{j-1} x_{a,r}, \forall j = 2, \dots, n \tag{5}$$

La carga académica máxima está definida por la ec. (2). Lo que puede ser representado por el siguiente conjunto de restricciones lineales:

$$c_j \leq c, \forall j = 1, \dots, n \tag{6}$$

La carga académica del periodo  $j$  debe ser mayor o igual que el mínimo requerido:

$$c_j \geq \beta, \forall j = 1, \dots, n \tag{7}$$

La carga académica del periodo  $j$  debe ser menor o igual que el máximo requerido:

$$c_j \leq \gamma, \forall j = 1, \dots, n \tag{8}$$

El número de cursos del periodo  $j$  debe ser mayor o igual que el mínimo requerido:

$$\sum_{i=1}^m \xi_i \geq \delta, \forall j = 1, \dots, n \tag{9}$$

$$\xi_i = \begin{cases} 1 & \text{si } x_i = j \\ 0 & \text{si } x_i \neq j \end{cases}$$

El número de cursos del periodo  $j$  debe ser menor o igual que el máximo requerido:

$$\sum_{i=1}^m \xi_i \leq \varepsilon, \forall j = 1, \dots, n \tag{10}$$

$$\xi_i = \begin{cases} 1 & \text{si } x_i = j \\ 0 & \text{si } x_i \neq j \end{cases}$$

**Optimización basada en las Luciérnagas**

La optimización basada en las luciérnagas es una de las heurísticas más novedosas inspirada en los comportamientos naturales para problemas de optimización. Es comúnmente sabido que las luciérnagas poseen un brillo característico inconfundible, e incluso las personas que no han visto una en su vida saben que estas emanan una luz.

El fundamento del algoritmo está orientado al brillo de estas criaturas que, en su necesidad de aparearse, se acercan a cierta luciérnaga la cual libera una luz que es atractiva para las demás. Es este

comportamiento el que ha dado origen al algoritmo de optimización, el cual considera como función objetivo la intensidad del brillo de las luciérnagas y su necesidad de acercarse a la luciérnaga más brillante u óptima.

**Las luciérnagas en la naturaleza:** Existen diversos tipos de insectos y animales en la naturaleza que, dependiendo de cada especie, presentan organizaciones y habilidades que hacen que cada elemento de esa colonia cumpla un objetivo para un bien común. Esto se logra a través de diversas formas de comunicación y habilidades.

Un ejemplo de comunicación es la de la luciérnaga, la que combina el oxígeno que absorbe dentro de unas células especiales con una sustancia llamada luciferina, y reacciona produciendo luz sin apenas generar calor. La luz de la luciérnaga es por lo general intermitente, y brilla de un modo específico en cada especie. Cada forma de brillar es una señal óptica que ayuda a las luciérnagas a encontrar posibles parejas.

**Del comportamiento hacia la metaheurística:** FA está especialmente recomendado para problemas de optimización multimodal. De hecho, FA se ha aplicado en optimización continua, problema del viajero (TSP), segmentación (*clustering*), procesamiento de imágenes y selección de características [4].

A continuación, se identifican tres propiedades importantes del comportamiento de las luciérnagas [5]:

Todas las luciérnagas son unisexuales y se sienten atraídas por otras luciérnagas, independientemente de su sexo.

El brillo de una luciérnaga se determina por el valor de la función objetivo. Para un problema de maximización, el brillo de cada luciérnaga es proporcional al valor de la función objetivo y viceversa.

El grado de atracción de una luciérnaga es proporcional a su brillo, y por lo tanto para cualquier par de luciérnagas parpadeantes, la que es menos brillante se moverá hacia la más brillante. Más brillo significa menos distancia entre dos luciérnagas. Sin embargo, si las dos luciérnagas parpadeantes tienen el mismo brillo, éstas se mueven aleatoriamente.

Del análisis de estas propiedades se obtienen las fórmulas y procedimientos para el funcionamiento de la metaheurística FA, el cual se explica en detalle en la siguiente sección.

**Modelos de optimización basados en el comportamiento de las luciérnagas:** El algoritmo FA tiene muchas variantes, que han sido usadas en casi todas las áreas de la ciencia, existiendo más de 20 de ellas [6]. A continuación, se explican brevemente algunas:

*Discrete FA (DFA):* Este algoritmo puede ser aplicado directamente para resolver problemas de optimización discreta [7-9].

*Multiobjective FA:* Este método tiene como propósito la resolución de problemas con múltiples objetivos, es decir, opera con varias funciones objetivo, para esto combina todas las funciones objetivo en una sola, así el algoritmo no cambia de gran manera [10].

*Lagrangian FA:* Este algoritmo es propuesto para solucionar el problema de optimización de compromiso de unidades en sistemas de poder no regulados [11].

*Chaotic FA:* Este algoritmo busca solucionar problemas relacionados con sistemas que se comportan de manera impredecible, esto no quiere decir que sean sistemas con una completa ausencia de orden, sino sistemas que tienen orden perfecto, pero incluyen factores al azar [12].

*Hybrid Algorithms:* Esta es una combinación entre el algoritmo de luciérnaga y el algoritmo de hormiga [13].

*Parallel FA with predation (pFAP):* Esta es una implementación para entornos con memoria compartida con un mecanismo de depredación agregado que ayuda al método a escapar del óptimo local [14].

*Modified FA:* Este es un mecanismo usado por distintos autores que buscan optimizar el rendimiento del algoritmo, estos le agregaron sus propios métodos o variantes al algoritmo, donde incluso se puede cambiar el proceso de actualización para mantener el mejor resultado a través de las iteraciones [15].

**Algoritmo FA**

Para aplicar el algoritmo FA en los problemas de optimización, el brillo es proporcional al valor de la función objetivo. En el caso de algoritmos genéticos el brillo podría ser definido en el mismo sentido que la función objetivo [16].

El atractivo de una luciérnaga es proporcional a la luz que emana, y ven a su vez otras luciérnagas, este atractivo puede ser definido como:

$$\beta = \beta_0 e^{-\gamma r^2} \tag{11}$$

Donde  $\beta_0$  es el atractivo a una distancia  $r = 0$ . La distancia entre dos luciérnagas  $\tau_{ij}$  es calculada usando el método de distancia cartesiana y  $\gamma$  representa el coeficiente de absorción de luz y está asociado a la escala y naturaleza del problema.

Una luciérnaga  $i$  es atraída a una luciérnaga más brillante  $j$  y su movimiento está determinado por:

$$x_i = x_i + \beta_0 e^{-\gamma r^2} (x_j - x_i) + \alpha \left( rand - \frac{1}{2} \right) \tag{12}$$

Donde  $x_i$  y  $x_j$  es la posición actual de las luciérnagas, el segundo término corresponde a la atracción y el tercer término introduce una componente aleatoria, siendo  $\alpha$  un parámetro de asignación al azar y  $rand$

un número aleatorio generado uniformemente distribuido entre 0 y 1.

La capacidad del algoritmo FA para resolver problemas de optimización ha sido demostrada por distintos autores. Por otro lado, en [5] se ha demostrado la eficacia de este algoritmo para resolver problemas con representación binaria.

La principal idea para resolver el BACP propuesta en este trabajo se basa en representar el problema mediante una disposición binaria, debido a que se ha comprobado que plantear la solución de esta manera es efectivo y se puede decir con seguridad que el algoritmo de luciérnagas se adapta perfectamente a una representación binaria como la que se plantea más adelante en este artículo.

En la Figura 1 se observa el esquema general del algoritmo FA; el objetivo del problema está definido por la función objetivo, además se deben inicializar los parámetros de la luciérnaga ( $\gamma$ ,  $\beta_0$ , el tamaño de la población de luciérnagas:  $n$  y el número máximo de generaciones: *MaxGeneration*).

**Solución Propuesta**

Se propone aplicar el algoritmo FA para resolver el modelo de programación lineal entera del BACP presentado anteriormente. La solución desarrollada en este artículo se obtendrá mediante tres simples pasos que se muestran en la Figura 2; primero:

---

**Algoritmo 1 – Pseudocódigo del Algoritmo FA**

---

```

Función objetivo  $f(x), x = (x_1, \dots, x_d)^T$ 
Generar una población inicial de  $n$  luciérnagas  $x_i (i = 1, 2, \dots, n)$ 
Intensidad de luz  $I_i$  en  $x_i$  es determinada por  $f(x_i)$ .
Definir el coeficiente de absorción de luz  $\gamma$ 
while( $t < MaxGeneration$ ).
  for  $i = 1:n$  (Todas las  $n$  luciérnagas)
    for  $j = 1:n$  (Todas las  $n$  luciérnagas) (loop interno)
      if ( $I_i < I_j$ )
        Mover luciérnaga  $i$  hacia luciérnaga  $j$ 
      end if
      Variar atractivo con la distancia  $r$  via  $exp| -\gamma r^2|$ 
      Evaluar nueva solución y actualizar intensidad de luz
    end for  $j$ 
  end for  $i$ 
  Clasificar las luciérnagas y encontrar el mejor optimo global  $g$ .
end while
Procesar y visualizar
    
```

---

Figura 1. Esquema general del algoritmo FA.

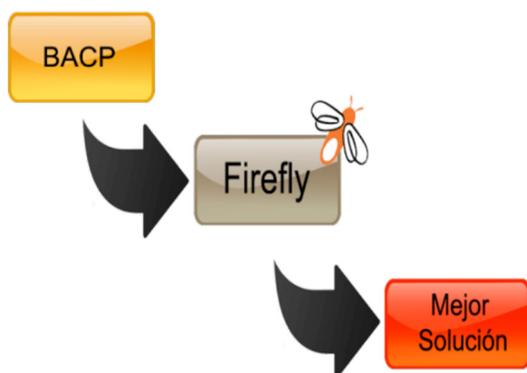


Figura 2. Esquema de solución propuesto.

se carga una instancia del problema, segundo: se aplica el algoritmo FA en una representación binaria del problema y finalmente se obtiene una representación de la mejor solución encontrada por el algoritmo FA.

Cada instancia del problema está definida por el conjunto de restricciones, parámetros y variables decisionales mencionadas al comienzo de esta sección. Los cursos y periodos académicos forman una malla curricular (o plan de estudios), que se puede representar mediante una matriz binaria sobre la cual se aplicará el algoritmo FA.

Además de las características propias del problema, se deben realizar los siguientes pasos: (1) definir el coeficiente de absorción, (2) generar una población inicial de luciérnagas, (3) definir el coeficiente de atractivo y (4) determinar el número máximo de generaciones.

### Representación de la Solución

Es sabido que al momento de buscar la solución a un problema un aspecto importante es buscar una representación simple, lógica y útil que sea capaz de resolver la problemática de manera eficiente y eficaz.

Para resolver el BACP se ha planteado una representación matricial binaria  $m \times n$ , es decir, filas con periodos y columnas con cursos, el propósito es que la matriz posea casillas con valores posibles 0 o 1. Esta representación indicara si un curso  $i$  es asignado a un periodo  $j$  (valor 1), o por el contrario que el curso  $i$  no este asignado a ese periodo  $j$  (valor 0). En la Figura 3, se muestra la representación base para una posible solución al problema.

		Períodos							
CURSOS	1	1	0	0	0	0	0	0	0
	2	0	0	0	1	0	0	0	0
	3	0	0	0	0	0	0	0	1
	4	0	0	1	0	0	0	0	0
	5	0	0	0	0	1	0	0	0
	6	0	1	0	0	0	0	0	0
	7	0	0	0	0	0	0	0	0
	8	0	0	0	0	0	0	0	1

Figura 3. Representación matricial de una solución.

## RESULTADOS

La solución propuesta fue implementada en lenguaje de programación Java utilizando el entorno de programación Netbeans IDE 8.0. Además, para la ejecución de las pruebas se usó un computador PC compatible con sistema operativo Windows 7 Ultimate 64 bits y procesador Phenom II X4 de 3.4 GHZ y 8 GB de RAM.

### Modificaciones al algoritmo FA

Se debe modificar el algoritmo FA para que pueda operar con representaciones binarias. Al aplicar la fórmula de movimiento a una luciérnaga, el algoritmo genera valores que no cumplen con las condiciones definidas en la matriz binaria propuesta para representar la solución.

Lo anterior se debe a que al aplicar la fórmula de movimiento en cada una de las dimensiones de la matriz se obtiene un valor real, el cual debe ser transformado a binario mediante una función de transferencia.

Una función de transferencia es un modelo matemático que a través de un cociente relaciona la respuesta de un sistema (modelada) con una señal de entrada o excitación (también modelada).

En teoría de control, a menudo se usan las funciones de transferencia para caracterizar las relaciones de entrada y salida de componentes o de sistemas que se describen mediante ecuaciones diferenciales lineales e invariantes en el tiempo.

En este trabajo se utiliza la siguiente función de transferencia [5]:

$$\text{Tanh}\left(|X_p|\right) = \frac{\exp\left(2|X_p|-1\right)}{\exp\left(2|X_p|+1\right)} \quad (12)$$

Para pasar los valores desde un espacio de búsqueda continuo a uno discreto, se sigue la misma la regla utilizada en [5]:

$$x_i^k(t+1) = \begin{cases} 1 & \text{si } \text{rand} < T\left(x_i^k(t+1)\right) \\ 0 & \text{en otro caso} \end{cases} \quad (13)$$

Donde *rand* es un número aleatorio entre 0 y 1 distribuido uniformemente, *xikt* es el valor de la dimensión *k* de una luciérnaga *i* de la iteración.

### Experimentos computacionales

Para validar los resultados del sistema, y su desempeño al momento de generar soluciones, se realizaron 50 pruebas a cada una de las instancias disponibles en la CSPLib. Los parámetros utilizados para resolver estas instancias de prueba son los que se presentan en la Tabla 1.

Las instancias de prueba son problemas propuestos que sirven para medir el comportamiento del algoritmo y además son la base de comparación con el trabajo de otros autores que han resuelto el mismo problema. Para el problema BACP existen 3 instancias de prueba las cuales son analizadas y resueltas de forma independiente.

Tabla 1. Parámetros utilizados en experimentos.

Parámetro	Valor
Número de iteraciones	1000
Número de luciérnagas	30
$\alpha$	0.5
$\beta$	1
$\gamma$	1

**BACP8:** El BACP8 corresponde a la instancia más pequeña del problema, con 46 asignaturas que deben ser distribuidas en 8 periodos académicos. En la Figura 4 se puede observar cómo se distribuyen los resultados obtenidos.

El gráfico de convergencia del algoritmo (ver Figura 5) muestra como el sistema encuentra una solución óptima inicial de 18 créditos y cambia

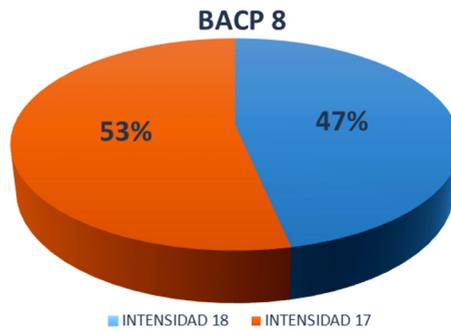


Figura 4. Resultados BACP8.

rápidamente a 17 en las primeras iteraciones, manteniéndose en ese óptimo hasta finalizar la ejecución del algoritmo.

**BACP10:** Esta instancia está compuesta de 42 asignaturas, las que deben ser asignadas en 10 periodos académicos, por lo que la carga académica máxima es menor que en el caso del BACP8. En esta instancia de prueba, el sistema tuvo mayor dificultad para encontrar el óptimo conocido. En la Figura 6 se muestran los resultados obtenidos.



Figura 5. Convergencia BACP8.

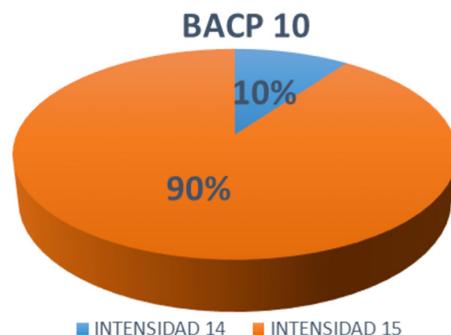


Figura 6. Resultados BACP10.

En la Figura 7 se puede observar la convergencia del algoritmo para una ejecución sobre la instancia BACP10. Los resultados obtenidos muestran que para esta instancia de prueba el valor óptimo encontrado por el sistema varía desde 16 en las primeras 2 iteraciones a 15 en la iteración 3 y finalmente llega al valor óptimo conocido 14 en la iteración 41.

**BACP12:** La instancia de 12 periodos es la más compleja y cuenta con 66 asignaturas a asignar, por lo que el esfuerzo computacional requerido para encontrar buenas soluciones es mayor que en los casos anteriores. Para esta instancia del problema el sistema logró encontrar la solución óptima conocida sólo en el 23% de las ejecuciones. La distribución de los resultados obtenidos se puede apreciar en la Figura 8.



Figura 7. Convergencia BACP10.

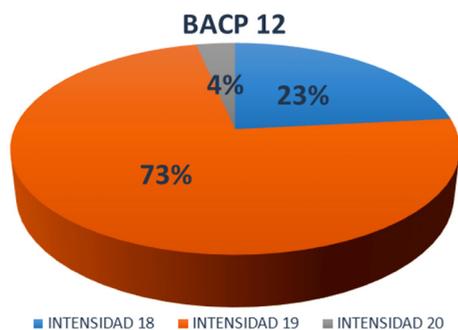


Figura 8. Resultados BACP12.

En la Figura 9 se muestra como el algoritmo converge durante una ejecución para la instancia BACP12, encontrando una solución de 20 créditos en las iteraciones iniciales, luego 19 y convergiendo finalmente a 18 créditos en la iteración 13.

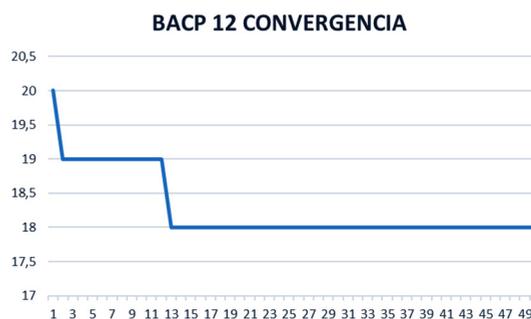


Figura 9. Convergencia BACP12.

### Instancias Reales

En esta sección se muestran los resultados obtenidos al aplicar el sistema desarrollado en la optimización de distintos planes de estudios reales de 2 universidades chilenas. Las instancias se han denominado REAL8 (8 periodos), REAL10 (10 periodos) y REAL12 (12 periodos).

**REAL8:** Esta instancia cuenta con 34 asignaturas que deben ser planificadas en 8 periodos. Además, 27 de estas asignaturas son obligatorias, 2 asignaturas son de estudio general y 5 asignaturas son electivas. Para esta instancia se logró encontrar un valor óptimo de 17 créditos, mientras que en el plan de estudios actual el valor es de 18 créditos. Estos resultados se pueden observar en la Figura 10.

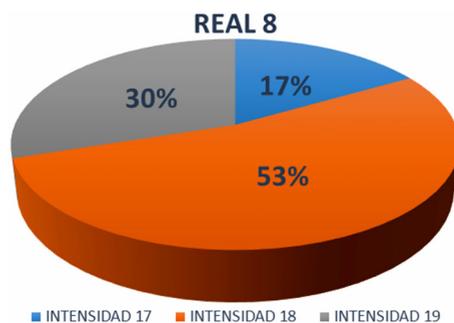


Figura 10. Resultados REAL8.

En el gráfico de convergencia de la Figura 11, se muestra que el sistema converge desde 22 créditos a 17 créditos rápidamente.

**REAL10:** La segunda instancia real resuelta corresponde a un plan de estudios que tiene una duración de 5 años y en el cual se deben planificar 49 asignaturas en 10



Figura 11. Convergencia REAL8.

periodos académicos. En esta instancia se obtuvo un valor óptimo de 13, el que coincide con el valor actual del plan de estudios. Los resultados obtenidos se pueden apreciar en la Figura 12.

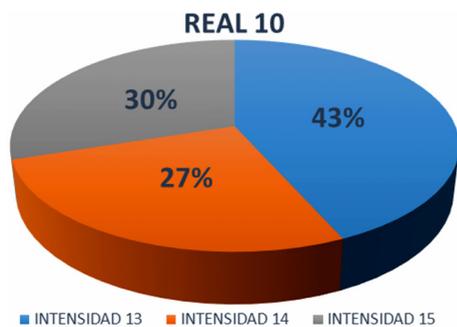


Figura 12. Resultados REAL10.

En la Figura 13 se observa que el comportamiento del algoritmo es similar a las instancias anteriores resueltas, mostrando una rápida convergencia hacia el óptimo conocido para el problema.

**REAL12:** La última instancia real del problema es la que presenta mayor complejidad y está compuesta



Figura 13. Convergencia REAL10.

por 53 asignaturas, de las cuales 49 son obligatorias y 4 electivas, que deben ser asignadas a 12 periodos. Para esta instancia en el 3% de las ejecuciones se obtuvo un valor óptimo de 17 créditos, lo que mejora el óptimo del plan actual cuyo valor es 18. La distribución de los resultados obtenidos para esta instancia se muestra en la Figura 14.

En la Figura 15 se muestra como el sistema converge al óptimo en la iteración 15. Nuevamente queda demostrada la rápida convergencia del algoritmo propuesto, tanto al resolver las instancias de prueba como las instancias reales del problema.

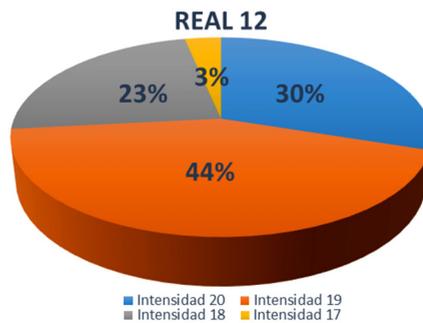


Figura 14. Resultados REAL12.



Figura 15. Convergencia REAL12.

### Síntesis de Resultados

En términos generales, los resultados obtenidos por el sistema se consideran alentadores. Estos resultados se resumen en la Tabla 2.

En la tabla resumen se puede apreciar que en la totalidad de las instancias resueltas se logra obtener el óptimo conocido para el problema e incluso mejorar dicho óptimo en algunos casos (instancias REAL8 y REAL12). También se observa una alta

Tabla 2. Síntesis de Resultados.

Solución	BACP8	BACP10	BACP12	REAL8	REAL10	REAL12
Mejor	17	14	18	17	13	17
Media	17.5	14.9	18.8	18.1	13.9	19.0
Peor	18	15	20	19	15	20
$\sigma$	.509	.305	.484	.681	.860	.831
Optimo	17	14	18	18	13	18

dispersión, la que queda expresada en el valor de la desviación estándar de las soluciones encontradas para las instancias REAL10 y REAL12, esto se puede producir debido a que estas instancias presentan una mayor complejidad de resolución.

### CONCLUSIONES

En este trabajo se propuso un algoritmo basado en el comportamiento de las luciérnagas naturales para resolver el problema de balanceo de mallas curriculares. La evaluación experimental muestra la efectividad de las luciérnagas artificiales para resolver este tipo de problemas.

Todas las pruebas realizadas muestran que es posible encontrar buenas soluciones en la mayoría de las ejecuciones del algoritmo propuesto. En las instancias de prueba la calidad de las soluciones es satisfactoria y se obtiene el valor óptimo conocido de cada instancia en la mayoría de los casos. En instancias reales del problema se obtiene el valor óptimo conocido e incluso es posible mejorar dicho valor en 2 de las 3 instancias que comprende esta investigación.

Será interesante investigar a futuro cómo se comporta este algoritmo enfrentado a otras instancias reales del problema y evaluar otras variantes del algoritmo.

También es de interés de los autores evaluar el comportamiento del sistema propuesto con instancias de la versión generalizada del problema, conocida como GBACP. Estas instancias tienen una mayor

complejidad que el problema original y han sido propuestas inicialmente en [17].

### REFERENCIAS

- [1] Ian P. Gent and Toby Walsh. "CSplib: A Benchmark Library for Constraints. In: Jaffar J. (eds) Principles and Practice of Constraint Programming - CP'99. CP 1999". Lecture Notes in Computer Science. Vol. 1713. Springer, Berlin, Heidelberg. 1999.
- [2] X. Yang. Nature-Inspired Metaheuristic Algorithms. Frome: Luniver Press. 2008.
- [3] C. Castro and S. Manzano. "Variable and Value Ordering When Solving Balanced Academic Curriculum Problems". Proceedings ERCIM WG on constraints, pp. 3-4. Octubre 2, Chile. 2001.
- [4] K.N Krishnanand and D. Ghose. "Glowworm swarm optimization for simultaneous capture of multiple local optima of multimodal functions". pp. 87-124. Swarm Intelligence. 2009.
- [5] B. Crawford, R. Soto, M. Olivares-Suárez and F. Paredes. "A Binary FA Algorithm for the Set Covering Problem". Modern Trends and Techniques in Computer Science". pp. 65-73. ISBN 978-3-319-06739-1. 2014.
- [6] Ngaam J. Cheung, Xue Ming Ding and Hong-Bin Shen. "Adaptive Firefly Algorithm: Parameter Analysis and its Application". PLOS ONE 9 (11): e112634. DOI:10.1371/journal.pone.0112634. 2014.
- [7] M. K. Sayadi, R. Ramezani and N. Ghaffari-Nasab. "A discrete firefly metaheuristic with local search for makespan minimization in permutation flow shop scheduling problems". Int. J. of Industrial Engineering Computations. 1: 1-10. DOI: 10.5267/j.ijiec.2010.01.001. 2010.
- [8] T. Hassanzadeh, H. Vojodi and A.M.E. Moghadam. "An image segmentation approach based on maximum variance intra-cluster method and firefly algorithm". Vol. 3. 2011.
- [9] K. Durkota. "Implementation of a discrete firefly algorithm for the QAP problem within the sage framework". BSc thesis, Czech Technical University. 2011. <http://cyber.felk.cvut.cz/research/theses/papers/189.pdf>
- [10] T. Apostolopoulos and A. Vlachos. "Application of the Firefly Algorithm for Solving the Economic Emissions Load Dispatch

- Problem”. *International Journal of Combinatorics*. Volume 2011. Article ID 523806. 2011.
- [11] B. Rampriya, K. Mahadevan and S. Kannan. “Unit commitment in deregulated power system using Lagrangian firefly algorithm. *International Conference on communication control and computing technologies*. 2010.
- [12] L. dos Santos Coelho, D.L. de Andrade Bernert and V.C. Mariani. “A chaotic firefly algorithm applied to reliability-redundancy optimization”. *IEEE Congress on Evolutionary Computation (CEC’11)*, pp. 517-521. 2011.
- [13] G. Giannakouris, V. Vassiliadis and G. Dou-nias. “Experimental study on a hybrid nature-inspired algorithm for financial portfolio optimization”. *SETN 6040*: 101-111. DOI:10.1007/978-3-642-12842-4\_14. 2010.
- [14] Luz E.F.P., H.F. Campos Velho and J.C. Becceneri. “Firefly Algorithm with Predation: A parallel implementation applied to inverse heat conduction problem, in: *Proc. of 10<sup>th</sup>*. *World Congress on Computational Mechanics (WCCM 2012)*. 2012.
- [15] S.L. Tilahun and H.C Ong. “Modified Firefly Algorithm”. *Journal of Applied Mathematics*. Volume 2012, Article ID 467631. URL: <http://dx.doi.org/10.1155/2012/467631>.
- [16] Sankalap. “Arora and Satvir Singh. Article: The FA Optimization Algorithm: Convergence Analysis and Parameter Selection”. *International Journal of Computer Applications* 69(3): 48-52. May 2013.
- [17] M. Chiarandini, L. Di Gaspero, S. Gualandi and A. Schaerf. “The balanced academic curriculum problem revisited”, *Journal of Heuristics*, Vol. 18 N° 1, pp. 119-148. 2012.