

ARTÍCULO ORIGINAL

Nodos *Knime* para ajustar modelos usando la biblioteca de clases *Biciam*

*Knime Nodes for Model Adjustment
by using Biciam*

Taday González Chaveco

tadayglezchav@gmail.com • <http://orcid.org/0000-0001-8971-6379>

Cynthia Porras Nodarse

cporrasn@ceis.cujae.edu.cu • <http://orcid.org/0000-0002-4557-1961>

Alejandro Rosete Suárez

rosete@ceis.cujae.edu.cu • <http://orcid.org/0000-0002-4579-3556>

UNIVERSIDAD TECNOLÓGICA DE LA HABANA "JOSÉ ANTONIO ECHEVERRÍA", CUJAE, CUBA

Recibido: 2020-11-16 • Aceptado: 2021-01-19

RESUMEN

En muchos problemas de la vida se conoce la forma de un modelo matemático que lo describe, por medio de expresiones con parámetros que deben ser ajustados a cada situación concreta. El ajuste de esos modelos muchas veces implica la solución de un problema de optimización, que busca los valores de los parámetros que minimizan la diferencia entre lo que indica el modelo y los datos reales. Estos problemas se pueden resolver a través del uso de metaheurísticas, lo cual, por lo general, requiere de habilidades en programación. BiCIAM es una biblioteca de clases que resuelve problemas de optimización utilizando metaheurísticas, pero solo personas con conocimientos de programación pueden hacer uso de ella. En el presente trabajo se propone la implementación de nuevos nodos en *Knime* que optimicen y ajusten modelos matemáticos, mediante el uso de BiCIAM. Con estos nodos se podrán resolver modelos de optimización sin necesidad de programar; se podrán ajustar modelos no lineales de carácter predictivo y los datos podrán ser analizados con el empleo de los nodos estadísticos que brinda *Knime*.

PALABRAS CLAVE: BiCIAM; interfaz gráfica, *Knime*; optimización de modelos.

ABSTRACT

Many problems in real life can be expressed using a mathematical model to describe them, through expressions with parameters that must be adjusted to each specific situation. The adjustment of these models often implies the solution of an optimization problem that looks for the values of the parameters in order to minimize the difference between what the model indicates and the real data. Metaheuristic can be used to solve optimization problems, but this generally requires programming skills. BiCIAM is a class library that solves optimization problems using metaheuristics, but only people with programming skills can make use of it. In this work we propose the implementation of new nodes in Knime that optimize and adjust mathematical models, using BiCIAM. With these nodes, optimization models can be solved without the need for programming, non-linear predictive models can be adjusted and the data can be analyzed using the statistical nodes provided by Knime.

KEYWORDS: *BiCIAM; graphic interface, Knime; model optimization.*

INTRODUCCIÓN

La optimización consiste en hallar la mejor solución a un problema determinado ya sea con el propósito de maximizar o minimizar una función objetivo. En la actualidad los problemas que deben ser resueltos tienen tanta complejidad que los métodos exactos no logran resolverlos en un tiempo polinomial. Esto ha llevado al desarrollo de los algoritmos metaheurísticos que son capaces de encontrar muy buenas soluciones en un tiempo razonable (Talbi, 2009).

Una alternativa para resolver problemas de optimización lo constituye el uso de bibliotecas de algoritmos metaheurísticos. En la Facultad de Ingeniería Informática de la Universidad Tecnológica de La Habana “José Antonio Echeverría” (Cujae) se implementó una biblioteca llamada BiCIAM, la cual posee variedad de algoritmos metaheurísticos y ha sido utilizada para resolver gran cantidad de problemas. Para usarla se requiere dominio del lenguaje de programación Java, lo cual que limita su utilización (Fajardo, 2015). Existe una herramienta llamada *Knime (Konstanz Information Miner)* que es una plataforma de exploración de datos basada en el diseño de flujos de trabajo (*workflows*). Posee un ambiente de trabajo visual e interactivo que permite al usuario explorar visualmente los resultados. Además, tiene integrada herramientas que permiten realizar experimentación y análisis estadístico de los datos (Berthold, *et al.*, 2009).

Se conocen distintos modelos que permiten presentar la realidad en términos matemáticos. Estos tienen una forma genérica expresada en términos de parámetros que deben ajust-

tarse para modelar problemas concretos. En este sentido, se procede a un proceso de ajuste de parámetros del modelo. Este procedimiento se puede ver como un problema de optimización en sí, donde los parámetros de las ecuaciones son las variables del problema y la función objetivo a minimizar es el error entre el valor que predice el modelo y la situación real medida (Chapra, *et al.*, 2007; López Jiménez, *et al.*, 2003; Walpole, *et al.*, 2012).

En el presente trabajo se propone desarrollar nuevos nodos en Knime que permitan la optimización del ajuste de funciones usando BiCIAM. De esta manera, expertos en temas concretos que dispongan de modelos matemáticos ajustables podrán usar el componente desarrollado para ajustar esos modelos a situaciones concretas de las que dispongan datos. Esto puede aumentar la cantidad de usuarios potenciales que podrían estar empleando este tipo de modelos para diversas situaciones.

El trabajo está estructurado de la siguiente forma: en la sección *Metodología* se realiza una descripción del ajuste de los modelos matemáticos como problema de optimización, así como la manera de resolver estos problemas usando BiCIAM y la propuesta de creación de un nodo en *Knime* con este fin. En la sección *Resultados y discusión* se exponen los resultados del trabajo a través de la descripción de un caso de estudio donde se ajusta un modelo matemático que predice el comportamiento de la pandemia COVID-19. Finalmente se exponen las *Conclusiones* del trabajo.

METODOLOGÍA

AJUSTE DE MODELOS COMO PROBLEMA DE OPTIMIZACIÓN

Un modelo matemático es una abstracción selectiva de la realidad, expresadas en términos de símbolos y expresiones matemáticas que juntas captan la esencia de una situación de interés. En un modelo matemático, la idea es introducir alguna información en el modelo, que se deduce de la situación modelada y hacer observaciones de los resultados. El modelo mismo es una combinación de diferentes tipos de funciones matemáticas que representan las dependencias entre las entradas y salidas (Biembengut y Hein, 2004).

Los modelos matemáticos, en ocasiones, también necesitan ser ajustados. Esto se debe a que los valores medidos en el mundo real no siempre se ajustan a este de forma perfecta, debido a errores de medida o a variedad de factores no controlables. Por tal motivo, se debe admitir que un modelo cometerá cierto error, por lo cual es necesario realizar un ajuste al mismo (Friswell y Mottershead, 2013). Para ello se modifican los parámetros del modelo, con el fin de obtener un mejor acuerdo entre los resultados numéricos y la información experimental. Por ejemplo, un experto podría saber que el caudal de un río puede depender de la precipitación que cae en ciertos puntos pero la forma de la relación (lineal, cuadrática, cúbica) puede variar según los datos concretos, y por tanto es necesario obtener los exponentes que ajustan el modelo para ciertos datos concretos. De igual forma, puede decirse de los modelos que describen epidemias, los cuales requieren ser ajustados, para luego ser utilizados en predicciones futuras (Brauer, 2005; Medina Mendieta, *et al.*, 2020; Mendieta, *et al.*, 2020; Villalobos-Arias, 2020).

Finalmente, el ajuste de los parámetros se puede ver como un problema de optimización, en el cual las variables son los parámetros de las ecuaciones, y la función objetivo sería minimizar el error entre estos parámetros y las variables reales del modelo.

HERRAMIENTAS PARA RESOLVER O MODELAR PROBLEMAS DE OPTIMIZACIÓN

Para la modelación y resolución de problemas de optimización, como el ajuste de modelos, se pueden utilizar una gran variedad de *softwares*. Entre las herramientas mayormente utilizadas se encuentran:

- **XLSTAT** (Binumol, *et al.*, 2020) y **Minitab Statistical Software** (Clewett, 1986): ambos son *softwares* de pago que ofrecen variedad de métodos de ajuste de curvas tanto en regresión lineal como no lineal.
- **Solver** (Zhu, 2003), **Evolver** (Brakke, 1994), **CPLEX Optimizer** (García-López, *et al.*, 2017): son *softwares* que permiten resolver problemas de optimización lineales y no lineales. En el caso de **Evolver** cuenta con una extensión comercial que permite la modelación de problemas de mayor complejidad.
- **AMPL** (Gay, 2015), **GAMS** (Bussieck y Meeraus, 2004), **LINGO** (Cunningham y Schrage, 2004): son herramientas que se utilizan para modelar problemas de optimización. El principal inconveniente de estos, es que requieren conocimientos del lenguaje de programación de que implementan y que no modelan problemas de optimización de gran complejidad.
- **MatLab** (Gupta, 2014), **Maple** (Bernardin *et al.*, 2011), **Maxima** (Mysovskikh, 2004), **R** (Tippmann, 2015): son herramientas con gran variedad de prestaciones para el modelamiento lineal y no lineal. En el caso de **MatLab** y **R** son *softwares* que permiten la inclusión de paquetes que contienen algoritmos metaheurísticos. Lo que permite la modelación de problemas más complejos. En todos los casos para utilizarlas se requiere dominio del lenguaje que implementan.
- **MALLBA** (Alba, *et al.*, 2002), **METSlib** (Maischberger, 2011), **MOMHLib++** (Jaszkiewicz y Dąbrowski, 2010), **Open Metaheuristics** (Dreo, *et al.*, 2007), **BiCIAM** (Fajardo J., *et al.*, 2016): son bibliotecas de algoritmos metaheurísticos, diseñadas para la modelación de problemas de mayor complejidad. Todas excepto BiCIAM, se encuentran publicadas bajo Licencia Pública General (o GPL por sus siglas en inglés). Para utilizarlas se necesitan conocimientos del lenguaje que implementan.

Las principales alternativas para la modelación de problemas de optimización mediante el empleo de los *softwares* anteriores están bajo licencias de pago, o requieren conocimientos del lenguaje de programación que implementan. La mayoría permite la modelación de problemas de optimización complejos, pero se debe hacer uso de la programación, es este un problema común en las bibliotecas de algoritmos metaheurísticos.

BIBLIOTECA DE CLASES BiCIAM

BiCIAM es una biblioteca de clases que implementa un modelo unificado de algoritmos metaheurísticos. BiCIAM provee diferentes tipos de algoritmos tanto basados en un punto como basados en poblaciones de puntos, para resolver problemas monobjetivos y multiobjetivos.

Entre estos algoritmos monobjetivos (con versión multiobjetivo) se encuentran: Escalador de colinas, Recocido Simulado, Estrategia Evolutiva, Algoritmo Genético, NSGAI (solo multiobjetivo), entre otros. En Jenny Fajardo (2015) se le añadió un algoritmo llamado “Portafolio de algoritmos”, el cual puede ser utilizado en problemas de optimización dinámicos y combina diferentes metaheurísticas.

Varios problemas de optimización han sido resueltos utilizando BiCIAM, pero solo usuarios con conocimientos de programación hacen uso de estas. Algunos de los problemas que han sido resueltos en BiCIAM son:

- **El problema de Ruteo de Vehículos (VRP):** es aquel donde una flota de vehículos parte de un depósito para servir a un conjunto de clientes dispersos geográficamente (Torres, *et al.*, 2016).
- **El problema Localización de Máxima Cobertura (MCLP):** consiste en satisfacer la mayor cantidad de demandas posibles, dada una cantidad de instalaciones. Este se aplica en la ubicación de patrullas de policía (Fajardo J., *et al.*, 2017).
- **El problema de conformación de equipos de proyecto de *software*:** este consiste en asignar los roles del equipo de *software* a las personas con las competencias apropiadas, además considera las incompatibilidades entre los miembros y la carga de trabajo (Infante Abreu, *et al.*, 2015).
- **Descubrimiento de predicados difusos en bases de datos:** propone un método de minería de datos basado en metaheurística para obtener predicados difusos en forma normal, se nombra *FuzzyPred* (Ceruto, *et al.*, 2014).

KNIME COMO HERRAMIENTA DE ANÁLISIS DE DATOS

Knime (*Konstanz Information Miner*) es una plataforma de exploración de datos modular que guía el proceso de descubrimiento de conocimiento en bases de datos. Esta herramienta se basa en el diseño de flujos de trabajo (*workflows*) de forma visual y, además, permite ejecutar selectivamente algunos o todos los pasos de análisis, para luego investigar los resultados a través de vistas interactivas de los datos y modelos. Además, posibilita la presentación de informes e integración abierta, el análisis selectivo de pasos y su posterior visualización de resultados a través de vistas interactivas (Berthold, *et al.*, 2009).

Knime fue creada en julio de 2006, surge en el Departamento de Bioinformática y Minería de Datos de la Universidad de Constanza en Alemania. *Knime* está desarrollado sobre la plataforma *Eclipse* y programado, esencialmente en *Java*; es un *software* libre. Permite añadir *plugins*, importar y exportar flujos de trabajos, tiene integrada herramientas que permiten incorporar código en *R* o *Python*. El carácter abierto de la herramienta hace posible su extensión mediante la creación de nuevos nodos que implementen algoritmos a la medida del usuario (Berthold, *et al.*, 2009).

IMPLEMENTACIÓN DE NUEVOS NODOS EN KNIME UTILIZANDO LA BIBLIOTECA DE CLASES BiCIAM

A continuación, se presenta el diseño a alto nivel o diseño arquitectónico de nuevos nodos que se añaden en *Knime*. El primer nodo, llamado “*Mathematical Model Optimization*”, permite la

modelación de problemas de optimización definidos por un conjunto de variables, restricciones y función(es) objetivo. El segundo nodo, llamado “*Adjustment of Mathematical Function*”, permite ajustar modelos matemáticos, utilizar un conjunto de datos de entrada y determinar los coeficientes que mejor ajustan el modelo, modelando el ajuste de modelos como problema de optimización. Por último, el tercer nodo, llamado “*Mathematical Function Value Predictor*”, se utiliza para evaluar con nuevos datos, el modelo que haya sido ajustado en el nodo “*Adjustment of Mathematical Function*”. Los nodos “*Mathematical Model Optimization*” y “*Adjustment of Mathematical Function*” utilizan los algoritmos que provee BiCIAM para resolver y ajustar los modelos matemáticos.

Dado que los nodos propuestos resuelven problemas de optimización expresados como un sistema de ecuaciones matemáticas, se deben tomar en cuenta varios elementos tales como: una representación de una solución del problema, operadores de vecindad y una manera de calcular la función objetivo (Talbi, 2009). A continuación, se detalla la forma en la que se modelan cada uno de estos elementos, así como la arquitectura y flujo de trabajo para usar los nodos en *Knime*.

Definición del problema de optimización usando BiCIAM para optimizar modelos expresados como funciones matemáticas y ajustarlos

Para modelar un problema de optimización usando BiCIAM se debe definir inicialmente una codificación de la solución del problema. Esta codificación representa la lista (combinación) de variables que tiene el problema, que toman un valor particular por cada solución. La combinación de estas variables obtiene un determinado valor en la función objetivo. Los valores de estas variables son modificados para buscar nuevas combinaciones que optimicen la función objetivo del problema a través de los operadores de vecindad en las metaheurísticas. Para describir el procedimiento que se utiliza en optimización de modelos matemáticos se presenta un ejemplo a continuación:

Función objetivo:

$$\text{Maximizar: } 2 + a + b \quad (1)$$

Sujeto a:

$$0 < a < 15 \in \mathbb{R} \quad (2)$$

$$2 < b < 7 \in \mathbb{Z} \quad (3)$$

Donde para la variable de a pertenece al dominio de los números reales y es necesario usar un valor de precisión para convertirlo a un número binario. En este caso se utilizará 15.

El valor de precisión indica la cantidad de combinaciones de valores que se pueden obtener para a . En el caso de la variable entera b , el valor de precisión se calcula utilizando la fórmula siguiente:

$$\text{Precisión } (E) = \text{max} - \text{min} \quad (4)$$

Por tanto, para b el valor de precisión $=7-2=5$

Una vez obtenido el valor de precisión por cada variable, se calculan la cantidad de bits necesarios para representar estos valores. Para ello se divide el valor entre 2 y se ponen los valores del resto de la división. A continuación, se muestra el procedimiento que se realiza:

- | | |
|---|---|
| <p>a precisión =15</p> <ul style="list-style-type: none"> • $15/2 = 7$ resto 1. • $7/2 = 3$ resto 1. • $3/2 = 1$ resto 1 • $1/2 = 0.5$ resto 1 <p>○ Se obtiene 1111, con cantidad de bits=4.</p> | <p>b precisión =5</p> <ul style="list-style-type: none"> • $5/2 = 2$ resto 1. • $2/2 = 1$ resto 0. • $1/2 = 0,5$ resto 1. <p>○ Se obtiene: 101, con cantidad de bits=3, no se cuentan los ceros de la izquierda.</p> |
|---|---|

Una vez obtenido la cantidad de bits necesarios para representar la solución. Se conforma inicialmente la solución de forma aleatoria con un tamaño igual al total de bits obtenido entre todas las variables. En la figura 1 se muestra un ejemplo de conformación de la solución. Se muestran las posiciones que ocupa cada bit en la solución y el valor de codificación obtenido.

posición(i) en la solución	0	1	2	3	4	5	6
codificación	1	1	1	1	1	0	1

$a = 4$ bits $b = 3$ bits

Figura 1. Ejemplo de codificación de la solución en una variable entera

Luego, para conocer los valores no binarios de a y b y calcular la función objetivo del problema, primero se debe llevar el código binario a decimal de la siguiente manera:

- $a = 1011 = 1 * (2^3) + 0 * (2^2) + 1 * (2^1) + 1 * (2^0) = 11 = \text{valorDecodificado}.$
- $b = 101 = 1 * (2^2) + 0 * (2^1) + 1 * (2^0) = 5 = \text{valorDecodificado}.$

Luego el valor de cada variable entera (\mathbb{Z}) y/o real (\mathbb{R}) se obtiene utilizando las fórmulas presentadas en la tabla 1.

Tabla 1. Proceso para la decodificación y obtención de valores para las variables

$\mathbb{Z} = \min + \text{valorDecodificado} (5)$	$\mathbb{R} = \min + \frac{\text{valorDecodificado}}{2^{\text{cantidadBits}-1}} (\max - \min) (6)$
$a = 2+5$	$b = 0 + \frac{11}{(2^4-1)} (5 - 0) = 3,66667$
$a = 7$	$b = 3,66667$

Una vez obtenido el valor de las variables, estos son sustituidos en la función objetivo $(1) = 2+a+b$ donde se obtiene $2+7+3,66667 = 12.667$. Dicho procedimiento se realiza en el nodo “*Mathematical Model Optimization*” por medio de una interfaz visual.

Si se supone que se desee ajustar el resultado del modelo explicado anteriormente, se deben dar las soluciones obtenidas por los algoritmos metaheurísticos al nodo “*Adjustment of Mathematical Function*”. Este se utiliza para encontrar los parámetros óptimos del modelo que debe ser ajustado y modela el ajuste como problema de optimización. En la ecuación (7) se describe la función objetivo que permite ajustar modelos matemáticos.

$$\text{Función objetivo} = \text{Minimizar } \sum_i^n |y - y'| \quad (7)$$

Donde:

y : es el valor de referencia que se utiliza para ajustar el modelo (debe encontrarse en el conjunto de datos de entrada).

y' : es el resultado que se obtiene con la función luego de ser ajustada.

n : cantidad total de valores de los datos de entrada.

i : índice de cada valor de la tabla de entrada.

Tanto la descripción general del modelo y las variables que se describen en los ejemplos como los parámetros a ajustar se definen por medio de una interfaz visual.

Descripción de la arquitectura

En la figura 2 se muestra la arquitectura del sistema propuesto, la cual está basada en el enfoque de reutilización. Esta se encuentra proyectada en 3 capas, donde se puede ver la aplicación desplegada desde sus componentes más específicos hasta los más reutilizables.

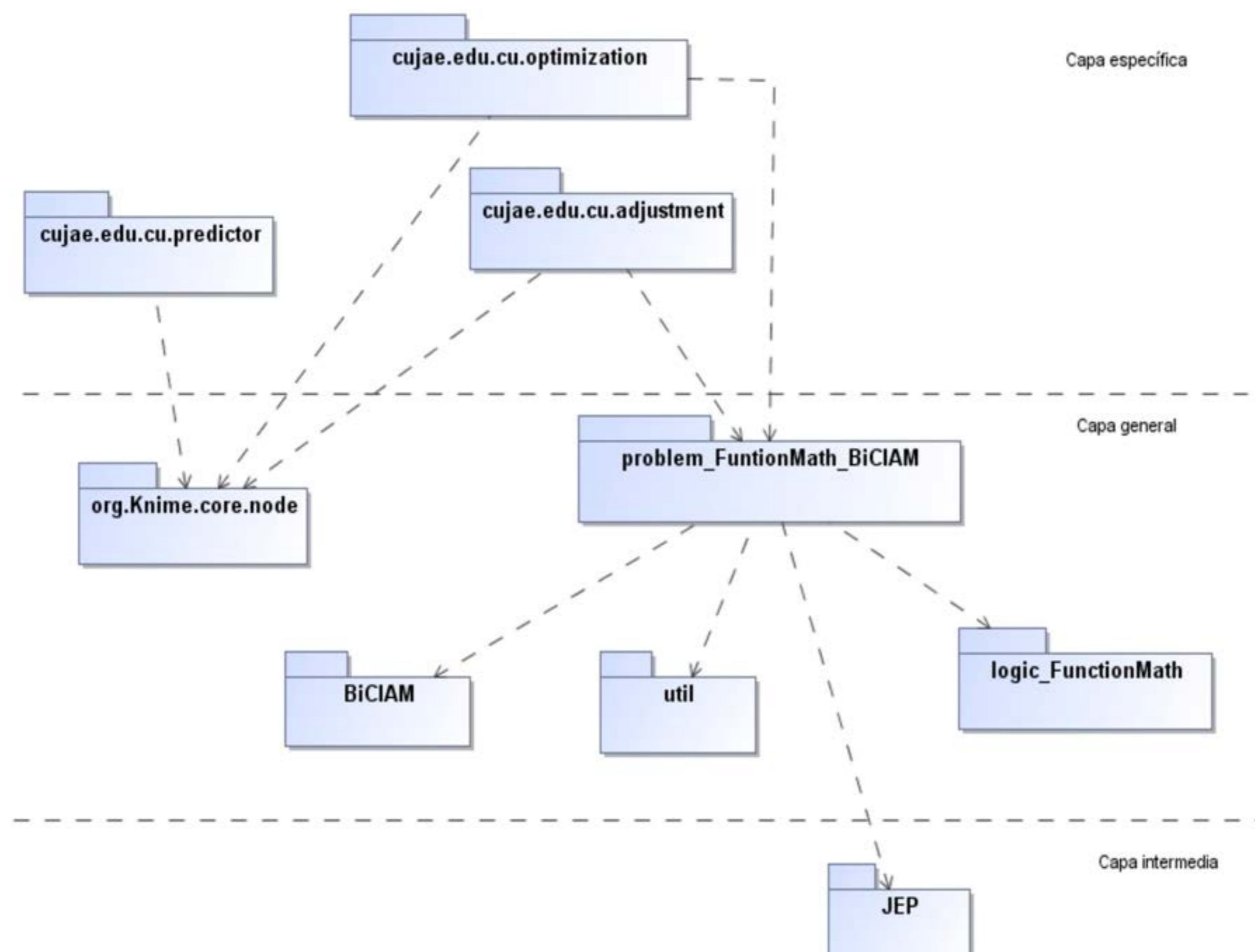


Figura 2. Vista de la arquitectura del sistema utilizando el enfoque de reutilización.

En la capa específica se encuentran los paquetes `cujae.edu.cu.predictor`, `cujae.edu.cu.optimization` y `cujae.edu.cu.adjustment`, que contienen las clases para la implementación de los nodos *Mathematical Function Value Predictor* y *Adjustment of Mathematical Function* respectivamente.

En la capa general se ven los paquetes `org.knime.core.node`, el cual es el punto de extensión de *Knime* para la implementación de nodos. El paquete `problem_FunctionMath_BiCIAM` contiene las clases que permiten modelar el problema de optimización en BiCIAM. El paquete llamado BiCIAM contiene las clases y algoritmos para implementar un problema de optimización.

El cuarto paquete, llamado “útil”, contiene las clases para el tratamiento de las excepciones y la salva de los datos. Por último, el paquete `logic_FunctionMath` contiene las clases que forman parte de la lógica del problema. Finalmente, en la capa intermedia se encuentra el paquete JEP que representa una biblioteca que permite la evaluación de expresiones matemáticas. En la figura 3 se muestra un ejemplo de un flujo de trabajo en *Knime* usando los nodos “*Mathematical Function Value Predictor*” (verde) y “*Adjustment of Mathematical Function*” (amarillo).

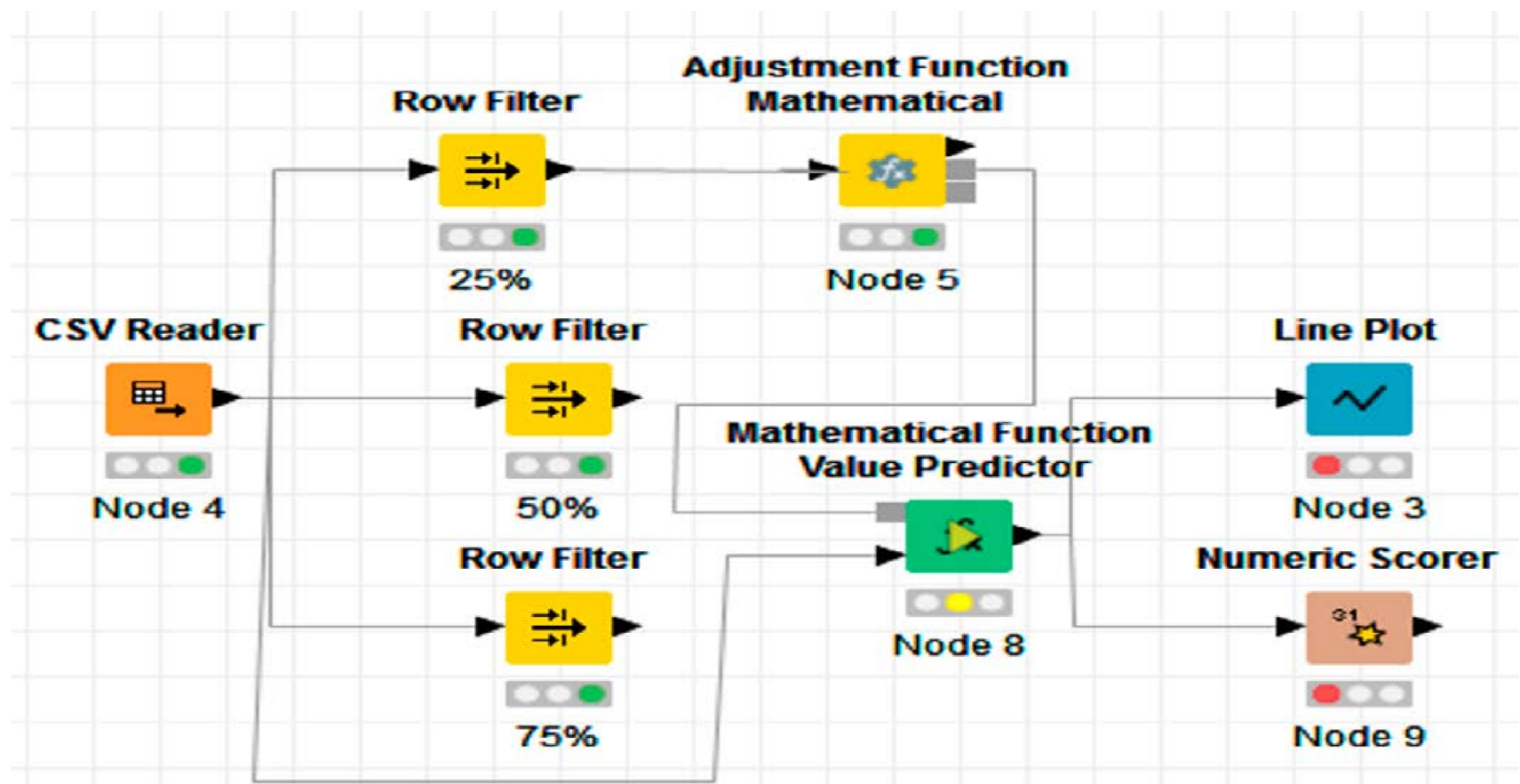


Figura 3: Flujo de trabajo para el ajuste de modelos matemáticos.

RESULTADOS Y DISCUSIÓN

Para mostrar la utilidad práctica de los nodos se realizaron dos casos prácticos. En el primer caso se modela el ajuste del modelo que define la curva de infectados por COVID-19 en Costa Rica hasta el día 27 de la pandemia y se compara con los resultados obtenidos en el artículo (Villalobos-Arias, 2020). En el segundo caso, se estima el pico de casos activos por COVID-19 en Cuba, se utilizaron distintas cantidades de datos para ajustar el modelo. En ambos casos

el ajuste como problema de optimización se le realiza al modelo matemático de Gompertz (Medina Mendieta, *et al.*, 2020; Valle, 2020; Villalobos-Arias, 2020) y para ello se utilizan el nodo “*Adjustment of Mathematical Function*”. Luego se usa el nodo “*Mathematical Function Value Predictor*” para la estimación del por la pandemia del COVID-19. Se utilizó la estrategia evolutiva con una población de 200 individuos y se ejecutó una vez con 10 mil iteraciones para resolver el problema del ajuste con el nodo “*Adjustment of Mathematical Function*”.

PRIMER CASO: AJUSTE DE LA CURVA DE INFECTADOS POR COVID-19 EN COSTA RICA

Para ajustar la curva de infectados de Costa Rica se utilizan los datos del 6 de marzo hasta el 1ro. de abril. Los datos que se utilizaron fueron extraídos del artículo (Villalobos-Arias, 2020) y corroborados en *El observador* (2020). El modelo que se utiliza para ajustar es el de Gompertz. A continuación se muestra dicho modelo (Villalobos-Arias, 2020) en la ecuación (8).

$$G(t) = a * e^{-b * e^{-c * t}} \quad (8)$$

Las variables a , b y c son parámetros libres que permiten ajustar el modelo a los datos, dado el instante de tiempo t . Este modelo permite predecir el crecimiento de la población infectada en un país o región, y que con solo los primeros valores de la curva se pueda predecir el punto donde los casos diarios empiezan a descender.

Al utilizar los datos de los infectados en Costa Rica, el modelo de Gompertz se ajusta con el nodo “*Adjustment of Mathematical Function*” y se compara con los obtenidos en la literatura (Villalobos-Arias, 2020) para analizar la calidad de las soluciones que obtuvo el nodo. Por último, con el modelo de Gompertz ajustado, se predicen los restantes valores hasta el 19 de mayo de 2020 comparándolos con los reales. La experimentación se realiza con el objetivo de comprobar la aplicación de los nodos “*Adjustment of Mathematical Function*” y “*Mathematical Function Value Predictor*”.

En la figura 4 se muestra la cantidad de casos por día para 27 instantes de tiempo t . En el eje horizontal la cantidad de días desde que comenzó la pandemia y en el eje vertical. La línea de color azul representa los datos reales, la de color naranja representa los valores que se obtuvieron en el nodo “*Adjustment of Mathematical Function*” y la verde representa los obtenidos en el ajuste de la literatura.

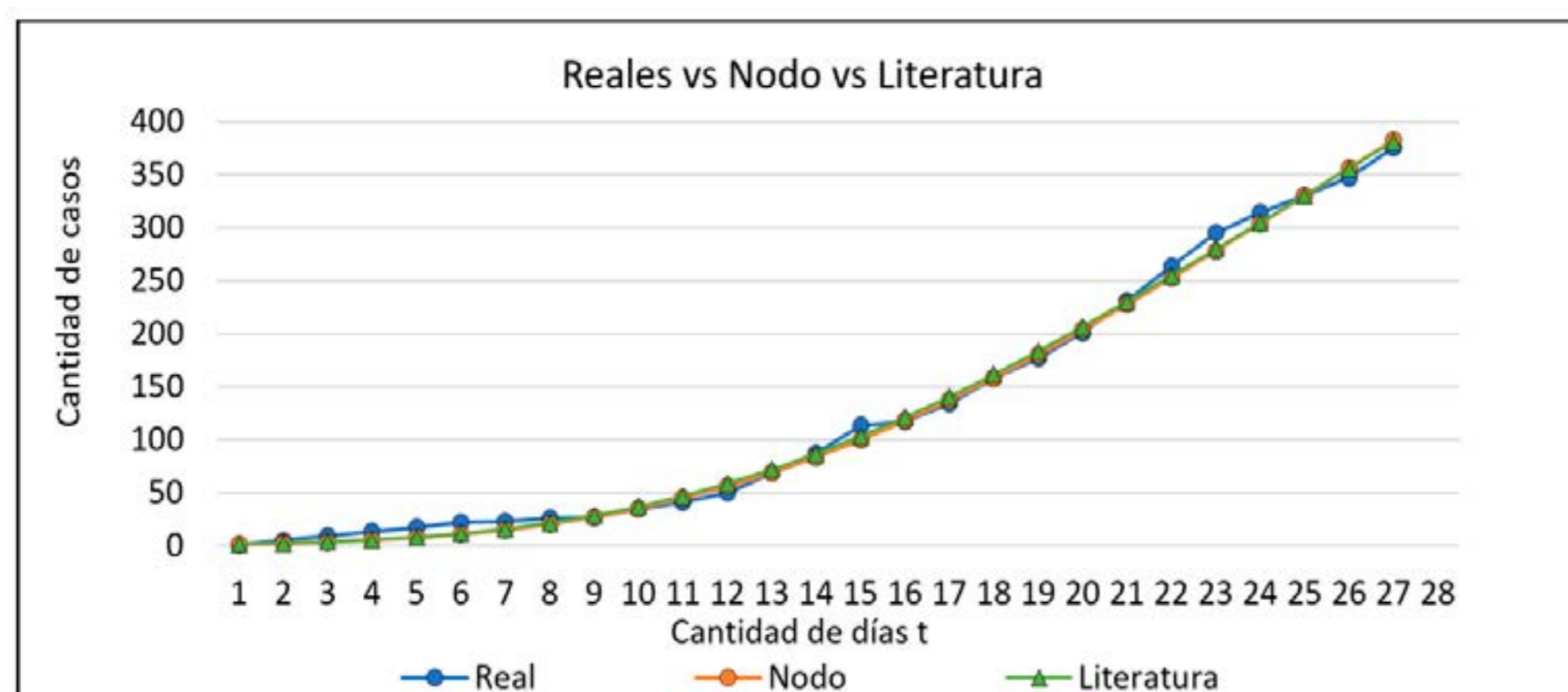


Figura 4: Valores reales de los infectados en Costa Rica, contra los obtenidos por el modelo de Gompertz ajustado en el nodo y en la literatura (Villalobos-Arias, 2020).

Se puede observar que no hay mucha diferencia entre los valores reales, los del nodo propuesto y los de la literatura. En la tabla 2 se muestran los coeficientes obtenidos para el modelo de Gompertz en la literatura (Villalobos-Arias, 2020) y en el nodo “*Adjustment of Mathematical Function*”.

Tabla 2. Resultados del ajuste del modelo de Gompertz para Costa Rica.

Recursos disponibles	Literatura	Nodo
Parámetros del modelo de <u>Gompertz</u> . $G(t) = a * e^{-b * e^{-c * t}}$	a=887, b=6.92, c=0.07.	a=872, b=7.05, c=0.078.
R^2	0,997	0,997
Error	151,901	146,195

También se muestra el valor de R^2 (coeficiente de determinación) y el valor del error entre el obtenido por el modelo ajustado y los reales. El valor de R^2 no difiere en ambos casos, además se obtiene un valor cercano a 1. Este indica que el 99,7 % de la variación de los valores del modelo ajustado explican su relación con los valores a los se ajusta. Con todo ello se comprueba que el nodo permite ajustar modelos y obtiene buenos valores con respecto a los que se obtienen en la literatura y en la vida real. Por lo que, con el nodo “*Adjustment of Mathematical Function*”, se pueden ajustar modelos matemáticos no lineales como el de Gompertz.

A continuación, para comprobar el poder de predicción del modelo que se ha ajustado anteriormente en el nodo usando 27 instantes de tiempo t, se utiliza el nodo “*Mathematical Function Value Predictor*” usando hasta 75 instantes de tiempo t (número de días de pandemia). En la Figura 5 se muestran los valores reales contra los obtenidos por el nodo y por el ajuste de la literatura. En esta se señala con una línea de color negro el punto límite de los valores que se utilizaron para ajustar el modelo.

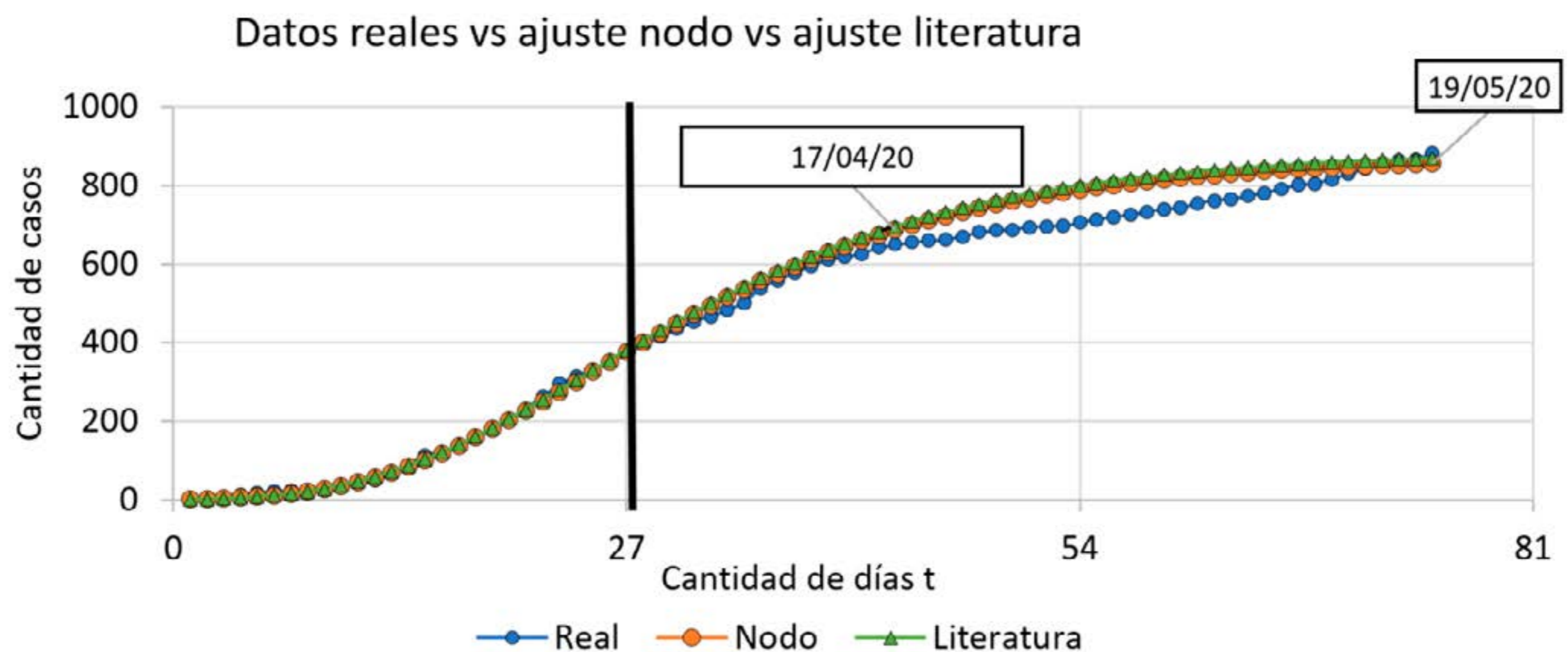


Figura 5: Curva de infectados en Costa Rica hasta el 19/05/20.

Se observa que el modelo ajustado predijo correctamente los datos hasta el 17 de abril y luego vuelve a coincidir el 19 de mayo. Esto significa que utilizando 27 valores reportados es posible obtener un modelo que predice el resto de los valores. Además, que las líneas de los valores del nodo y la literatura (Villalobos-Arias, 2020), coinciden casi de manera perfecta. Este experimento demuestra que la aplicación del nodo de predicción luego de obtener el modelo ajustado predice valores muy similares con el reportado en la literatura.

SEGUNDO CASO: PREDICCIÓN DEL PICO DE CONTAGIOS POR COVID-19 EN CUBA

En esta sección se realizará el ajuste y predicción de la curva de infectados, recuperados y fallecidos del COVID-19 en Cuba, utilizando los datos del 11 de marzo al 2 de junio tomados de (Covid19CubaData, 2020). El objetivo de la experimentación es demostrar la aplicación de los nodos en la estimación del pico de contagios en Cuba. Para ello primeramente utilizando el modelo de Gompertz, se ajustan la curva de infectados, recuperados y fallecidos utilizando el nodo “*Adjustment of Mathematical Function*”. El ajuste se realiza utilizando el 25%, 50%, 75% y 100% de los datos en cada caso. Con los modelos ajustados, se estiman los posibles valores hasta el 2 de junio de 2020, fecha que corresponde al día 84 de la pandemia usando el nodo “*Mathematical Function Value Predictor*”. En cada estimación por grupo de datos se determina los casos activos del virus. Estos se calculan con la fórmula siguiente:

$$\text{activos} = \text{infectados} - \text{recuperados} - \text{fallecidos} \quad (9)$$

Con los casos activos se tiene la cantidad de enfermos por el virus en cada instante de tiempo t (días de la pandemia), con estos datos se puede determinar cuándo será el mayor número de activos y con ello el pico de contagios. El ajuste del modelo de Gompertz y la predicción de los valores hasta el día 83 de la pandemia se muestran en la figura 6. Las líneas azules corresponden a la cantidad de casos activos reales y las naranjas a las estimadas por el modelo de Gompertz que utiliza el nodo.

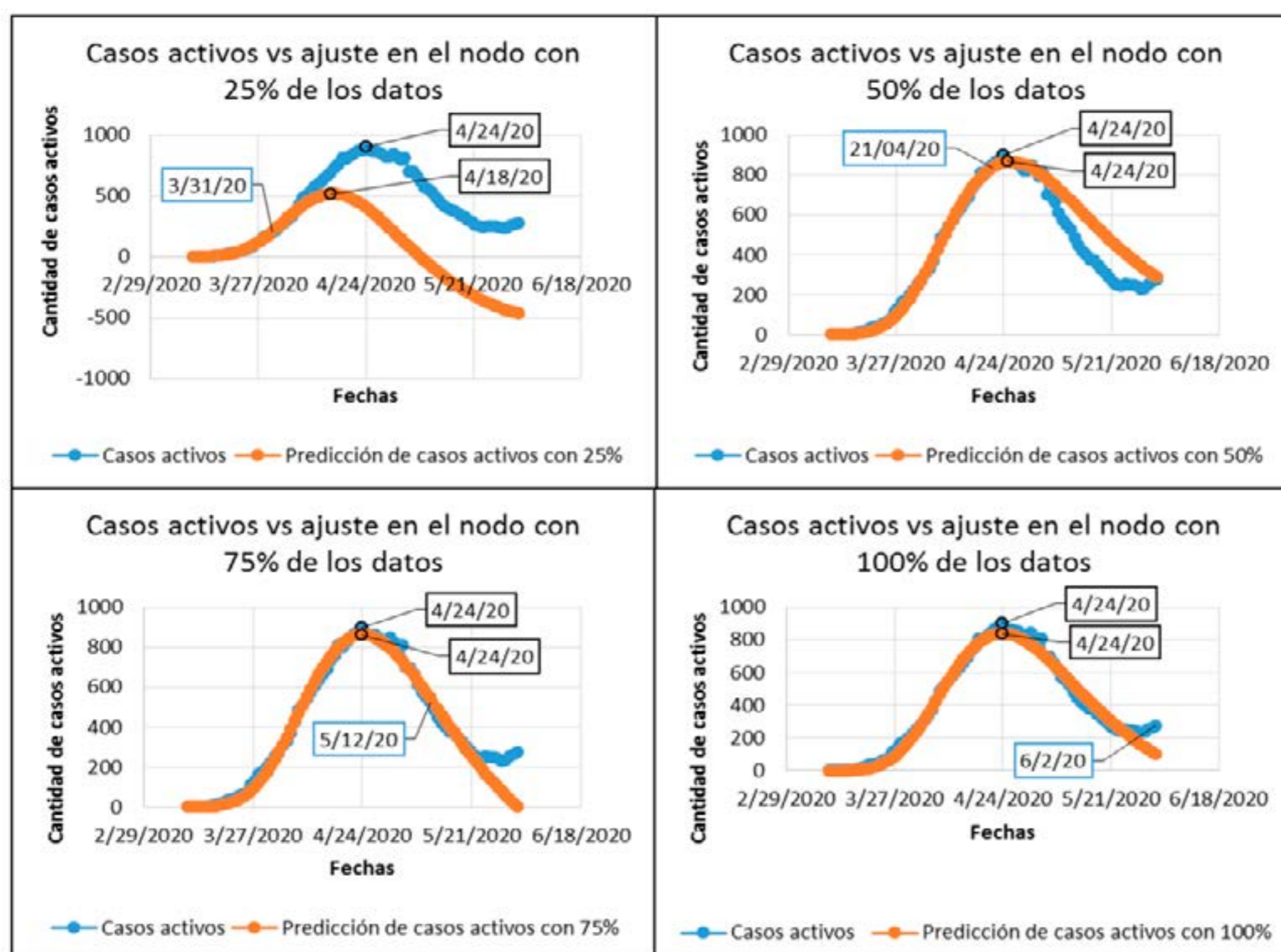


Figura 6.
Valores reales de los
casos activos contra los
predichos
en el nodo.

El gráfico del primer cuadrante corresponde a los valores reales contra los estimados cuando se utiliza solo el 25 % de los infectados, fallecidos y recuperados, el segundo cuadrante corresponde al 50 %, el tercero al 75 % y el cuarto al 100 %. Por cada por ciento de datos se muestra una etiqueta con la fecha límite de los datos que se tomaron para ajustar. Se puede observar que, a medida que aumentan los datos proporcionados al nodo “*Mathematical Function Value Predictor*”, la curva obtenida por el nodo es más ajustada al valor real.

Finalmente, a partir de los resultados de los casos prácticos, se puede decir que mediante los nodos propuestos para la herramienta *Knime* es posible modelar el proceso de ajuste de modelos matemáticos como problema de optimización y realizar predicciones a partir de este ajuste. Para ajustar el modelo de Gompertz solo bastó con las ecuaciones y un fichero con los valores posibles de las variables, para de esta manera ajustar sus parámetros usando meta-heurísticos. Las ecuaciones de los modelos a ajustar pueden tener cualquier tipo de expresión matemática. Desde el punto de vista del usuario, no es necesario tener conocimientos de programación para utilizar los nodos, simplemente debe conocer las características del modelo que desea ajustar, lo que permite una aplicación más amplia de la propuesta.

CONCLUSIONES

El ajuste de modelos matemáticos se puede expresar como problema de optimización y BiCIAM es una biblioteca que ha sido ampliamente utilizada para resolver una variedad de problemas de optimización, pero solo usuarios con conocimientos de programación pueden utilizarla. Con la implementación de los nodos “*Adjustment of Mathematical Function*” y “*Mathematical Function Value Predictor*” se puede modelar como problema de optimización el ajuste de modelos no lineales de carácter predictivo que pueden ser descritos con funciones matemáticas. Los nodos implementados permiten el uso de la biblioteca de clases BiCIAM sin necesidad conocimientos de programación. Además, al estar implementados en *Knime*, permite al usuario realizar análisis posteriores utilizando nodos ya creados dentro de la plataforma.

REFERENCIAS

- Alba, E., Almeida, F., Blesa, M., Cabeza, J., Cotta, C., Díaz, M., . . . Xhafa, F. (2002). *MALLBA: A Library of Skeletons for Combinatorial Optimisation*. Presentado en: Euro-Par 2002, Parallel Processing, 8th International Euro-Par Conference, Paderborn, Germany.
- Bernardin, L., Chin, P., DeMarco, P., Geddes, K., Hare, D., Heal, K., . . . Monagan, M. (2011). *Maple programming guide*: Citeseer.
- Berthold, M. R., Cebron, N., Dill, F., Gabriel, T. R., Kötter, T., Meinl, T., . . . Wiswedel, B. (2009). KNIME-the Konstanz information miner: version 2.0 and beyond. *AcM SIGKDD explorations Newsletter*, 11(1), 26-31.
- Biembengut, M. S., y Hein, N. (2004). Modelación matemática y los desafíos para enseñar ma-

- temática. *Educación matemática*, 16(2), 105-125.
- Binumol, S., Rao, S., & Hegde, A. V. (2020). Multiple Nonlinear Regression Analysis for the Stability of Non-overtopping Perforated Quarter Circle Breakwater. *Journal of Marine Science and Application*, 19(2), 293-300. doi:10.1007/s11804-020-00145-3
- Brakke, K. A. (1994). Surface evolver manual. *Mathematics Department, Susquehanna University, Selinsgrove, PA, 17870(2.24)*, 20.
- Brauer, F. (2005). The Kermack–McKendrick epidemic model revisited. *Mathematical biosciences*, 198(2), 119-131.
- Bussieck, M. R. & Meeraus, A. (2004). General Algebraic Modeling System (GAMS). In J. Kallrath (Ed.), *Modeling Languages in Mathematical Optimization* (pp. 137-157). Boston, MA: Springer US.
- Ceruto, T., Lapeira, O., Tonch, A., Plant, C., Espin, R., & Rosete, A. (2014). Mining medical data to obtain fuzzy predicates. *Lecture Notes in Computer Science*. Vol. 8649 (pp. 103-118): Springer.
- Clewett, A. J. (1986). The Minitab Statistical Package. *Journal of the Operational Research Society*, 37(12), 1201-1202. doi:10.1057/jors.1986.204
- Covid19CubaData. (2020, 2 de junio de 2020). Tablero estadístico interactivo sobre la COVID-19 en Cuba 2020. Recuperado de <https://covid19cubadata.github.io/#cuba>
- Cunningham, K. y Schrage, L. (2004). The LINGO Algebraic Modeling Language. In J. Kallrath (Ed.), *Modeling Languages in Mathematical Optimization* (pp. 159-171). Boston, MA: Springer US.
- Chapra, S. C., Canale, R. P., Ruiz, R. S. G., Mercado, V. H. I., Díaz, E. M., y Benites, G. E. (2007). *Métodos numéricos para ingenieros* (Vol. 5). México: McGraw-Hill.
- Dreo, J., Aumasson, J.P., Tfaili, W., & Siarry, P. (2007). Adaptive learning search, a new tool to help comprehending metaheuristics. *International Journal on Artificial Intelligence Tools*, 16(03), 483-505.
- El observador. (2020, 18 Mayo de 2020). COVID-19: Estadísticas. Recuperado de <https://observador.cr/covid19-estadisticas/>
- Fajardo, J. (2015). *Soft Computing en Problemas de Optimización Dinámicos*. (Ph.D. Thesis), Universidad de Granada, España.
- Fajardo, J., Masegosa, A. D., & Pelta, D. A. (2016). An algorithm portfolio for the dynamic maximal covering location problem. *Memetic Computing*, 9(2), 141-151. doi:10.1007/s12293-016-0210-5
- Fajardo, J., Porras, C., Sanchez, L., & Estrada, D. E. (2017). Software tool for model and solve the maximum coverage location problem, a case study: Locations police officers. *Revista de Investigación Operacional (RIO)*, 38(2), 141-149.
- Friswell, M., & Mottershead, J. E. (2013). *Finite element model updating in structural dynamics* (Vol. 38): Springer Science & Business Media.
- García-López, J. M., Ilchenko, K., & Nazarenko, O. (2017). *Optimization Lab Sessions: Major Features and Applications of IBM CPLEX*, Cham.

- Gay, D. M. (2015). *The AMPL Modeling Language: An Aid to Formulating and Solving Optimization Problems*, Cham.
- Gupta, A. K. (2014). *Numerical Methods using MATLAB* (1st ed.). USA: Apress.
- Infante Abreu, A. L., Díaz Hernández, R., André Ampuero, M., Rosete Suárez, A., Fajardo Calderín, J., y Escalera Fariñas, K. (2015). Solución al problema de conformación de equipos de proyectos de software utilizando la biblioteca de clases BICIAM. *Revista Cubana de Ciencias Informáticas*, 9, 126-140.
- Jaszkiewicz, A., & Dąbrowski, G. (2010). MOMHLIB++: Multiple Objective Metaheuristics Library in C++. Recuperado de <http://www.swmath.org/software/4708>
- López Jiménez, A. P., Espert Alemany, V., Carlos Alberola, M., y Martínez Solano, F. J. (2003). Metodología para la calibración de modelos de calidad de aguas. *Ingeniería del agua*, 10(4), 501-516.
- Maischberger, M. (2011). *COIN-OR METSlib a Metaheuristics Framework in Modern C++*. Firenze University. Recuperado de <https://projects.coin-or.org/metslib>
- Medina Mendieta, J. F., Cortés Cortés, M. E., Cortés Iglesias, M., Pérez Fernández, A. d. C., y Manzano Cabrera, M. (2020). Estudio sobre modelos predictivos para la COVID-19 en Cuba. *MediSur*, 18(3), 431-442.
- Mendieta, J. FM., Cortés, M. E. C., y Iglesias, M. C. (2020). Ajuste de curvas de crecimiento poblacional aplicadas a la COVID-19 en Cuba. *Revista Habanera de Ciencias Médicas*, 19, 3.
- Mysovskikh, V. I. (2004). Computer Algebra Systems and Symbolic Computations. *Journal of Mathematical Sciences*, 120(4), 1613-1617. doi:10.1023/B:JOTH.0000017891.69062.9c
- Talbi, E.-G. (2009). *Metaheuristics: from design to implementation* (Vol. 74): John Wiley & Sons.
- Tippmann, S. (2015). Programming tools: Adventures with R. *Nature News*, 517(7532), 109.
- Torres, I., Rosete, A., Cruz, C., y Verdegay, J. L. (2016). Solving a multiobjective truck and trailer routing problem with fuzzy constraints *Fuzzy Logic in Its 50th Year* (Vol. 341, pp. 237-255): Springer.
- Valle, J. A. M. (2020). Predicting the number of total COVID-19 cases and deaths in Brazil by the Gompertz model. *Nonlinear Dynamics*, 102(4), 2951-2957. doi:10.1007/s11071-020-06056-w
- Villalobos-Arias, M. (2020). Estimation of population infected by Covid-19 using regression Generalized logistics and optimization heuristics. *arXiv preprint arXiv:2004.01207*.
- Walpole, R. E., Myers, R. H., Myers, S. L., y Ye, K. (2012). Probabilidad y estadística para ingeniería y ciencias. Norma, 162, 157.
- Zhu, J. (2003). DEA Excel Solver *Quantitative Models for Performance Evaluation and Benchmarking: Data Envelopment Analysis with Spreadsheets and DEA Excel Solver* (pp. 263-283). Boston, MA: Springer US.

