



# TECNOLOGÍA *en marcha*

## Contenido

<i>Presentación</i> .....	3
Estrategia basada en el aprendizaje de máquina para tratar con conjuntos de datos no etiquetados usando conjuntos aproximados y/o ganancia de información	
Strategy based on machine learning to deal with untagged data sets using rough sets and/or information gain	
<i>Luis Alexander Calvo-Valverde</i> .....	4
Optimización multiobjetivo con funciones de alto costo computacional. Revisión del estado del arte	
Multiobjective optimization with expensive functions. Survey on the state of the art	
<i>Cindy Calderón-Arce, Pablo Alvarado-Moya</i> .....	16
Identificación de parámetros de un modelo térmico para un motor trifásico de inducción usando algoritmos genéticos	
Parameters identification of a thermal model for a three-phase induction motor using genetic algorithm	
<i>Luis Diego Murillo-Soto, Geovanni Figueroa-Mata, Osvaldo Guerrero-Castro</i> .....	25

Análisis de un modelo SIS para el estudio de la dinámica de propagación de la enfermedad al aplicar medidas de control

Analysis of a SIS model to study the dynamics of disease spread when control measures are applied

*Geisel Yajaira Alpízar-Brenes* ..... 42

Identificación de abejas sin aguijón (Apidae: Meliponini) a partir de la clasificación de los descriptores SIFT de una imagen del ala derecha anterior

Identification of stingless bees beginning from the classification of SIFT descriptors of an image of the right forewing

*Geovanni Figueroa-Mata, Juan Pablo Prendas-Rojas, Melvin Ramírez-Bogantes, Ingrid Aguilar-Monge, Eduardo Herrera-González, Carlos Manuel Travieso-González* ..... 51

Movimiento de centroides y transferencias: alternativas para construir vecinos en sobrecalentamiento simulado

Centroid movement and transferences: alternatives for generating neighbors in simulated annealing

*Jeffry Chavarría Molina, Juan José Fallas Monge* ..... 64

Modelo de corrección de errores y pronósticos de inflación

Error Correction Model and Inflation Forecast

*David Lowell Lovelady* ..... 77

# Presentación

La presente edición de Tecnología en Marcha es la culminación del seminario: Tópicos de Matemática Aplicada, realizado entre agosto y noviembre del 2014. El seminario surgió como una iniciativa de un grupo de colegas deseosos de compartir sus investigaciones en sus respectivos campos.

Durante casi tres meses, cada lunes fuimos partícipes de una charla en la que se tocaron temas muy variados como: series de tiempo, redes de Petri, dinámica de fluidos, modelos epidemiológicos, técnicas metaheurísticas, reconocimiento de patrones, técnicas de visión artificial, sistemas inteligentes, aprendizaje de máquina y minería de datos.

Este número recopila algunos de estos aportes, en los cuales, la constante fue el uso de la matemática como una herramienta fundamental para el análisis, estudio y solución de problemas provenientes de muy diversos campos de estudio.

La matemática aporta los modelos y las técnicas que permiten: hacer pronósticos de inflación, estudiar la dinámica de propagación de enfermedades, identificar especies, estimar los parámetros térmicos de un motor, optimizar funciones y plantear estrategias de aprendizaje de máquina para el manejo de grandes volúmenes de datos, como podrá descubrir el lector en este número.

Para finalizar, quiero agradecer de manera muy especial a cada uno de los colegas investigadores que hicieron posible el seminario y al programa de Bioingeniería por su apoyo.



# **Estrategia basada en el aprendizaje de máquina para tratar con conjuntos de datos no etiquetados usando conjuntos aproximados y/o ganancia de información**

**Strategy based on machine learning to deal with untagged data sets using rough sets and/or information gain**

Luis-Alexánder Calvo-Valverde<sup>1</sup>

---

*Fecha de recepción: 19 de junio del 2015*

*Fecha de aprobación: 26 de setiembre del 2015*

Calvo-Valverde, L. Estrategia basada en el aprendizaje de máquina para tratar con conjuntos de datos no etiquetados usando conjuntos aproximados y/o ganancia de información. *Tecnología en Marcha*. Edición especial. Matemática Aplicada, Mayo 2016. Pág 4-15.

---

<sup>1</sup> Doctorado en Ciencias Naturales para el Desarrollo (DOCINADE), Instituto Tecnológico de Costa Rica, Centro de Investigaciones en Computación, Programa Multidisciplinar eScience. Correo electrónico: lcalvo@itcr.ac.cr

## Palabras clave

Aprendizaje de máquina; minería de datos; conjuntos aproximados; entropía; ganancia de información; reducción de atributos.

## Resumen

Hoy en día se recogen datos de muy diversa índole y a un bajo costo, como no se había visto antes en la historia de la humanidad; por ejemplo, sensores que registran datos a cada minuto, páginas *web* que almacenan todas las acciones que realiza el usuario, supermercados que guardan todo lo que sus clientes compran y en qué momento lo hacen. Pero estas grandes bases de datos presentan un gran reto a sus propietarios ¿Cómo sacarles provecho?, ¿cómo convertir datos en información para la toma de decisiones?

Este artículo presenta una estrategia basada en el aprendizaje de máquina para tratar con conjuntos de datos no etiquetados utilizando conjuntos aproximados y/o ganancia de información. Se propone una estrategia para agrupar los datos utilizando *k-means*, considerando cuánta información aporta un atributo (ganancia de información), además de poder seleccionar cuáles atributos son realmente indispensables para clasificar nuevos datos y cuáles son dispensables (conjuntos aproximados), lo cual es muy beneficioso pues permite tomar decisiones en menor tiempo.

## Keywords

Machine Learning; Data Mining; Rough Sets; Entropy; Information Gain; Feature Reduction.

## Abstract

As had been seen in the history of humanity, today data of various kinds and cheaply collected, for example sensors that record information every minute, web pages that store all the actions performed by the user on the page supermarkets that keep everything their customers buy and when to do it and many more examples like these. But these large databases have presented a challenge to their owners How to take advantage of them? How to turn data into information for decision making? This paper presents a strategy based on machine learning to deal with unlabeled datasets using rough sets and/or information gain. A method is proposed to cluster the data using *k-means* considering how much information provides an attribute (information gain); besides being able to select which attributes are really essential to classify new data and which are dispensable (rough sets), which is very beneficial as it allows decisions in less time.

## Introducción

La aparición de grandes bases de datos (*Big Data*) en las organizaciones ha producido un auge en la búsqueda de medios que permitan sacarle provecho a estos grandes repositorios.

Es en este contexto en el que la aplicación del aprendizaje de máquina como medio para realizar minería de datos se ha utilizado con un éxito relativo. En particular, dos aspectos han llamado la atención de muchos autores, y tienen que ver con el objetivo de realizar el aprendizaje de una manera más eficiente:

- ¿Cómo etiquetar datos históricos para luego poder clasificar datos nuevos?
- ¿Cómo reducir la dimensionalidad de estas grandes bases de datos, por ejemplo, eliminando atributos redundantes o innecesarios, con el fin de hacer más eficiente el proceso?

Al respecto, la aplicación de los conjuntos aproximados ha tenido un resultado interesante al manejar conceptos de aproximación y reducción de atributos redundantes; esto se puede apreciar en trabajos como los de Bello y Verdegay (2010), César, Caicedo y Pérez (2010), Hedar, Wang y Fukushima (2008), Mahajan, Kandawal y Vijay (2012), Zdzislaw y Ziarko (1995), Rissino y Torres (2009), Thangavel, Shen y Pethalakshmi (2006), Velayutham y Thangavel (2011) y Zhang, Li y Chen (2013).

Un aspecto que queda claro al analizar los trabajos anteriores es que el tema del presente artículo está abierto y resta mucho por investigar, sobre todo en la búsqueda de algoritmos eficientes y no solo eficaces. En este trabajo se presentan algunas consideraciones que contribuyen a esta línea de investigación.

En primer lugar, se hace una presentación introductoria de conceptos que permiten comprender la propuesta del autor, luego se presentan los materiales y métodos utilizados y finalmente los resultados y algunas conclusiones.

## Terminología y conceptos

### A. Aprendizaje de máquina (*Machine learning*)

El aprendizaje de máquina (*machine learning*, como se denomina en inglés) se refiere al estudio de algoritmos de computadora que mejoran automáticamente a través de la experiencia. Este tipo de aprendizaje se ha utilizado en aplicaciones que van desde la minería de datos que descubren las reglas en grandes conjuntos de datos hasta sistemas de filtración de información que automáticamente aprenden los intereses de los usuarios. De acuerdo con Murphy (2012), el aprendizaje de máquina es un conjunto de métodos que automáticamente pueden detectar patrones en los datos y usar los patrones descubiertos para predecir datos futuros o para ejecutar otra clase de toma de decisión bajo incertidumbre, como, por ejemplo, planificar cómo recolectar más datos.

En cuanto a la clasificación de las técnicas de aprendizaje de máquina, el mismo Murphy las divide en:

- Aprendizaje supervisado: también llamado predictivo, cuyo objetivo es aprender a mapear desde  $X$  entradas a  $Y$  salidas, dado un conjunto etiquetado de  $N$  pares de entrada-salida; este conjunto se denomina *Training set*.
- Aprendizaje no supervisado: también llamado descriptivo, cuyo objetivo es encontrar patrones interesantes en las  $N$  entradas.
- Reforzamiento del aprendizaje: se usa para conocer cómo actúa o se comporta cuando se dan ciertas señales ocasionales de premio o castigo.

En algunas publicaciones, el aprendizaje de máquina se identifica con el reconocimiento de patrones; para algunos autores, este último tiene sus orígenes en la ingeniería, mientras que el aprendizaje de máquina creció en las ciencias de la computación. Sin embargo, ambas actividades pueden verse como dos facetas del mismo campo y han experimentado un desarrollo importante en los últimos años (Bishop, 2006).

Por otra parte, ante la pregunta de ¿en qué se diferencia el aprendizaje de máquina de la minería de datos?, se puede decir que la segunda pone más énfasis en modelos interpretables, mientras que el primero le da mayor relevancia a que los modelos sean precisos (Murphy, 2012).

## B. Conjuntos aproximados (*Rough Sets*)

En 1982, Pawlak introdujo la teoría de los conjuntos aproximados, siguiendo a Thangavel, Shen y Pethalakshmi (2006). En esta sección se presentan los conceptos fundamentales al respecto.

### Conceptos generales

Siguiendo a Rissino y Torres (2009), se puede decir que esta teoría fue desarrollada inicialmente para un universo finito en el cual la base de conocimiento es una partición, que se obtiene por una relación de equivalencia definida en ese universo. En la teoría de los conjuntos aproximados, el dato se organiza en una tabla llamada tabla de decisión. Las filas de la tabla de decisión corresponden a los objetos y las columnas a los atributos. En el conjunto de datos, una etiqueta de clase indica la clase a la cual pertenece cada fila. La etiqueta de clase se denomina un atributo de decisión (D), el resto de los atributos son los de condición (C), donde  $C \cup D = \emptyset$ , y  $t_j$  indica la  $j^{\text{ésima}}$  tupla de la tabla de datos. La teoría de los conjuntos aproximados define tres regiones basadas en las clases de equivalencia inducidas por los valores de atributo: aproximación por abajo, aproximación por arriba y frontera.

La aproximación por abajo contiene todos los objetos clasificados con seguridad basados en los datos recolectados, la aproximación por arriba contiene los objetos que pueden ser clasificados probablemente y la frontera es la diferencia entre la aproximación por arriba y la aproximación por abajo.

Sea  $U$  un universo finito. Sea  $R$  una relación de equivalencia definida en  $U$ , la cual particiona a  $U$ .  $(U, R)$  es una colección de todas las clases de equivalencia, llamada espacio de aproximación. Sean  $w_1, w_2, w_3, \dots, w_n$  elementos del espacio de aproximación  $(U, R)$ . Esta conexión se conoce como base de conocimiento. Entonces, para cualquier subconjunto  $B$  de  $U$ , la aproximación por arriba  $\bar{B}$  y la aproximación por abajo  $\underline{B}$  se definen como:

$$\bar{B} = \cup \{w_i / w_i \subseteq B\}$$

$$\underline{B} = \cup \{w_i / w_i \subseteq B \neq \emptyset\}$$

Al par ordenado  $(\underline{B}, \bar{B})$  se le denomina un conjunto aproximado. Además se tiene (Rissino & Torres, 2009):

$POS(B) = \underline{B} \Rightarrow$  ciertamente miembro de  $X$ .

$NEG(B) = U - \bar{B} \Rightarrow$  ciertamente no miembro de  $X$ .

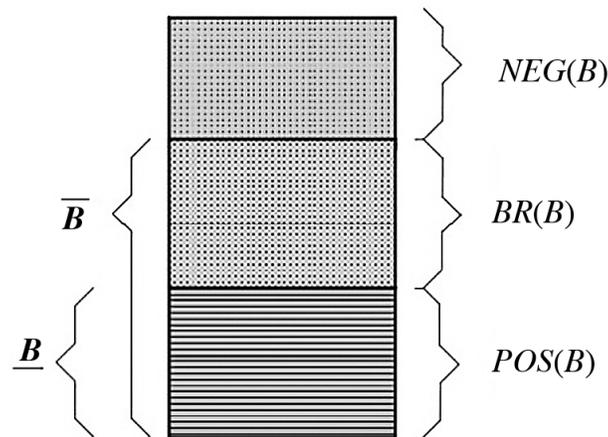
$BR(B) = \bar{B} - \underline{B} \Rightarrow$  posiblemente miembro de  $X$ .

En la figura 1 se aprecia cómo es esta distribución

Donde  $POS(B)$  refiere a la región positiva de  $B$ ,  $NEG(B)$  refiere a la región negativa de  $B$  y  $BR(B)$  refiere a la región frontera de  $B$ .

### Relación de indiscernibilidad

En una tabla de decisión existen atributos de condición y atributos de decisión. Se llama concepto a un conjunto de atributos de decisión para los cuales todos los objetos tienen el mismo valor de decisión.



**Figura 1.** Representación gráfica de las regiones en un conjunto aproximado, basado en Rissino y Torres (2009).

Conjuntos indiscernibles o elementales son aquellos que no se diferencian entre sí por sus atributos. Por tanto, una relación de indiscernibilidad es realmente una relación de equivalencia. La unión de conjuntos elementales, o indiscernibles, es un conjunto definible.

Una tabla de decisión se llama inconsistente o conflictiva si para al menos dos objetos los atributos de condición son iguales pero con diferente atributo de decisión.

Si un conjunto de atributos y su superconjunto definen la misma relación de indiscernibilidad, entonces cualquier atributo que pertenece al superconjunto y no al conjunto es redundante.

El conjunto de atributos que no tienen atributos redundantes es llamado mínimo (o independiente).

$P$  es una reducción de  $Q$ , si  $P$  es mínimo y las relaciones de indiscernibilidad definidas por  $P$  y  $Q$  son la misma.

Representado de manera más formal:

Sea  $IS$  una tabla de decisión donde:

$$IS = \{U, AT\}$$

donde  $U$  representa a todos los registros y  $AT$  son todos los atributos.

Sea  $a$  un atributo  $a \in AT$ .

Cada atributo tiene un dominio de valores  $V_a$  que puede tomar el atributo  $a$ .

$$a : U \rightarrow V_a$$

A cada objeto  $x$  en el universo  $U$  se le asigna un valor  $a(x)$  desde  $V_a$  a cada atributo  $a$ , y a cada objeto  $x$  en el universo  $U$ . Si  $V_a$  contiene valores perdidos para al menos uno de los atributos  $a$ , entonces  $IS$  es llamada una tabla de información incompleta, de lo contrario es completa.

Para cualquier subconjunto de atributos  $P \subset AT$ , hay una relación de equivalencia (indiscernibilidad).  $IND(P)$  es llamada relación de indiscernibilidad de  $P$ .

$$IND(P) = \{(x, y) \in U^2 \mid \forall a \in P, a(x) = a(y)\}$$

$$IND(P) = \cap IND(a) \text{ donde } a \in P$$

Con lo anterior, sea  $P$  un conjunto de atributos,  $a \in P$ , el atributo  $a$  es dispensable en  $P$  si:

$$IND(P) = IND(P - \{a\})$$

De lo contrario  $a$  es un atributo dispensable.

El conjunto de atributos  $A$ , donde  $A \subset P$  es llamado reducción de  $P$  si:

$$IND(A) = IND(P)$$

Y puede tener muchas reducciones, denotado como  $RED(P)$ . El conjunto de todos los atributos indispensables en  $P$  es llamado el core de  $P$  y se denota como  $CORE(P)$ , donde:

$$CORE(P) = \cap RED(P)$$

Si  $IND(P - \{a\}) = IND(P)$ ,  $a$  es un atributo dispensable y el conjunto de atributos constituye una reducción de  $P$ .

La intersección de todas las reducciones de  $P$  produce el core que son los atributos más importantes para clasificar correctamente. B3. Medidas de incertidumbre

Dado un conjunto de ejemplo  $X$ , no necesariamente definido por un conjunto  $P$  de atributos, una manera de estimar la calidad de la aproximación de  $X$  es la siguiente (Mahajan, Kandawal & Vijay, 2012).

*Calidad Aproximada por abajo* =

$$\frac{\#total \text{ de elementos en la Aproximación por abajo de } X}{\#total \text{ de ejemplos de } X}$$

*Calidad Aproximada por arriba* =

$$\frac{\#total \text{ de elementos en la Aproximación por arriba de } X}{\#total \text{ de ejemplos de } X}$$

### C. Reducción de atributos

La reducción de atributos tiene dos objetivos fundamentales (Mahajan, Kandawal & Vijay, 2012):

- Disminuir el número de atributos de condición.
- Maximizar la información contenida en los atributos seleccionados.

El logro de estos dos objetivos se ve reflejado en la mejora del tiempo de respuesta de los algoritmos de aprendizaje máquina, al tener que realizar menos comparaciones y cálculos y eliminar variables que pueden generar ruido a tal punto que produzcan generalizaciones bajo supuestos incorrectos. Todo por considerar variables que aportan poca información para la toma de decisiones o no son representativas del grupo de datos en estudio.

### D. Criterios de comparación de algoritmos en *k-means*

Existen varios criterios para comparar los resultados de diferentes versiones de los algoritmos *k-means*, pero hay dos en particular que son muy claros para su interpretación:

- Minimizar la suma de distancias: Se trata de sumar las distancias de todos los datos con respecto a los centros de los grupos en que quedaron clasificados y seleccionar la de menor valor.

- Maximizar el número de casos de éxito: Se trata de seleccionar la solución que tiene el mayor número de casos de éxito al relacionar la predicción con los valores reales.

### E. Entropía y ganancia de información

Dadas dos clases  $P$  y  $N$  en un espacio muestral  $S$ , donde:

$$S = P \cup N$$

Las cardinalidades están dadas por:

$$\begin{aligned} |P| &= p \\ & \text{y} \\ |N| &= n \end{aligned}$$

El término *entropía* se refiere a la cantidad de información necesaria para decidir si una muestra de  $S$  pertenece a  $P$  ó a  $N$ . y se define como (González, 2013):

$$E(S) = \frac{p}{p+n} \log_2 \left( \frac{p}{p+n} \right) - \frac{n}{p+n} \log_2 \left( \frac{n}{p+n} \right)$$

Partiendo de que al seleccionar un atributo  $b$  el espacio muestral es dividido en subconjuntos hijos de  $b$ , el modo de determinar cuánta información aporta un atributo  $b$  en un conjunto total de atributos  $A$  está dado por:

$$\text{Aporte}(b) = E(A) - \sum (\forall \text{ los subconjuntos hijos de } b)$$

Finalmente, si se tienen  $k$  clases,  $N$  instancias en el conjunto de datos, la entropía de todo el conjunto es  $E$ , la entropía de cada uno de los subconjuntos es  $E_1$  y  $E_2$ , la cantidad de instancias en una clase es  $k_1$  y en la otra  $k_2$ , entonces el mínimo aporte de información sería (Witten, Frank & Hall, 2001):

$$\frac{\log_2 N - 1}{N} + \frac{\log_2 3^k - 2 - k^E + k_1 * E_1 + k_2 * E_2}{N}$$

### F. Algoritmo *k-means*

*K-means* es un algoritmo para realizar aprendizaje no supervisado (Thangavel, Shen y Pethalakshmi, 2006). Su idea general es:

- Se particiona el conjunto de datos en  $K$  grupos (*clusters*) de modo aleatorio.
- Se seleccionan aleatoriamente  $K$  puntos centrales, uno de cada grupo (centroides).
- Para cada dato se calcula la distancia del punto a cada punto central de los grupos y el dato pasa a formar parte del grupo cuya distancia es menor a su centro.
- Si el dato es más cercano a su propio grupo, se queda en su grupo, de lo contrario pasa a formar parte del grupo del centro más cercano.
- Se repite el proceso anterior hasta que ningún punto se pase de grupo.

La selección inicial de los centros puede afectar significativamente los resultados.

## Materiales y métodos

Para realizar las pruebas se utilizaron conjuntos de datos provenientes de la Universidad de California (UCI) (Bache & Lichman, 2013). En el cuadro 1 se muestran más detalles de los mismos:

**Cuadro 1.** Conjuntos de datos utilizados de UCI.

Conjuntos de datos	Tipos de datos	Número de instancias	Número de atributos
<i>Car Evaluation</i>	Multivariados	1728	6
<i>Credit Approval</i>	Multivariados	690	15
<i>Chess</i>	Multivariados	3196	36
<i>Skin Segmentation</i>	Univariados	245057	4

Se tomaron conjuntos de datos con atributos de decisión para poder correr los algoritmos excluyendo este atributo y luego comparar los resultados obtenidos con lo que originalmente indicaban dichos atributos. Además, el tener conjuntos de datos para los cuales se conoce el atributo de decisión, permite determinar el número de grupos del algoritmo *k-means*.

Como en la mayoría de los procesos de minería de datos, cada repositorio tuvo que pasar por las etapas de: limpiar, integrar, seleccionar, transformar, minar, interpretar y presentar.

Se utilizó el IDLE de Python 2.7.2 para programar los algoritmos y se corrió en la siguiente plataforma:

- Fabricante del sistema; Hewlett-Packard
- Modelo del sistema: HP ProBook 6450b
- Tipo de sistema: PC basado en x64
- Procesador Intel(R) Core(TM) i5 CPU M 460 @
- 2.53GHz, 2534 Mhz, 2 procesadores principales, 4 procesadores lógicos
- Sistema operativo: Microsoft Windows 7 Professional.

En cuanto al criterio de comparación de los algoritmos de *k-means*, se escogió el de maximizar el número de casos de éxito, por cuanto al final el interés último es determinar qué tan bien hizo el agrupamiento.

Con el fin de comparar los resultados obtenidos, se corrieron tres procesos con los mismos conjuntos de datos. Se hicieron 100 repeticiones cuando se realizaban procesos aleatorios, con el fin de determinar el efecto medio del algoritmo. Cuando se utilizaron rangos, no tenía sentido repetirlo más de una vez pues el algoritmo es determinístico para un conjunto de datos dado. Solo en el caso de *k-means* clásico para *Skin Segmentation* se hicieron 50 corridas en lugar de 100, por motivo del tiempo computacional requerido para completar las 100, pues en realidad con 50 ya se podía estimar una media confiable del proceso aleatorio.

### *k-means* clásico con centroides aleatorios

Se utilizó *k-means* como algoritmo de agrupamiento de modo que los grupos resultantes sirvieran luego para etiquetar los objetos en su atributo de decisión (D); utilizando el número de grupo en el que el objeto quedó agrupado como valor del atributo de decisión (D).

### *k-means* utilizando solo los atributos con un aporte de información superior a una frontera

Se calculó la entropía de cada uno de los atributos y su ganancia de información. El método utilizado fue el siguiente:

- Sea  $E(C)$  la entropía de todo el conjunto de atributos.
- Se calcula cuanta información aporta la entropía de cada uno de los  $c$  atributos de condición (C), considerando:
- Sea  $E(c_i)$  la entropía del atributo de condición  $c_i$ .
- Como la selección del criterio de en qué valor, de los  $V_c$  valores, de dividir el atributo  $c$  para calcular la entropía puede ser muy diferente para cada atributo, se decide ordenar los  $V_c$  valores de menor a mayor y tomar la media como criterio de división.
- El aporte de información del atributo  $c$  es igual a:  $(C) - \sum E(\forall \text{ los subconjuntos hijo de } c)$
- Se utilizan los atributos de condición que aportan la mayor cantidad de información como los seleccionados para elegir de ellos los centros iniciales para el algoritmo *k-means*.

Una vez elegidos los atributos a considerar, si se desea que el atributo de decisión (D) tome  $V_d$  valores diferentes, entonces se corre *k-means* para formar  $V_d$  grupos, utilizando para el cálculo de distancias solo los atributos seleccionados por su mayor aporte de información. Se pueden inicializar los centros aleatoriamente o bien dividir el rango total de los valores del atributo  $c$  en  $k$  trozos uniformes y tomar estos valores como centros iniciales del algoritmo *k-means*. Al respecto, dado que son los atributos que más información aportan, se decidió inicializar los centros con rangos uniformes.

### *k-means* utilizando solo los atributos seleccionados por conjuntos aproximados

También se podría utilizar la teoría de los conjuntos aproximados, para determinar qué atributos de condición son indispensables y cuales dispensables y, por lo tanto, proceder a la reducción de atributos, calculando la relación de indiscernibilidad de cada uno de ellos. Recuérdese que siendo  $P$  el conjunto de atributos,  $a \in P$ , el atributo  $a$  es dispensable en  $P$  si:

$$IND(P) = IND(P - \{a\})$$

De manera similar, una vez elegidos los atributos a considerar, si se desea que el atributo de decisión (D) tome  $V_d$  valores diferentes, entonces se corre *k-means* para formar  $V_d$  grupos utilizando para el cálculo de distancias solo los atributos indispensables. Parecido al caso anterior, los centros se pueden inicializar aleatoriamente o se puede dividir el rango total en  $k$  trozos uniformes; con el fin de comparar los resultados se inicializaron los centros con rangos uniformes.

## Resultados

Luego de ejecutar los experimentos indicados, en el cuadro 2 se presentan los resultados obtenidos.

**Cuadro 2.** Resultados obtenidos para los diferentes conjuntos de datos.

Característica / Conjuntos de datos	Credit approval	Car Evaluation	Chess	Skin Segmentation
Total de registros	690	1728	3196	245057
Total de atributos incluyendo el de decisión	16	7	37	4
<i>k-means clásico</i>				
Media de la tasa de éxito <i>k-means</i> clásico	60.34	33.52	52.19	51.45
Desviación estándar de la tasa de éxito <i>k-means</i> clásico	3.46	8.40	02.59	9.19
Coefficiente de variación de la tasa de éxito <i>k-means</i> clásico	0.06	0.35	0.05	0.18
<i>k-means usando ganancia de información</i>				
Cantidad de atributos eliminados por ganancia de información	3	3	19	0
Media de la tasa de éxito usando los atributos restantes	56.81	28.41	59.73	46.67
Desviación estándar de la tasa de éxito usando solo los atributos restantes	0.00	0	0	0.00
Coefficiente de variación de la tasa de éxito usando solo los atributos restantes	0.00	0.00	0.00	0.00
<i>Conjuntos aproximados</i>				
Cantidad de atributos eliminados por conjuntos aproximados	12	0	5	0
Media de la tasa de éxito usando los atributos restantes	59.86	39.35	55.48	50.78
Desviación estándar de la tasa de éxito usando solo los atributos restantes	0.00	0.00	0.00	0.00
Coefficiente de variación de la tasa de éxito usando solo los atributos restantes	0.00	0.00	0.00	0.00

Del cuadro 2 se pueden resaltar varios aspectos:

1. Como es bien sabido, el algoritmo *k-means* clásico es muy dependiente de la selección de los centros iniciales. La inicialización de centros aleatorios tiende a tener desviaciones estándar altas y, por consiguiente, coeficientes de variación también altos.
2. Si se utiliza entropía y ganancia de información y solo se usan los atributos que aportan más información que el conjunto, y se usan rangos uniformes para los centroides en lugar de centros aleatorios, el proceso se vuelve determinístico para el mismo conjunto de datos; por eso la desviación estándar y el coeficiente de variación se muestran en cero.
3. Una vez etiquetados los datos, o si se dispone de conjuntos de datos ya etiquetados, y aunque la determinación de los atributos indispensables y los dispensables utilizando conjuntos aproximados es un proceso caro en tiempo computacional, una vez determinados, la reducción de atributos beneficia igualmente el proceso de clasificación a futuro. De hecho, se corrió *k-means* solo con los atributos indispensables y utilizando

rangos uniformes para los centroides, lo que mostró un rendimiento aceptable para todos los conjuntos de datos.

- Si bien *k-means* clásico con centros aleatorios mostró que en algunos casos obtenía mayor tasa de éxito que los otros, el problema es que su desviación estándar es alta y, por tanto, como el caso promedio no siempre se dará, perfectamente se puede dar el peor de los casos, o casos cercanos a este, y en estos escenarios su rendimiento es menor que cuando se usa ganancia de información o conjuntos aproximados.

En el fondo hay un tema de conveniencia en la selección del método, ¿se prefiere una probable mayor tasa de éxito (*k-means* clásico con valores aleatorios) o un valor determinístico que no siempre será mejor que el aleatorio, pero sin el riesgo de que la aleatoriedad lleve al peor de los casos (media menos desviación estándar) o a casos cercanos a este?

En el cuadro 3 se muestran los cálculos en que se comparan el mejor, el peor y el caso promedio para los conjuntos de datos en estudio y con las diferentes opciones de cálculo.

**Cuadro 3.** Comparación de las tasas de éxito para el peor, el mejor y el caso promedio.

Tipo / Conjuntos de datos	Credit approval	Car Evaluation	Chess	Skin Segmentation
<i>k-means</i> clásico – aleatorio – mejor caso	63.80	41.92	<u>54.78</u>	60.64
<i>k-means</i> clásico – aleatorio – peor caso	56.88	25.12	49.60	42.26
<i>k-means</i> clásico – aleatorio – caso promedio	60.34	33.52	52.19	51.45
<i>k-means</i> usando ganancia de información – rangos – determinístico	56.81	28.41	59.73	46.67
<i>k-means</i> usando conjuntos aproximados – rangos – determinístico	59.86	39.35	55.48	50.78

Como se aprecia en el cuadro 3, “*k-means* clásico – aleatorio – mejor caso” supera a los otros esquemas en su tasa de éxito para *Credit Approval* (alcanza un 63,80%), para *Car Evaluation* (obtiene un 41,92%) y para *Skin Segmentation* (con un 60,61%); solo para el conjunto de datos *Chess*, el valor obtenido 54,78%, es inferior a cuando se usa ganancia de información (59,73%) y cuando se utilizan conjuntos aproximados (55,48%).

## Conclusiones

La estrategia de aprendizaje de máquina propuesta para tratar con los conjuntos de datos no etiquetados se puede resumir de la siguiente manera:

Utilizar entropía y ganancia de información para seleccionar de cuáles atributos calcular los centros de *k-means*. La idea es utilizar solo los atributos que aportan mayor información.

Utilizar *k-means* con solo los atributos seleccionados del paso anterior para etiquetar los datos en su atributo de decisión.

Una vez etiquetados los objetos con los pasos anteriores, se pueden usar conjuntos aproximados para determinar qué atributos son dispensables y cuales indispensables y, por tanto, proceder a la reducción de atributos.

Una de las tareas futuras es la experimentación con otras fuentes de datos y determinar si otras técnicas adicionales a la de ganancia de información y a la de conjuntos aproximados pueden mejorar los resultados presentados en este trabajo.

Un aspecto que se podría argumentar es que el cálculo de la entropía, la ganancia de información y los conjuntos aproximados requiere un esfuerzo computacional previo al cálculo del *k-means*, que el método clásico (sin reducir atributos) se ahorra. Pero ante este argumento hay dos aspectos muy importantes, el primero es que si se cuenta con muchos atributos, la reducción de atributos recuperará el tiempo invertido al calcular el *k-means* con mucho menos atributos; y el segundo es que se sabe que la reducción de atributos colabora en la reducción del sobreajuste (en inglés *overfitting*) en caso de presentarse.

## Agradecimientos

El autor desea agradecer al Dr. Carlos González Alvarado, profesor del curso Minería de Datos impartido en el Doctorado en Ciencias Naturales para el Desarrollo (DOCINADE), por sus orientaciones en la preparación del presente documento. Al Dr. Pablo Alvarado Moya, tutor de la tesis del doctorando, y al Instituto Tecnológico de Costa Rica, por su financiamiento para realizar los estudios doctorales en el DOCINADE.

## Bibliografía

- Bache, K., & Lichman, M. (2013). *School of Information and Computer Science*. UCI Machine Learning Repository, University of California. Obtenido de <http://archive.ics.uci.edu/ml>
- Bello, R., & Verdegay L.J. (2010). Los conjuntos apróximados en el contexto de softcomputing Rough sets in the Soft Computing context. *4(1-2 ENERO- JUNIO)*, 5-24. España: Revista del Departamento de Ciencias de la Computación, Universidad Central de Las Villas.
- Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*, New York: Springer Science+Business Media. LLC.
- César, J., Caicedo, C., & Pérez, N. (2010). Servicio web inteligente para la clasificación de imágenes digitales utilizando conjuntos aproximados. *Ingeniería e Investigación*, *30*, 45-51.
- González, C. (2013). *Material del Curso Data Mining*. San José: Doctorado en Ciencias Naturales para el Desarrollo.
- Hedar, A., Wang, J., & Fukushima, M. (2008). Tabu Search for Attributes Reduction in Rough Set Theory. *Soft Computing*, *12(9)*, 909-918.
- Mahajan, P., Kandawal, R., & Vijay, R. (2012). Rough Set Approach in Machine Learning: A review. *International Journal of Computer Applications (0975 – 8887)*, *56(10)*, 1-12.
- Murphy, K. P. (2012). *Machine Learning: A probabilistic perspective*. Massachusetts: MIT Press.
- Pawlak, Z., Grzymala-Busse, J., & Slowinski, R. (1995). Rough Sets. *Communications of the ACM*, 88-95.
- Rissino, S., & Lambert-Torres, G. (2009). Rough Set Theory Fundamental Concepts , Principals , Data Extraction , and Applications. En *Data Mining and Knowledge Discovery in Real Life Applications* (pág. 438). Vienna: Julio Ponce and Adem Karahoca.
- Thangavel, k., Shen, Q., & Pethalakshmi, A. (2006). Application of Clustering for Feature Selection Based on Rough Set Theory Approach. *AIML Journal*, 19-27.
- Velayutham, C., & Thangavel, K. (2011). Unsupervised Quick Reduct Algorithm Using Rough Set Theory. *Journal of electronic science and technology*, *VOL. 9, NO. 3*, 193-201.
- Witten, I., Frank, E., & Hall, M. (2001). *Data Mining*. USA: ELSEVIER.
- Zhang, J., Li, T., & Chen, H. (2013). Composite rough sets for dynamic data mining. *Information Sciences*. *Vol 257*, 81-100.

# Optimización multiobjetivo con funciones de alto costo computacional. Revisión del estado del arte

## Multiobjective optimization with expensive functions. Survey on the state of the art

Cindy Calderón-Arce<sup>1</sup>, Pablo Alvarado-Moya<sup>2</sup>

---

*Fecha de recepción: 11 de junio del 2015*

*Fecha de aprobación: 17 de setiembre del 2015*

Calderón-Arce, C; Alvarado-Moya, P. Optimización multiobjetivo con funciones de alto costo computacional. Revisión del estado del arte. *Tecnología en Marcha*. Edición especial. Matemática Aplicada, Mayo 2016. Pág 16-24.

---

1 Correo electrónico: ccalderon@itcr.ac.cr. Escuela de Matemática, Instituto Tecnológico de Costa Rica.

2 Correo electrónico: palvaradomoya@gmail.com. Escuela de Electrónica, Instituto Tecnológico de Costa Rica.

## Palabras clave

Optimización multiobjetivo; costo computacional; algoritmos evolutivos; aproximaciones; modelos gaussianos; superficies de seudorespuesta.

## Resumen

La optimización multiobjetivo es un proceso complejo, más aún cuando las funciones objetivo que definen los problemas no están bien condicionadas o no cumplen con los requisitos mínimos para garantizar la convergencia de algoritmos clásicos, como convexidad, continuidad y diferenciabilidad. La literatura, entonces, se enfoca en el estudio de técnicas de optimización para problemas definidos por funciones con características particulares, por ejemplo, que el costo de su evaluación sea elevado, no convexas o no diferenciables. Este artículo hace una revisión general de las técnicas predominantes en problemas este tipo de funciones.

## Keywords

Multi-objective optimization; computational cost; evolutionary algorithm; approximations; gaussian models; pseudo response surface.

## Abstrac

The multi-objective optimization is a complex process, even more when the functions that define the problems are not well conditioned or do not meet the minimum set requirements to ensure the convergence of classical algorithms, such as convexity, continuity and differentiability. Hence, the technical literature focuses on optimization techniques for problems defined by functions with specific characteristics, like high evaluation cost, non-convexity or non-differentiability. This article provides an overview of some of the prevailing techniques for problems with these kind of functions.

## Introducción

La complejidad de las sociedades modernas y el progreso científico mundial han dado lugar al desarrollo y evolución de herramientas que contribuyan con el quehacer diario. El diseño, la implementación y la construcción de dichos instrumentos involucra elementos que envuelven la solución de problemas, basados en sistemas o modelos complejos y multifactoriales.

Esto ha llevado a la implementación de técnicas que optimizan la producción y eficiencia de las herramientas, como lo son los algoritmos creados para la simulación de circuitos y ejecución de cadenas completas de procesamiento de señales, los cuales evidencian la optimización de funciones objetivo con un número considerable de parámetros (Pereira et al., 2007).

En una subclase de problemas de ciencia e ingeniería, el costo de evaluar las funciones que los describen es alto, lo que provoca procesos de optimización en órdenes inaceptables para el desarrollo de los proyectos que los requieren y, al ampliar las dimensiones del espacio de búsqueda en el mundo multiobjetivo, ese costo se eleva exponencialmente.

Durante décadas se han utilizado métodos de optimización analíticos pero, para los problemas de interés en este artículo, la cantidad de variables que se deben tomar en cuenta y la naturaleza matemática de los modelos obligan a realizar simplificaciones que permitan la aplicación de dichos métodos (Alexandrov & Lewis, 2000; Mosat, 2006), lo que a su vez los hace poco fiables.

En la búsqueda de estrategias que soporten la optimización simultánea de más de un criterio de costo o aptitud surge la optimización multiobjeto (Deb, 2001). Esta optimiza objetivos de manera conjunta, poniéndolos a competir entre sí, de modo que para cada valor de una función de aptitud o costo se obtiene lo mejor de los otros objetivos. Así, la solución del problema general es una combinación de las soluciones óptimas de cada objetivo, por lo que dicha solución no es única (Deb, 2001; Marler & Arora, 2004).

En particular, la comunidad científica que trabaja en la implementación de algoritmos para la resolución de problemas de optimización multiobjetivo se divide en dos categorías, definidas por el tipo de funciones objetivo utilizadas. El primer grupo supone funciones objetivo de bajo costo computacional, se preocupa por la eficiencia global de los algoritmos de optimización en sí. Por otro lado, el segundo grupo asume un alto costo computacional de las funciones objetivo, por lo que se interesa por un proceso de optimización que reduzca el número de evaluaciones de dicha función, pues esta última domina la duración total de la optimización.

Este artículo hace una revisión del estado del arte orientado a los problemas de interés para el segundo grupo mencionado, es decir, enfocado en los problemas de optimización multiobjetivo con funciones objetivo costosas. En la siguiente sección se presenta la definición de optimización multiobjetivo y en las secciones posteriores se describen y analizan los algoritmos en tendencia propuestos para los problemas.

## Optimización multiobjetivo

En la optimización multiobjetivo, cada una de las funciones objetivo representa una aptitud o un costo a optimizar, las cuales dependen de  $n$  parámetros. Sin pérdida de generalidad, en el resto del artículo se mencionan problemas de optimización de aptitudes dejando claro que los principios y resultados también son válidos para problemas de optimización de costos. En particular, la  $r$ -ésima aptitud está dada por una función  $f_r: \mathbb{R}^n \rightarrow \mathbb{R}$  y la función evaluadora del problema a optimizar está compuesta por dichas funciones de aptitud. Suponiendo que se deben optimizar  $m$  aptitudes, la función a optimizar es de la forma  $F: \mathbb{R}^n \rightarrow \mathbb{R}^m$  y el problema general está dado por:

$$\begin{aligned} \max_x F(x) &= (f_1(x), f_2(x), f_3(x), \dots, f_m(x)) \\ \text{sujeto a } P_i &\text{ con } i=1,2,3,\dots,k \end{aligned}$$

donde  $x \in \mathbb{R}^n$  y la  $i$ -ésima restricción está dada por la proposición  $P_i$ . En la búsqueda de la solución óptima del problema general, se trabaja sobre un espacio de posibles soluciones en  $\mathbb{R}^n$ , donde todas satisfacen un criterio multiobjetivo de optimalidad.

En otras palabras, el objetivo es optimizar de manera simultánea todas las funciones objetivo, puesto que en la mayoría de los problemas no es posible encontrar un solo valor óptimo para todas ellas, ya que el óptimo de una función no necesariamente lo es para las demás (Zitzler et al., 2003).

Para comparar dos elementos del espacio solución se establece el concepto de dominancia de Pareto como un orden parcial sobre elementos en  $\mathbb{R}^n$ , en el que una solución  $x_i$  domina a otra solución  $x_j$  si se cumplen las siguientes dos condiciones:

- $x_i$  no es peor que  $x_j$  en todas las aptitudes

$$(\forall r)[f_r(x_i) \geq f_r(x_j)]$$

- $x_i$  es mejor que  $x_j$  en al menos una de las aptitudes

$$(\exists r)[f_r(x_i) > f_r(x_j)]$$

y se denota con  $x_i > x_j$ . Si alguna de las condiciones anteriores no se cumple, se dice que  $x_j$  no domina a  $x_i$ .

Una solución es Pareto óptima si y solo si no es dominada por ninguna otra solución, en ese caso todas las soluciones Pareto óptimas forman el conjunto de soluciones óptimas de Pareto denotado por  $S^* = \{x / (\neg \exists x_i) [x_i > x_j]\}$  y al conjunto  $F^* = \{F(x^*) / x^* \in S^*\}$  se le llama Frente de Pareto (Parmee et al., 2000; Ngatchou, 2008).

### Métodos clásicos adaptados

Estos métodos se derivan del desarrollo en series de Taylor de la función objetivo y se clasifican de acuerdo con la cantidad de términos que utilizan de dicha serie. Por ejemplo, el Steepest Descent (Gradient Descent) usa una aproximación de primer orden y el método de Newton una aproximación de segundo orden; ambos métodos realizan la búsqueda del óptimo por medio de direcciones de descenso, lo que acelera la convergencia a la solución si dicha aproximación es lo suficientemente cercana a la función real (Fliege & Svaiter, 2008).

Usualmente, los métodos clásicos se usan para resolver problemas uniobjetivo; sin embargo, existen adaptaciones para utilizarlos en problemas multiobjetivo. Por ejemplo, Fliege et al. (2006) adaptaron el método de Newton a problemas de optimización multiobjetivo con funciones convexas, formando el conjunto de soluciones óptimas de Pareto a partir de puntos llamados puntos críticos o estacionarios (Ortega & Rheinboldt, 2000), donde  $x \in \mathbb{R}^n$  es un punto crítico si  $(\forall s \in \mathbb{R}^n)(\exists r_0)[\nabla f_{r_0}(x)^T s \geq 0]$ .

A pesar de su exactitud, este tipo de métodos tiene la desventaja de que su comportamiento y convergencia dependen del tipo de función con la que trabajen. Por ejemplo, se requiere que la función sea derivable y convexa, condiciones no siempre garantizables en problemas de ingeniería.

Además, considerando que se trabaja con funciones cuya evaluación tiene un alto costo computacional y que la evaluación numérica de sus derivadas utiliza al menos dos llamadas de esa función, el costo total para evaluar dichas derivadas y gradientes es aún más alto; de manera que este tipo de métodos no es adecuado para esa clase de problemas.

### Métodos heurísticos de optimización

Las heurísticas o métodos heurísticos son algoritmos construidos de manera intuitiva. Algunos autores denominan metaheurística a aquellos casos en los que el algoritmo de optimización utiliza a su vez subprocesos heurísticos, pero esta es una distinción no generalizada (Schweickardt, 2009).

Entre las heurísticas más populares están Colonia de hormigas (AC: Ant Colony), Recocido Simulado (SA: Simulated Annealing), Búsqueda Tabú (TS: Tabu Search), Enjambre de Partículas (PS: Particle Swarm) y los algoritmos evolutivos (EA: Evolutionary Algorithm). Los EA se inspiran en conceptos de la evolución biológica tales como la reproducción, mutación, recombinación y selección (Vasunhara et al., 2015); además, balancean la exploración y explotación del espacio de búsqueda de manera exhaustiva con pocas evaluaciones. En contraste, AC, SA y TS son los métodos más antiguos y tienden a realizar más llamadas a la función evaluadora (Eskandari & Geiger, 2008).

Los algoritmos genéticos (GA: Genetic Algorithm) son parte de los algoritmos evolutivos y utilizan técnicas aleatorias y estocásticas para simular la evolución por medio de la selección de individuos con características prometedoras (Gen & Cheng, 2000; Reeves & Rowe, 2003). Non-dominated Sorting in Genetic Algorithm II (NSGA-II) y Strength Pareto Evolutionary Algorithm 2 (SPEA2) son dos de los algoritmos más utilizados en los EA. NSGA-II es un algoritmo genético que implementa una técnica de capas o rangos para clasificar en distintos niveles de dominancia a la población completa. Además, utiliza torneos binarios de selección con reemplazo y una distancia de aglomeración para conservar la diversidad en los elementos de la población (Chen & Chiang, 2014). SPEA2 es un algoritmo evolutivo que, a partir de un valor de bondad, conserva en una población o archivo externo a los mejores individuos para tomar de allí a los padres de los descendientes. Para la selección de padres utiliza un procedimiento de truncamiento basado en un torneo binario y en las distancias entre los individuos del archivo externo (Zitzler et al., 2001; Dastfan et al., 2014).

Por otro lado, los métodos híbridos combinan las heurísticas con los métodos clásicos para acelerar el proceso de búsqueda. En la mayoría de los casos, estos métodos están adaptados a problemas particulares que hacen que la combinación trabaje de manera adecuada.

S-Metric Selection Evolutionary Multiobjective Algorithm (SMS-EMOA) es un algoritmo que combina los GA con métodos como Newton y Gradiente Conjugado para realizar una búsqueda local (Koch et al., 2009). Por medio de una mutación polinomial y un cruzamiento binario simulado, SMS-EMOA genera una cantidad fija de descendientes, analiza probabilísticamente cada objetivo de manera independiente y, con base en los resultados, realiza búsquedas locales en algún objetivo específico por medio de alguno de los métodos clásicos.

Gutiérrez-Méndez (2011) toma en cuenta el comportamiento social para la selección de los individuos a los que se les aplicarán los operadores genéticos, de manera similar a PS. También, como parte de la búsqueda de óptimos globales, aplica búsquedas locales en ciertos sectores del espacio de búsqueda o en individuos con características particulares previamente definidas para refinar la búsqueda con candidatos prometedores. Las búsquedas locales en general se realizan por medio de procesos iterativos o algún método clásico.

El éxito de las heurísticas se debe a que encuentran soluciones satisfactorias en la mayoría de los problemas. Sin embargo, requieren de muchas evaluaciones para garantizar buenas soluciones globales como, por ejemplo, en problemas de 10 dimensiones se requieren alrededor de 1000 evaluaciones (Müller & Shoemaker, 2014), un aspecto que contribuye al aumento del costo computacional del proceso de optimización con funciones objetivo caras.

## Modelos gaussianos estocásticos

Autores como Ong et al. (2003) y Zhang et al. (2009) mencionan que los modelos basados en procesos gaussianos estocásticos se consideran los más eficientes para tratar problemas caros de optimización uniojetivo. Estos asumen que la función objetivo es un proceso gaussiano estocástico, donde los valores de la función objetivo para un punto nuevo se pueden estimar a través de los datos obtenidos en búsquedas anteriores y un factor de calidad, como la mejora esperada y la probabilidad de mejora, lo cual guía la decisión de cuáles puntos evaluar.

Efficient Global Optimization (EGO) es un conjunto de algoritmos diseñados para problemas uniojetivo que buscan soluciones de tal manera que se maximice la mejora esperada al menor costo posible, encontrando respuestas con alto potencial para mejorar el costo mínimo con un valor de predicción bajo. Así, en la búsqueda de algoritmos evolutivos para problemas multiobjetivo que reduzcan el costo computacional, surgen los llamados ParEGO como una extensión de EGO (Zhang et al., 2009). ParEGO predice el siguiente punto a evaluar a partir

de un subconjunto del espacio de búsqueda; sin embargo, no hace distinción entre un óptimo local y uno global.

ParEGO genera un número de soluciones en un hipercubo y, a partir de un modelo gaussiano, encuentra la probabilidad de que una solución particular mejore la solución general del problema, maximizando dicha probabilidad por medio del método Down Hill Simplex (Knowles & Hughes, 2005). La principal idea es recolectar datos de los candidatos previamente evaluados que puedan ser utilizados durante el proceso de evolución para construir y refinar el modelo de aproximación y a través de ellos evitar las evaluaciones de los candidatos menos prometedores. Las funciones serán evaluadas solo en los miembros más prometedores de la población, disminuyendo considerablemente el costo computacional (Ong, 2003). ParEGO requiere de una cantidad mínima de evaluaciones de la función original (cientos), de lo contrario, no converge al frente de Pareto real.

Eskandari y Geiger (2008) implementaron un algoritmo para tratar problemas con funciones de alto costo computacional llamado Fast Pareto Genetic Algorithm (FastPGA), que usa una técnica de clasificación basada en la información acerca de la dominancia de Pareto a través de soluciones y relaciones de nichos. Su desarrollo es similar al de NSGA-II e incluye algunas de las técnicas de ParEGO. Por medio de NSGA-II, optimiza las mejoras esperadas de todos los objetivos de manera independiente y así localiza el siguiente candidato a ser evaluado, sin utilizar toda la información dada por el modelo gaussiano estocástico, lo cual reduce la eficiencia (Deb et al., 2002).

Por otro lado, Multiobjective Evolutionary Algorithm/Descomposition (MOEA/D) descompone el problema en varios subproblemas uniobjetivo, obtenidos a partir de escalarizaciones del problema original definidas por una combinación lineal de las funciones que conforman la función objetivo general. Los subproblemas se relacionan entre sí por medio de las distancias entre los vectores peso de cada combinación lineal, donde dos subproblemas cercanos deben generar soluciones similares (Zhang & Li, 2007).

Así, Multiobjective Evolutionary Algorithm Descomposition with Gaussian Process Model (MOEA/D-EGO) combina MOEA/D con modelos gaussianos estocásticos. En cada iteración se construye un modelo gaussiano para cada subproblema a partir de los datos obtenidos en las búsquedas anteriores y las mejoras esperadas son optimizadas simultáneamente. Por medio de MOEA/D, este método genera nuevos candidatos y se usa un modelo basado en subconjuntos para mejorar la calidad de la predicción sin aumentar el costo computacional. En cada iteración se construye una distribución de predicción para cada objetivo de manera individual (Zhang et al., 2009).

## Aproximación de funciones

Recientemente se han propuesto técnicas de optimización para funciones costosas basadas en modelos sustitutos que aproximan a la función original. En vez de usar modelos computacionales costosos durante la optimización, se utilizan métodos que trabajan con una réplica de la función reduciendo la cantidad de evaluaciones de la función original a cientos en vez de miles, lo que produce resultados con un alto grado de exactitud a un costo menor (Müller & Shoemaker, 2014).

Las primeras propuestas de aproximación construían metamodelos que actuaban sobre la mayor parte del espacio de búsqueda, trabajando con un modelo para cada objetivo, pero eso requiere de evaluaciones en todo el espacio de búsqueda, cuando en realidad lo que interesa son los puntos cerca del frente de Pareto y no los puntos en el interior (Voutchkov & Keane,

2010). Debido a esto, ahora se trabaja con modelos localizados, eficientes en regiones de interés y no en el espacio completo.

Para minimizar la cantidad de evaluaciones que se le realizan a la función objetivo original, se usan modelos de aproximación en los que se toma una muestra de puntos y su respectiva evaluación en la función original, a partir de los cuales se crea una función sustituta que se utiliza en la mayoría de las evaluaciones para guiar el proceso. La función original se evalúa únicamente para actualizar los datos de la muestra y así el modelo utilizado en cada iteración (Haanpää, 2012).

Las superficies de seudorespuesta (PRS: Pseudo Response Surface) o métodos de superficie de respuesta (RSM: Response Surface Methods) evalúan solo puntos cercanos a la región Pareto óptima realizando un mínimo de evaluaciones a las funciones objetivo originales, pero para ello se requiere una estrategia de muestreo predeterminada. RSM utiliza modelos de predicción basados en redes neuronales para aproximar la superficie y predecir cuál o cuáles puntos es mejor evaluar (Messac & Mullur, 2008).

Los modelos más utilizados en la construcción de funciones sustitutas son interpolación, regresión, funciones de base radial (RBF: Radial Basis Functions) y funciones krigeanas (KF: Kriging Function). Sobresalen, entre estos, RBF y KF, porque el error es mínimo cerca de los puntos de muestra, requieren pocas muestras y funcionan adecuadamente en problemas con dimensión mayor a 25 (Müller & Shoemaker, 2014). Sin embargo, con el aumento en la cantidad de datos y variables del problema se incrementa la cantidad de evaluaciones y, en consecuencia, también el costo computacional (Voutchkov & Keane, 2010).

La eficiencia de estos métodos depende directamente de la muestra tomada. En algunos casos, la solución podría aproximarse a mínimos locales y en otros a óptimos globales. Es así como modelos del tipo Local Optima through Metamodels (LOOM) buscan todos los óptimos locales en un número reducido de llamadas a la función objetivo real, con base en repetidas búsquedas locales por medio de un metamodelo de esta. Dado que la mayor dificultad en los problemas de optimización es el costo de evaluar el desarrollo de cada solución en un espacio de búsqueda grande, LOOM trabaja de manera esparcida respecto al número de llamadas a la función objetivo, gracias al uso del metamodelo (Rivieri et al., 2013).

## Conclusiones

Los algoritmos evolutivos son los más predominantes en los métodos diseñados para optimizar problemas con funciones objetivo de alto costo computacional. Además, si las condiciones de un problema concreto lo permiten, es usual combinar dichos algoritmos con métodos clásicos e iterativos para refinar las búsquedas por medio de búsquedas locales intercaladas dentro del proceso de optimización.

La tendencia apunta al uso de modelos gaussianos y métodos de aproximación para disminuir la cantidad de llamadas a funciones costosas, utilizando aproximaciones de las mismas en la mayoría de las evaluaciones y de esa manera disminuir considerablemente el costo computacional del proceso de optimización con ese tipo de funciones.

Es claro que el éxito en el uso de superficies de seudorespuesta depende de los puntos o muestras que se utilicen para generar la aproximación, por lo que aún queda la interrogante de cómo seleccionar dichos puntos para generar una superficie suficientemente adecuada que se acerque al frente de Pareto con un mínimo error.

Por medio de PRS es posible encontrar buenas aproximaciones de óptimos locales, sin embargo, este método falla en la búsqueda de óptimos globales, por lo que es necesario

profundizar su estudio. En esa línea, se podría analizar la posibilidad de incorporar técnicas de aglomeración y criterios de separación para definir regiones de aproximación (Wikaisuksakul, 2014), en conjunto con metamodelos de aproximación dados por la combinación de varios modelos PRS (Lim et al., 2007). Así, en particular, en el uso de RBF se podría incluir un proceso para la optimización de los parámetros de dichas funciones y técnicas de aprendizaje como semiaprendizaje supervisado para el entrenamiento en la selección de los datos de entrada (Sun et al., 2014).

## Bibliografía

- Alexandrov, N. & Lewis, R. (April, 2000). *Analytical and Computational Aspects of Collaborative Optimization*. National Aeronautics and Space Administration, Langley Research Center, Virginia.
- Chen, S. & Chiang, T. (July, 2014). *Evolutionary Many-objective Optimization by MO-NSGA-II with Enhanced Mating Selection*. IEEE Congress on Evolutionary Computation, Beijing, pp. 1397-1404.
- Dastfan, A., Yassami, H. & Reza, M. (2014). Optimum Design of Passive Harmonic Filter by Using Game Theory. *Intelligence Systems in Electrical Journal*, (4), 13-22.
- Deb, K. (2001). *Multi-Objective Optimization using Evolutionary Algorithms*. Chichester, England: John Wiley & Sons, Inc.
- Deb, K., Pratap, A., Agarwal, S. & Meyarivan, T. (April, 2002). A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2), 182-197.
- Eskandari, H. & Geiger, C. (2008). A fast Pareto genetic algorithm approach for solving expensive multiobjective optimization problems. *Journal Heuristics, Springer Science*, 14, 203-241.
- Fliege, J. & Svaiter, B.F. (2000). Steepest Descent Methods for Multicriteria Optimization. *Mathematical Methods of Operation Research*, 51(3), 479-494.
- Fliege, J., Graña, L.M. & Svaiter, B.F. (May, 2006). Newton's Method for Multiobjective Optimization. *SIAM Journal on Optimization*, 20(2), 602-626.
- Gen, M. & Cheng, R. (2000). *Genetic Algorithms and Engineering Optimization*. New York: John Wiley & Sons, Inc.
- Gutiérrez-Méndez, F. (2011). *Optimización Multiobjetivo usando Algoritmos Genéticos Culturales*. Tesis de Maestro en Ciencias de la Computación. CINVESTAV, Unidad de Zacatenco, Departamento de Computación, México.
- Haanpää, T. (2012). Approximation Method for Computationally Expensive Nonconvex Multiobjective Optimization Problems. *Jyväskylä Studies in Computing*, 157. Jyväskylä.
- Knowles, J. & Hughes, E. (2005). *Multiobjective Optimization on a budget of 250 evaluation*. Springer-Verlag Berlin Heidelberg, EMO 2005, LNCS 3410, 176-190.
- Koch, P., Kramer, O., Rudolph, G. & Beume, N. (July, 2009). *On the Hybridization of SMS-EMOA and local search for Continuous Multiobjective Optimization*. GECCO'09. Montréal, Québec, Canada.
- Lim, D., Ong, Y. & Jin, Y. (July, 2007). *A Study on Metamodeling Techniques, Ensembles and Multi-Surrogates in Surrogates-Assisted Memetic Algorithms*. Genetic and Evolutionary Conference, London, pp. 1288-1295.
- Marler, R.T. & Arora, J.S. (2004). Survey of multi-objective optimization methods for engineering. *Structural and Multidisciplinary Optimization*, 26, 369-395. DOI: 10.1007/s00158-003-0368-6.
- Messac, A. & Mullur, A. (Spring, 2008). A computationally efficient metamodeling approach for expensive multiobjective optimization. *Optimization and Engineering*, 9(1).
- Mosat, A. (2006). *Deterministic and stochastic batch design optimization techniques*. Zürich: ETH. DOI:10.3929/ethz-a-005344678.
- Müller, J. & Shoemaker, C. (October, 2014). Influence of ensemble surrogate models and sampling strategy on the solution quality of algorithms for computationally expensive black-box global optimization problems. *Journal of Global Optimization*, 60(2), 123-144.
- Ngatchou, P., Zarei, A., Fox, W. & El-Sharkawi, M. (2008). *Modern Heuristic Optimization Techniques. Chapter 10: Pareto Multiobjective Optimization*. Wiley-IEEE Press.

- Ong, Y., Nair, P. & Keane, A. (2003). Evolutionary Optimization of Computationally Expensive Problems via Surrogate Modeling. *AIAA Journal*, 41(4), 687-696. DOI: 10.2514/2.1999
- Ortega, J.M. & Rheinboldt, W.C. (2000). Iterative solutions of nonlinear equations in several variables. *SIAM Classics in Applied Mathematics*.
- Parmee, I., Watson, A., Cvetkovic, D. & Bonham, C. (2000). Multiobjective Satisfaction within an Interactive Evolutionary Design Environment. *Evolutionary Computation, Massachusetts Institute of Technology*, 8(2), 197-222.
- Pereira, R., Alvarado, P. & Krawtschneider, H. (May, 2007). *Desing of a MCML Gate Library Applying Multiobjective Optimization*. IEEE Computer Society Symposium on VLSI 2007, Brasil.
- Reeves, C. & Rowe, J.E. (2003). *Genetic Algorithm: Principles and Perspectives. A guide to GA Theory*. Dordrecht: Kluwer Academic Publishers.
- Rivieri, J., Le Riche, R. & Picard, G. (September, 2013). LOOM, an algorithm for finding local optima of expensive functions. En *New and Smart Information Communication Science and Technology to support Sustainable Development*. France: Clermont Ferrand.
- Schweickardt, G. (diciembre, 2009). Metaheurística FPSO-X multiobjetivo. Una aplicación para la planificación de la expansión de mediano/largo plazo de un sistema de distribución eléctrica. *Energética*, (42), 73-88.
- Sun, C., Jin, Y., Zeng, J. & Yu, Y. (April, 2014). A two-layer surrogate-assisted particle swarm optimization algorithm. *Soft Computing*. DOI: 10.1007/s00500-014-1283-z.
- Vansundhara, R., Siva, S. & Selvaraj, M. (February, 2015). Evolutionary algorithms for de novo drug design. A survey. *Applied Soft Computing*, 27, 543-552.
- Voutchkov, I. & Keane, A. (2010). Multiobjective Optimization Using Surrogates. En *Computational Intelligence in Optimization*. Springer-Verlag Berlin Heidelberg, ALO 7, 155-175.
- Wikaisuksakul, S. (November, 2014). A multi-objective generic algorithm with fuzzy c-means for automatic data clustering. *Applied Soft Computing*, 24, 679-691.
- Zhang, Q. & Li, H. (December, 2007). MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition. *IEEE Transactions on Evolutionary Computation*, 11(6), 712-731.
- Zhang, Q., Liu, W., Tsang, E. & Virginias, B. (February, 2009). *Expensive Mutiobjective Optimization by MOEA/D with Gaussian Process Model. Technical Report CES-489*. School of Computer Science & Electronic Engineering, University of Essex.
- Zitzler, E., Laumanns, M. & Thiele, L. (May, 2001). SPEA2: Improving the Strength Pareto Evolutionary Algorithm. En *Computer Engineering and Networks Laboratory (TIK) Report*.
- Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C. & Grunert da Fonseca, V. (April, 2003). Performance Assessment of Multiobjective Optimizers: An Analysis and Review. *IEEE Transactions on Evolutionary Computation*, 7(2).

# Identificación de parámetros de un modelo térmico para un motor trifásico de inducción usando algoritmos genéticos

Parameters identification of a thermal model for a three-phase induction motor using genetic algorithm

Luis Diego Murillo-Soto<sup>1</sup>, Geovanni Figueroa-Mata<sup>2</sup>,  
Osvaldo Guerrero-Castro<sup>3</sup>

---

*Fecha de recepción: 19 de junio del 2015*  
*Fecha de aprobación: 26 de setiembre del 2015*

Murillo-Soto, L.; Figueroa-Mata, G; Guerrero-Castro, O. Identificación de parámetros de un modelo térmico para un motor trifásico de inducción usando algoritmos genéticos. *Tecnología en Marcha*. Edición especial. Matemática Aplicada, Mayo 2016. Pág 25-41.

---

1 Profesor Asociado, Máster en Computación, Instituto Tecnológico de Costa Rica, :  
lmurillo@tec.ac.cr

2 Profesor Catedrático, Máster en Computación,, Instituto Tecnológico de Costa Rica,  
gfigueroa@tec.ac.cr

3 Profesor Catedrático, Máster en Administración de la Ingeniería Electromecánica,,  
Instituto Tecnológico de Costa Rica, oguerrero@tec.ac.cr

## Palabras claves

Algoritmo genético; modelo térmico del motor de inducción; identificación.

## Resumen

Una forma alternativa de identificar los parámetros de un modelo térmico de un motor de inducción es por medio de algoritmos genéticos. Este trabajo en su inicio, describe brevemente los conceptos relacionados con los modelos térmicos de motores, así como los principales conceptos de los algoritmos genéticos. Posteriormente plantea un modelo térmico simplificado para un motor trifásico Baldor modelo M3558, de 2 hp, 230V, 60 Hz, dicho modelo posee dos nodos para los cuales se plantean sus ecuaciones diferenciales y sus soluciones analíticas. Para determinar los parámetros de este modelo térmico se alimenta un algoritmo genético con el cálculo de las pérdidas del motor eléctrico así como las temperaturas del motor capturadas por un banco de pruebas construido para tal fin. El algoritmo genético va modificando cada parámetro de la solución analítica hasta que se cumplan los criterios de ajuste. En la parte final del trabajo se confrontan las capacitancias y conductancias térmicas obtenidas contra cálculos teóricos de las constantes físicas del motor. Se muestra que los parámetros obtenidos con el algoritmo genético presentan un mejor ajuste respecto a los datos y no requieren del conocimiento de las magnitudes y constantes físicas relacionadas con el motor.

## Keywords

Genetic algorithms; induction motor thermal model; parameter identification.

## Abstract

Genetics algorithms are an alternative way to identify the parameters of an induction motor thermal model. Initially, this work briefly describes the concepts related to electric motors thermal models and the key concepts of genetic algorithms. Then it presents a simplified model for an M3558 Baldor motor, 2 HP, 230V, 60 Hz, the thermal model have two nodes, a set of differential equations and its analytical solutions is derived for each one. To determine the parameters of the thermal model, a genetic algorithm is fed with the calculation of motor losses and temperatures captured by a tests bench. The genetic algorithm changes each parameter of the analytical solution until adjustment criteria are met. At the end of this work, the thermal capacitances and inductances obtained are compared with theoretical calculations of the physical constants of the motor. We demonstrate that the parameters obtained with genetic algorithms fit better to the data and require little knowledge of the magnitude and physical constants related to the motor.

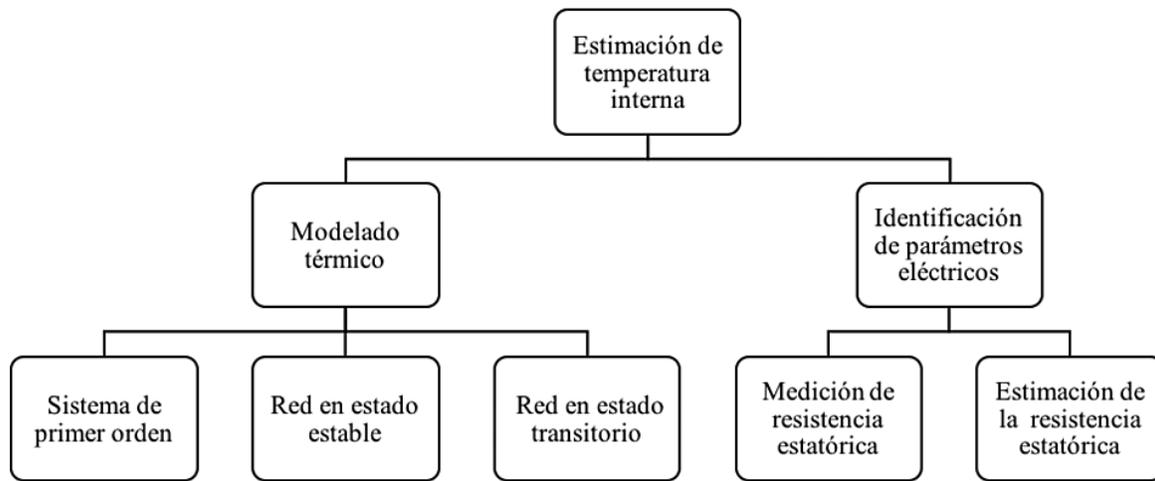
## Introducción

Debido a que los motores eléctricos de inducción representan la principal fuerza motriz para accionar los sistemas electromecánicos, estimar de forma correcta la temperatura interna de la máquina es una necesidad por su efecto sobre la vida útil de la máquina. Los materiales y barnices aislantes del motor se degradan de forma exponencial, por cada 10°C por encima de la temperatura máxima de la clase de aislamiento la vida útil se reduce a la mitad (Orae, 2000; Pillay & Manyage, 2006).

$$L_x = L_{100} \cdot 2^{(T_c - T_x)/k} \quad (1)$$

Donde  $L_x$  representa la estimación de la vida útil en la nueva condición de operación,  $L_{100}$  es la vida útil proyectada de fábrica usualmente 20.000 horas,  $T_x$  es la nueva temperatura de operación,  $T_c$  es la temperatura de la clase del aislamiento y  $k$  es una constante definida para cada clase de aislamiento donde 14, 11, 9.3, 8, y 10 corresponden a los aislantes clase A, B, F, H y H'.

La estimación de la temperatura interna del motor se puede dividir en cinco grandes estrategias tal y como se muestra en la figura 1.



**Figura 1.** Clasificación de los métodos para estimar temperaturas.

De acuerdo con (Gao, 2006), los modelos de primer orden son rápidos y eficientes, presentan una buena estimación de la temperatura y necesitan solamente la corriente estática, pero su estimación es un promedio general que no permite determinar los puntos calientes de la máquina. En (Zocholl, 2004) se realiza una comparación entre distintos tipos de modelos de primer orden, los cuales poseen una estructura similar a la ecuación (2).

$$\theta = I^2 R R_{th} \left( 1 - e^{-\frac{t}{R_{th} C_{th}}} \right) + \theta_A \quad (2)$$

Donde  $\theta$  y  $\theta_A$  son las temperaturas interna y la ambiental respectivamente, el término  $PR$  representa la totalidad de pérdidas del motor,  $R_{th}$  representa una resistencia térmica equivalente y  $C_{th}$  representa la capacitancia térmica del motor.

Por su parte, los modelos térmicos mallados para estados estables permiten predecir la temperatura en zonas específicas de la máquina, con el inconveniente de que la estimación de temperaturas en el transitorio térmico presenta errores de hasta el 100%. Estos modelos en estado estable tales como los que se plantean en (Kylander, 1995; Shenkman & Chertkov, 2000; Dutta & Chowdhury, 2012), obtienen las temperaturas resolviendo sistemas de ecuaciones lineales (3), donde  $P$  es el vector de pérdidas de la máquina,  $G$  es la matriz de conductancias térmicas del motor y  $\theta$  es el vector de temperaturas a calcular.

$$P = G \theta \quad (3)$$

Los modelos térmicos mallados para estados transitorios son aquellos que pueden predecir la temperatura en un punto específico de la máquina, tanto en régimen transitorio como en estado estable. Estos modelos incorporan un tercer vector a la ecuación (3), el cual se forma al

multiplicar una matriz  $C$  que contiene en su diagonal las capacitancias térmicas de la máquina, por un vector que contiene las derivadas de las temperaturas respecto al tiempo  $d\theta/dt$ , de tal forma que la expresión general queda:

$$C \frac{\partial \theta}{\partial t} = P - G\theta \quad (4)$$

Los modelos complejos que muestran el estado transitorio tal como y como se plantean (Oraee, 2000; Duran & Fernandez, 2004; Moreno, Hidalgo, & Martínez, 2001) ofrecen una predicción de la temperatura en el respectivo nodo, pero para poderla estimar se requiere conocer los materiales, geometrías, dimensiones y masas para calcular cada conductancia térmica entre los nodos  $G$  y la matriz de capacitancia  $C$ . Además, es necesario conocer en forma detallada cada una de las pérdidas reales  $P$  de la máquina, para eso se debe realizar pruebas de laboratorio. Cabe señalar que estos inconvenientes también se presentan en los modelos en estado estable, ecuación (3).

La estimación de la temperatura con base en parámetros eléctricos, se divide en dos esquemas fundamentales. El primero surge del estándar IEEE 112 ("IEEE Standard Test Procedure for Polyphase Induction Motors and Generators," 2004), en donde se indica que la variación de la temperatura interna provoca variaciones de las resistencias estáticas y rotóricas del motor en la siguientes proporciones.

$$R_2 = R_1 \frac{\theta_2 + k}{\theta_1 + k} \quad (5)$$

$$\theta_2 = \frac{R_2}{R_1}(\theta_1 + k) - k \quad (6)$$

Donde  $R_1$  representa la resistencia a la temperatura  $\theta_1$ , usualmente medido a temperatura ambiente con motor apagado,  $R_2$  la resistencia a la temperatura  $\theta_2$  y  $k$  es el coeficiente del material conductor, que para el cobre es 234.5 y para el aluminio es de 225. La ecuación (6) muestra el resultado de despejar la temperatura interna. Para obtener el valor de las resistencias se utilizan dos métodos, el primero calcula la resistencia aplicando la ley de Ohm a la medición del voltaje y corriente continua (CD). El segundo es la medición directa con puente wheatstone doble, mediante un miliohmímetro. Consecuentemente y dado que el punto más caliente de una máquina es el devanado estático (Boldea & Nasar, 2002), la medición de la resistencia estática es el método más sencillo para la estimación de temperaturas.

La desventaja del esquema de medición de la resistencia estática es que es invasivo, ya que requiere de equipamiento adicional y desconectar eléctricamente la máquina. Por tal motivo se han venido desarrollando métodos alternativos de obtención de la resistencia estática a partir de mediciones eléctricas de voltajes y corrientes, (Gao, 2006; Gao, Habetler, Harley, & Colby, 2005) o frecuencias (Cho & Seok, 2008). La idea de estos trabajos es desarrollar equipos en tiempo real que operen en línea con el motor y que realicen los respectivos cálculos de la estimación de temperatura interna. El principal inconveniente de estos métodos es que estiman la temperatura interna del devanado, pero no estima el comportamiento de temperaturas en otras zonas de interés en el motor, tal como la temperatura del núcleo, roles, carcasa, etc. Por otro lado, si se aplican técnicas metaheurísticas es posible estimar los parámetros de los modelos térmicos (3) y (4) que es la propuesta del presente trabajo.

### Algoritmo genético

Los algoritmos genéticos (GA) fueron desarrollados por John Holland en la década de los años 70. Se fundamentan en los principios biológicos de la evolución de las especies: selección natural y supervivencia del más fuerte.

Son algoritmos poblacionales, ya que trabajan con un conjunto de soluciones factibles llamado población. De este modo, en lugar de pasar de una única solución a la siguiente, se pasa

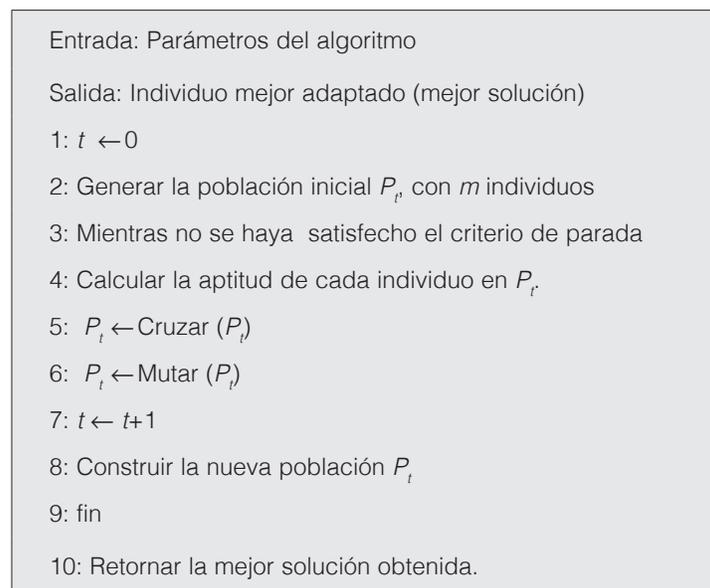
de una población a otra mediante la combinación y mutación de soluciones factibles. La combinación de soluciones factibles se lleva a cabo mediante cruzamiento, con la esperanza de producir descendientes mejor adaptados que sus ancestros. De esta manera, la especie evoluciona adaptándose mejor al entorno. Eventualmente, se producen mutaciones con la idea de producir descendientes con características distintas a las de sus progenitores, estas son las causantes de la aparición de individuos superiores. Sin mutaciones la especie no podría evolucionar hacia organismos más complejos y mejor adaptados a su medio. Por esta razón, el proceso de mutación se considera primordial en el proceso de evolución (Talbi, 2009).

La selección de individuos que formarán la nueva generación, buscan imitar la selección natural, en el sentido de que los individuos más adaptados tienen mayor probabilidad de ser seleccionados para formar la nueva generación mediante cruzamiento y mutación.

Existen muchas formas de realizar el proceso de selección. Entre las más usadas están los métodos de selección estocástica como la selección por ruleta simple y la selección por torneo determinístico o probabilístico.

Los algoritmos genéticos tradicionalmente se utilizan resolver problemas de búsqueda y/o optimización, en los cuales se tiene que determinar la mejor solución de un conjunto de soluciones factibles de gran tamaño.

A cada uno de las soluciones factibles de una generación se le asocia un valor llamado aptitud. Este valor cuantifica la calidad de la solución y permite comparar soluciones de la misma generación y/o con soluciones de generaciones anteriores.



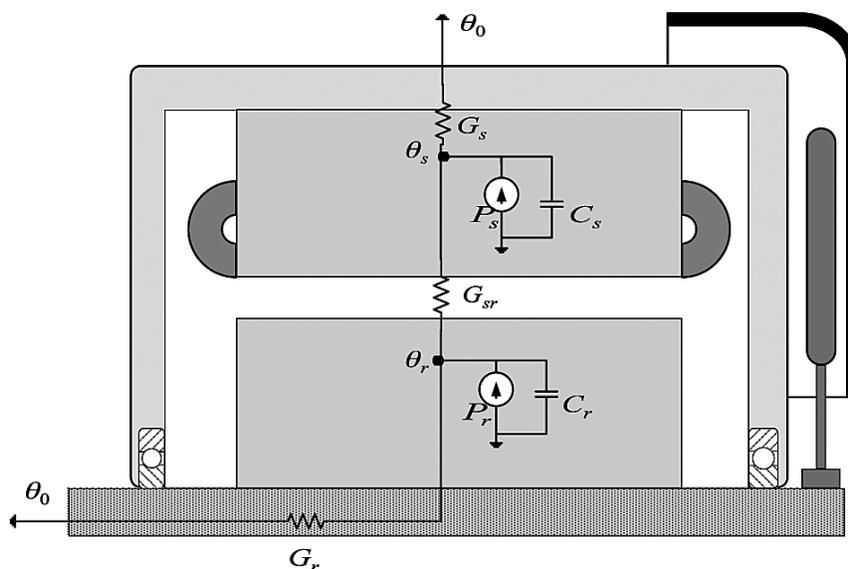
**Figura 2.** Algoritmo genético básico.

El algoritmo genético comienza con una población inicial de soluciones factibles, que se construye usualmente de forma aleatoria. Se evalúa la aptitud de cada individuo de la población y se construye nueva población mediante cruzamiento, mutación y selección. Este proceso se repite hasta que se satisfaga algún criterio de parada, como se muestra en la figura 2.

Con respecto a la convergencia de un algoritmo genético, Rudolph (Rudolph, 1994), demuestra que las variantes de un algoritmo genético básico que mantengan en la población, antes o después de la selección, la mejor solución hallada a lo largo del tiempo, siempre converge al óptimo global. Esto no se puede garantizar cuando se utiliza la selección elitista pues el mejor individuo no sólo se mantiene sino que también se utiliza para generar nuevos individuos y el algoritmo tiene una matriz de transición diferente y, por tanto, diferentes dinámicas de búsqueda que pueden ser mejores en algunos casos y peores en otros.

### Modelo Térmico Propuesto

El modelo térmico que se utiliza para el motor de estudio fue propuesto por (Duran y Fernandez, 2004; Moreno, Hidalgo y Martínez, 2001). Este modelo se construye a partir de la ecuación general (4). El modelo de la figura 3, posee sólo dos nodos llamados  $\theta_s$  y  $\theta_r$  que representan la temperatura promedio del estator y del rotor respectivamente. Por otra parte, la temperatura  $\theta_o$  se refiere a la temperatura ambiental. El modelo representa las pérdidas de potencia del motor como fuentes de calor que elevan las temperaturas internas. La fuente del estator es llamada  $P_s$  y la del rotor  $P_r$ . Los capacitores térmicos  $C_s$  y  $C_r$  representan la capacidad del estator y el rotor de almacenar calor, son el producto de la masa del componente por el calor específico del material.



**Figura 3.** Modelo térmico de un motor eléctrico de inducción. Adaptado de Duran y Fernandez, (2004).

### Estimación de las pérdidas

El incremento de temperatura dentro de un motor de inducción es causado por la energía que no se logra transformar por unidad de tiempo y se desperdicia como calor, es decir las pérdidas son resultado de un deficiente proceso de conversión de energía eléctrica a mecánica. Por lo tanto, la pérdida total  $P_r$  es la diferencia de potencia de la entra  $P_{Ent}$  menos la potencia de salida  $P_{Sal}$  de la máquina. Estas pérdidas usualmente se agrupan en tres clases: las llamadas pérdidas fijas  $P_f$ , las pérdidas variables  $P_v$  y pérdidas misceláneas  $P_{st}$ .

$$P_T = P_{Ent} - P_{Sal} = P_f + P_v + P_{st} \quad (7)$$

Las pérdidas fijas  $P_f$  son aquellas que se asocian a las pérdidas en el núcleo del motor  $P_N$  más las pérdidas mecánicas  $P_M$ , ecuación (8). La pérdida  $P_N$  se compone de la pérdida por histéresis  $P_H$  más la pérdida que producen las corrientes parásitas  $P_F$ . Las pérdidas mecánicas  $P_M$  se componen a su vez de las pérdidas por fricción en rodamientos  $P_R$  y por las pérdidas por vendaval  $P_V$ . De acuerdo con Fraile-Mora (2008) las potencia por pérdida en núcleo y mecánica se modelan como :

$$P_f = P_N + P_M \quad (8)$$

$$P_N = P_H + P_F = (k_H f B_m^\alpha + k_F f^2 B_m^2 a^2 \sigma) v \quad (9)$$

$$P_M = P_R + P_V = A\omega + B\omega^3 \quad (10)$$

Para la ecuación (9) las constantes  $k_H$  y  $k_F$  representan los efectos de histéresis y de Foucault respectivamente,  $f$  es la frecuencia,  $B_m$  es el campo magnético máximo, los parámetros  $\alpha, a, \sigma, v$  representan las contantes de diseño: el espesor de placas, la conductividad y volumen del hierro respectivamente. Para la ecuación (10)  $A, B$  son contantes y  $\omega$  representa la velocidad del eje.

Dado que en la ecuación (9) todos los parámetros son constantes a excepción de la frecuencia de funcionamiento y el campo magnético, pero estos son prácticamente constantes, entonces se dice que la potencia  $P_N$  es constante en la curva de bajo deslizamiento. Eventualmente el contenido armónico de la señal de voltaje incrementa estas pérdidas, ya que existirán frecuencias múltiplos de  $f$  que generan campos magnéticos residuales que se suman a las pérdidas provocada en la frecuencia fundamental. De igual forma sucede con las pérdidas mecánicas (10), la velocidad de régimen es prácticamente constante, y varía muy poco respecto a la velocidad en vacío. Las pérdidas fijas se obtienen aplicando la prueba en vacío que aparece en la norma IEEE 112 (“IEEE Standard Test Procedure for Polyphase Induction Motors and Generators” 2004), y la ecuación siguiente para cada voltaje:

$$P_f = P_{Ent0} - P_{Cu0} \quad (11)$$

donde  $P_{Ent0}$  es el consumo de potencia del motor cuando este no posee carga en el eje y  $P_{Cu0}$  se refiere a las pérdidas por efecto Joule de la resistencia por fase  $R_e$  del devanado estatórico, de tal forma que para un sistema que presenta corrientes con desbalance se puede calcular como:

$$P_{Cu0} = R_e (I_{A0}^2 + I_{B0}^2 + I_{C0}^2) \quad (12)$$

donde  $I_{A0}, I_{B0}, I_{C0}$  se refieren a las corrientes en vacío de cada una de las fases.

Las pérdidas variables  $P_v$  son todas aquellas que dependen del porcentaje de carga del motor y por tanto de la corriente que pasa por los conductores. Estas pérdidas son producto del efecto Joule en la resistencia del devanado estatórico  $P_{CuE}$ , así como o en las barras del rotor  $P_{CuR}$ .

$$P_{CuE} = R_e (I_A^2 + I_B^2 + I_C^2) \quad (13)$$

donde  $R_e$  se refiere a la resistencia a temperatura ambiente y  $I_A, I_B, I_C$  a las corrientes por fase.

La potencia de pérdida del rotor  $P_{CuR}$  se calcula como:

$$P_{CuR} = P_{Ent} - P_{Sal} - P_{CuE} - P_f - P_{st} \quad (14)$$

donde  $P_{st}$  se refiere a la potencia miscelánea que según la norma “IEEE Standard Test Procedure for Polyphase Induction Motors and Generators” (2004) se estima como el 1.8% de la potencia

de salida para máquinas menores a 90kW. Otra forma de estimar esta potencia es conociendo el deslizamiento en el punto de operación de la máquina y la potencia de pérdida del núcleo  $P_N$ . Para máquinas de 4 polos la pérdidas mecánicas  $P_M$  no superan el 1.5% de la potencia de salida según Kim, Choi, Chun, Koo, y Han, (2014), por lo que la potencia de pérdida del rotor se puede calcular como,

$$P_{CuR} = (P_{Ent} - P_{CuE} - (P_f - P_M))s \quad (15)$$

De acuerdo con el modelo planteado en la sección anterior, la potencia que se pierde en el estator  $P_s$  y en el rotor  $P_r$  se calculan como:

$$P_s = (P_{CuE} + (P_f - P_M)) \quad (16)$$

$$P_r = P_{CuR} \quad (17)$$

### Deducción del modelo

A partir de la ecuación (4) y del modelo térmico de la figura 3, se obtiene el sistema de ecuaciones diferenciales de primer orden, con sus condiciones iniciales:

$$\begin{bmatrix} C_s & 0 \\ 0 & C_r \end{bmatrix} \begin{bmatrix} \frac{d\theta_s}{dt} \\ \frac{d\theta_r}{dt} \end{bmatrix} = \begin{bmatrix} P_s \\ P_r \end{bmatrix} - \begin{bmatrix} (G_s + G_{sr}) & -G_{sr} \\ -G_{sr} & (G_r + G_{sr}) \end{bmatrix} \begin{bmatrix} \theta_s \\ \theta_r \end{bmatrix} \quad (18)$$

$$\theta(0) = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Donde dichas condiciones significan que el motor se encuentra a temperatura ambiente, es decir  $\theta_s(0) = 0$  y  $\theta_r(0) = 0$ .

Se solucionó de forma analítica la ecuación (18) y se obtuvieron las ecuaciones (19) que modelan la temperatura del estator  $\theta_s$  y del rotor  $\theta_r$  en función de los parámetros térmicos.

$$\begin{aligned} \theta_s &= C_1 e^{r_1 t} + C_2 e^{r_2 t} + \frac{E}{B} \\ \theta_r &= C_3 e^{r_1 t} + C_4 e^{r_2 t} + \frac{F}{B} \end{aligned} \quad (19)$$

donde las constantes de (19) en función de los parámetros son:

$$\begin{aligned} C_1 &= \frac{E(r_2 C_s + G_s + G_{sr}) - F G_{sr}}{B C_s (r_1 - r_2)} & C_2 &= \frac{E(r_1 C_s + G_s + G_{sr}) - F G_{rs}}{B C_s (r_2 - r_1)} \\ C_3 &= \frac{C_1 (r_1 C_s + G_s + G_{sr})}{G_{sr}} & C_4 &= \frac{C_2 (r_2 C_s + G_s + G_{sr})}{G_{sr}} \\ B &= \frac{G_s G_r + G_s G_{sr} + G_r G_{sr}}{C_s C_r} & E &= \frac{G_r P_s + G_{sr} P_s + G_{sr} P_r}{C_s C_r} \\ F &= \frac{G_s P_r + G_{sr} P_r + G_{sr} P_s}{C_s C_r} \end{aligned}$$

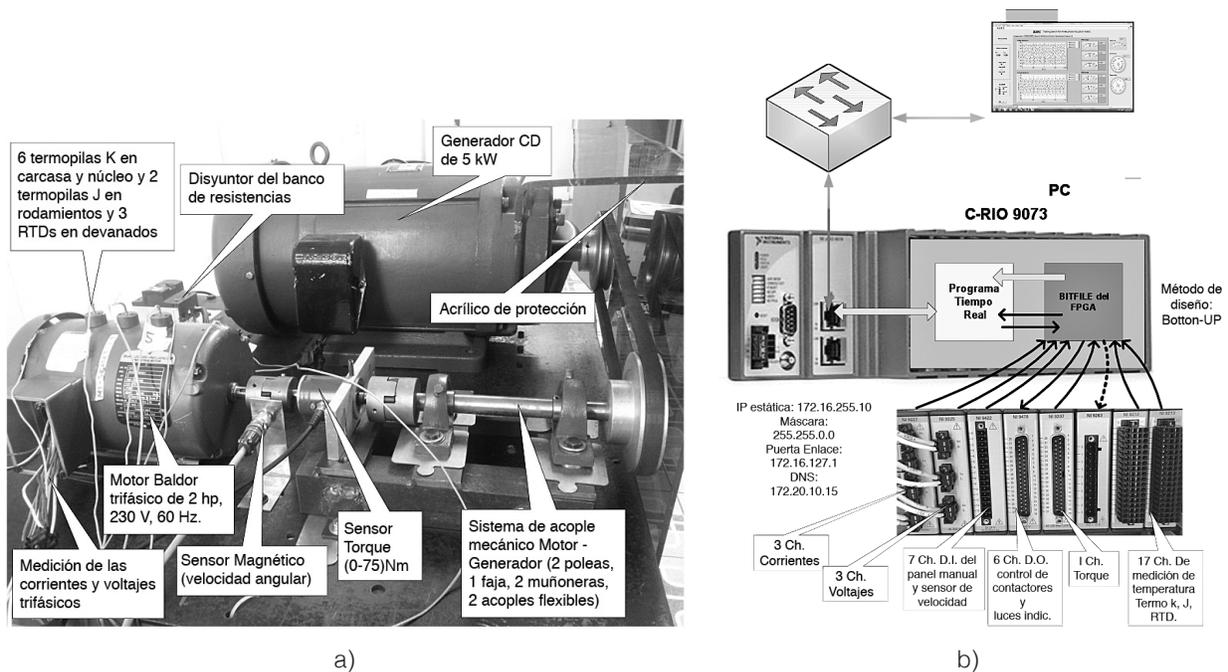
*Donde  $r_1, r_2$  son las soluciones de la ecuación característica asociada.*

## Banco de pruebas y adquisición de datos

El banco de pruebas para motores trifásicos se compone de un generador de corriente directa de 5 kW y un banco de resistencias de 7 kW. El banco posee tres autotransformadores monofásicos conectados en estrella para provocar bajos y altos voltajes de alimentación así como desbalances, finalmente poleas y fajas para acoplar mecánicamente el motor al generador tal como se aprecia en la figura 4a.

Se ha desarrollado un software (Murillo-Soto, 2014) que gestiona el banco de pruebas y somete los motores a experimentos controlados en los que se definen los parámetros de la prueba. Se controlan los estímulos del motor (carga, tiempos de encendido o apagado), se monitorea su reacción y finalmente se registran todas las variables de interés.

La aplicación informática se desarrolló desde el paradigma de programación gráfica usando *LabView* v12.0. La aplicación posee una arquitectura cliente-servidor punto a punto y se compone de dos programas uno que se ejecuta en una computadora personal y el otro que se ejecuta en un controlador de tiempo real C-RIO 9073 de National Instruments, tal como se muestra en la figura 4b. Los siete módulos que tiene instalado el C-RIO instalados son: un NI-9227 para corrientes trifásicas, un NI-9225 para voltajes trifásicos, un NI-9422 para entradas digitales, un NI-9478 para salidas digitales, un NI-9207 para entradas analógicas de voltaje, un NI-9263 para salidas analógicas, un módulo NI-9219 universal para RTD y un NI-9213 para termopilas j y k.



**Figura 4.** a) Fotografía del banco de pruebas implementado b) Flujo de datos entre la interfaz gráfica y la tarjeta de adquisición de datos, las flechas indican la dirección del flujo de información. Tomado de Murillo-Soto, (2014).

El programa que corre en la PC se conecta mediante una red Ethernet usando el protocolo TCP/IP al controlador C-RIO 9073. Este software controla el encendido y apagado del motor seleccionado, muestra las señales trifásicas de corrientes, voltajes y brinda los valores de torque y la velocidad angular del eje. Además muestra en una gráfica las 11 mediciones de

temperaturas del motor segregadas de la siguiente forma: las temperaturas de cada devanado del motor trifásico, la medición de temperatura de cada rodamiento, dos temperaturas en el núcleo, tres temperaturas en la carcasa y la temperatura ambiental.

Además el software de la PC calcula los valores RMS de corriente y voltaje trifásicos, así como el contenido espectral de las señales. Adicionalmente se calculan los ángulos de fase, las potencias de entrada y salida del motor, calcula y grafica la evolución de la eficiencia de la máquina, calcula el desbalance de la red, el contenido armónico de voltaje y corrientes, así como las potencias reactivas, aparentes y el factor de potencia. Este software permite exportar todas las 28 variables capturadas en dos tipos de archivos de extensión lvm.

El software que se ejecuta en el C-RIO configura cada uno de los ocho módulos del sistema, su frecuencia de muestro, la sincronización entre módulos de adquisición, el tratamiento de los datos (filtrado), la conversión de datos crudo a variables con significado físico (escalamiento), etc. El controlador manipula salidas digitales para señalización externa, control de contactores y control de un relé de estado sólido mediante modulación de ancho de pulso para el control de la carga eléctrica del generador, que su vez se traduce en carga mecánica en el motor. El software del C-RIO tiene programada la protección de sobrecarga, la protección a un determinado valor de desbalance del voltaje y la protección a valores de contenido armónico superiores a un umbral definido por el usuario.

### Cálculo del vector de pérdidas

El motor en estudio es de la marca Baldor modelo M3558 de 2hp, 230 V, 60 Hz. Lo primero que se realizó son las mediciones de las resistencias estáticas con un miliohmímetro, el cual arroja los valores por fase de  $R_e=1.03\Omega$ . Luego se realizó la prueba del motor en vacío la cual permite calcular las pérdidas fijas  $P_f$  ecuación (8), esta prueba consiste en desacoplar la carga mecánica y energizar la máquina con el fin de registrar la potencia de entrada  $P_{Ent0}$  y las corrientes de vacío por fase  $\{I_{A0}, I_{B0}, I_{C0}\}$ . Con dichas corriente se calcula la pérdida de potencia por efecto Joule en vacío, ecuación (12). Luego se reduce el voltaje y para cada cambio se registran la potencia de entrada en vacío y los nuevos valores de corrientes consumida. De esta forma aplicando las ecuaciones (11) y (12) se obtiene la potencia fija para cada voltaje, y se grafica como se muestra en la figura 5. Por lo tanto, las pérdidas de potencias obtenidas para voltaje nominal son:  $P_f=204,3 W$ ,  $P_N=200,4 W$  y  $P_M=3,9 W$  respectivamente.

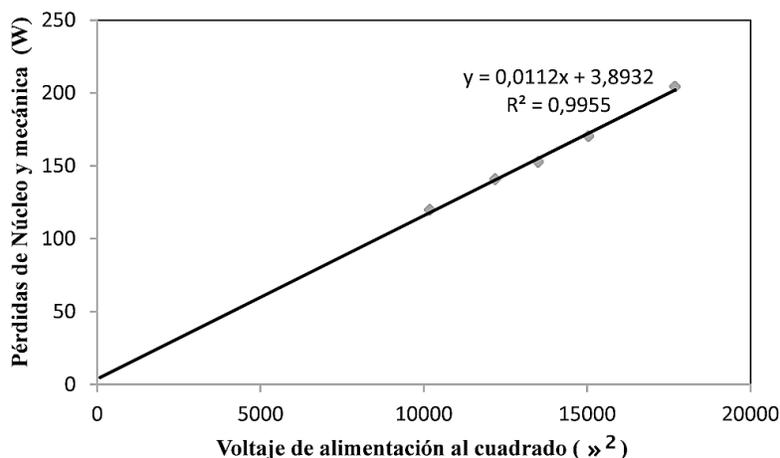


Figura 5. Pérdidas fijas del motor de 2 hp, 230 V, 60 Hz del laboratorio.

Como tercer paso y dado que el sistema de adquisición de datos calcula las potencias de entrada y salida en cada segundo para una carga dada, se utiliza el promedio de los valores registrados. Por ejemplo la ecuación (13) se modifica y se convierte en la ecuación (20), esto con el fin de utilizar los  $N$  registros. El valor calculado fue de  $P_{CuE}=109.4 W$ .

$$P_{CuE} = \frac{R_e}{N} \sum_{i=1}^N (I_{A_i}^2 + I_{B_i}^2 + I_{C_i}^2) \quad (20)$$

Con los valores de pérdidas hasta ahora obtenidos y usando la ecuación (16) se obtiene una potencia de pérdida del estator  $P_s$ , de 309.8 W.

La potencia promedio de entrada es de  $P_{Ent} = 1855.6 W$  y la potencia promedio de salida es de  $P_{Sal} = 1480.3 W$ , por lo que la potencia total de pérdidas ecuación (7), es de 375.3 W. Para el cálculo de la potencia de pérdidas del rotor se utilizó la expresión (15) ya que se conoce que el deslizamiento promedio es de 3.4%, por lo que  $P_r = P_{Cur} = 52.7 W$ . La potencia miscelánea  $P_{st}$  para este motor sería de 12.8 W. El vector de pérdidas  $P$  del modelo de la ecuación (18) posee por lo tanto los valores:  $P = [309.8, 52.7]^T W$ .

## Identificación de parámetros

Para estimar las conductancias térmicas  $G$  y las capacitancias térmicas  $C$  la literatura propone una serie de técnicas por ejemplo: Duran y Fernández, (2004); Moreno, Hidalgo y Martínez, (2001), Boglietti, Cavagnino y Staton, (2008). Alternativamente usando un algoritmo genético es posible estimar dichos parámetros sin mayores complicaciones como se describe en esta sección. Para estimar los parámetros  $G = [G_s, G_r, G_{sr}]^T$  y  $C = [C_s, C_r]^T$  se compara la respuesta  $\theta_s$  del modelo de referencia basado en la ecuación (19), con el incremento de temperatura en el estator  $\Delta\theta$ . El incremento  $\Delta\theta$  se entiende como la diferencia entre las temperaturas medidas en el devanado eléctrico  $\theta$  y la temperatura ambiente  $\Delta\theta = \theta - \theta_0$ . Los parámetros del modelo de referencia se van ajustando iterativamente mediante un algoritmo genético, de manera tal que la diferencia entre las respuestas de ambos modelos sea mínima.

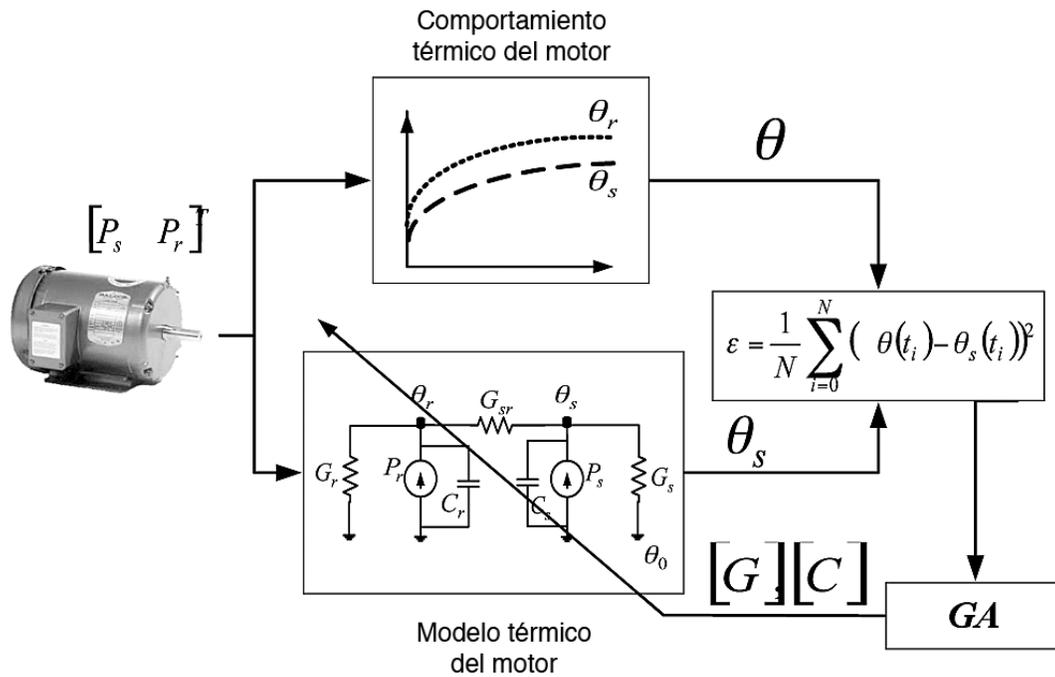
La figura 6 muestra un diagrama de bloques donde se ilustra dicho proceso. El modelo de referencia se alimenta con el vector de pérdidas  $P = [P_s, P_r]^T$  y los vectores de parámetros a estimar  $C = [C_s, C_r]^T$  y  $G = [G_s, G_r, G_{sr}]^T$  generados por el algoritmo genético. Luego se compara su respuesta con las medidas de temperatura del estator, se ajustan de nuevo los parámetros y se realimenta el modelo. El proceso de optimización se detiene cuando no hay cambios significativos en los vectores de parámetros  $C$  y  $G$ .

La función de incremento de temperatura en el estator  $\Delta\theta = \theta - \theta_0$  se puede estimar con precisión, pues se tiene la temperatura en el devanado y la temperatura ambiente, las cuales han sido medidas cada segundo a lo largo de un período de 2 horas 30 minutos. De esta forma la función objetivo a optimizar está dada por la función de error cuadrático medio  $\varepsilon = \frac{1}{N} \sum_{i=0}^N (\Delta\theta(t_i) - \theta_s(t_i))^2$  la cual se obtiene restando los datos experimentalmente medidos  $\Delta\theta$  y la respuesta del modelo térmico  $\theta_s$ .

Para optimizar la función objetivo  $\varepsilon$  con el algoritmo genético, se parte de una población inicial de  $m=100$  individuos que son generados de forma aleatoria en el espacio de búsqueda. Cada individuo de la población es un vector  $X^k = [C_s, C_r, G_s, G_r, G_{sr}]^T$  para  $k=1, \dots, m$ . El espacio de búsqueda se restringió con las siguientes condiciones  $0 \leq x_i^k \leq 100000$  para  $i=1, \dots, 5$ .

El operador de cruzamiento combina la información de dos individuos llamados padres para formar, en la siguiente generación, un nuevo individuo llamado hijo. Existen diferentes tipos de

cruzamiento, se utilizó el llamado cruzamiento heurístico el cual construye el nuevo individuo de la siguiente forma: si  $P_1$  y  $P_2$  son los padres tales que  $P_1$  es el padre mejor adaptado, entonces el nuevo individuo se construye como  $H=P_2+R(P_1-P_2)$ , donde  $R$  es la razón de cruzamiento, en nuestro caso  $R=1.2$ .



**Figura 6.** Diagrama de bloques para la identificación de parámetros del modelo térmico.

El operador de mutación realiza aleatoriamente pequeños cambios en los individuos de una población, lo cual provee diversidad genética y amplía el rango de búsqueda. Existen diferentes tipos de mutación, se utilizó la mutación factible adaptativa, la cual genera direcciones de búsqueda aleatorias que son adaptadas de acuerdo al éxito o fracaso de las últimas generaciones. A lo largo de cada dirección se escoge una longitud de avance de modo que se cumplan las restricciones lineales y cotas del problema.

La función de selección elige, tomando en cuenta los valores de aptitud de cada individuo, los padres que se reproducirán para formar la siguiente generación. En nuestro caso la función de selección que se utilizó es el muestreo universal estocástico. Para esta función los individuos son mapeados a un segmento continuo de línea de tal manera que la porción del segmento que se les asigna es proporcional a su aptitud. Se colocan sobre la línea marcas igualmente espaciadas, tantas como padres se quiera seleccionar. El algoritmo se mueve a lo largo de la línea con un paso de avance de igual tamaño. El primer paso de avance es un número aleatorio uniforme menor que las distancias entre las marcas. Los individuos seleccionados son aquellos que están bajo las marcas.

El algoritmo genético se detiene si el cambio acumulado en el valor de la función objetivo a lo largo de las generaciones es menor que la tolerancia de funciones  $TolFun=10^{-15}$  o si se alcanza el número máximo de 80000 generaciones que se utilizó. Luego de ejecutar el algoritmo genético para los datos de entrada se obtuvieron los resultados de el cuadro 1.

**Cuadro 1.** Parámetros térmicos del motor de 2 Hp, 230V, 60Hz.

Símbolo	Valor
$C_s$	5243.41 J/°C
$C_r$	5389.43 J/°C
$G_s$	4.15 W/°C
$G_r$	$3 \times 10^{-8}$ W/°C
$G_{sr}$	12.16 W/°C
Mejor aptitud	0.5317
Aptitud promedio	0.5350

## Validación

El motor de estudio posee una masa de 16 kg de los cuales 10 kg pertenecen al estator y carcasa y el resto al rotor y al eje. Una primera aproximación para ver el orden de magnitud de las capacitancias es aplicar las fórmulas:

$$C_s = m_{s,ir} \cdot c_{s,ir} + m_{s,cu} \cdot c_{s,cu} \quad (21)$$

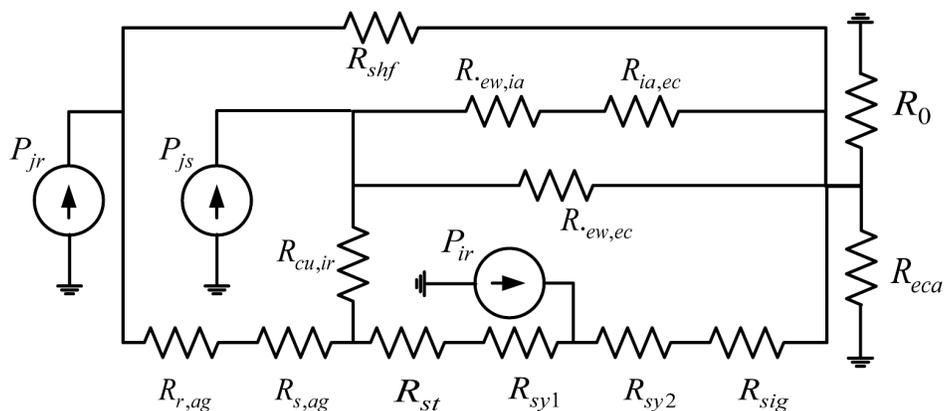
$$C_r = m_{r,ir} \cdot c_{s,ir} + m_{r,al} \cdot c_{s,cu} \quad (22)$$

donde la masa del acero del estator  $m_{s,ir}$  es de 8 kg, la masa del cobre en el estator  $m_{s,cu}$  es de 2 kg, la masa del acero en el rotor  $m_{r,ir}$  es 5,4 kg y la masa del aluminio en el rotor es de 0,6 kg. Los calores específicos del acero, cobre y aluminio son  $[0.50.390.9]^T$  J/(g°C) respectivamente. Aplicando (21) y (22) el vector  $[C_s C_r]^T$  es igual a  $[4780 \ 3240]^T$  J/°C.

Las resistencias térmicas se calcularon usando las fórmulas que propone Boglietti, Cavagnino, Lazzari, y Pastorelli, (2003), donde el cálculo de las resistencias térmicas se realiza a partir del conocimiento de las mediciones internas del motor, los materiales y constantes universales. Con esto se plantea la descomposición del motor en varios cilindros concéntricos y se calcula la resistencia radial para cada cilindro. El cuadro 2 muestra los resultados de las resistencias térmicas del modelo de la figura 7.

**Cuadro 2.** Valores de las resistencias térmicas obtenidas según Boglietti et al., (2003)

$R_{eca}$	0.5116 °C/W	Resistencia por convección forzada entre la carcasa y el ambiente
$R_o$	0.3063 °C/W	Resistencia por convección natural entre la carcasa y el ambiente
$R_{sy1}$	0.0041 °C/W	Resistencia por conducción radial de la parte baja del núcleo
$R_{sy2}$	0.0041 °C/W	Resistencia por conducción radial de la parte alta del núcleo
$R_{st}$	0.0291 °C/W	Resistencia por conducción de diente del estator
$R_{cu,ir}$	0.01796 °C/W	Resistencia por conducción entre cobre y el diente del núcleo
$R_{ew,ec}$	4.2265 °C/W	Resistencia por conducción entre devanado externo y carcasa
$R_{airgap}$	0.0888 °C/W	Resistencia del entrehierro
$R_{shf}$	37.4184 °C/W	Resistencia axial por conducción del eje
$R_{sig}$	0 °C/W	Resistencia entre el núcleo y la carcasa
$R_{ia,ec}$	$\infty$ °C/W, Sin datos	Resistencia por convección entre aire interno y tapas
$R_{ew,ia}$	$\infty$ °C/W, Sin datos	Resistencia por convección entre devanado externo y aire interno



**Figura 7.** Modelo térmico propuesto por Boglietti et al., (2003).

A partir del análisis del modelo de la figura 7, se plantean las ecuaciones para determinar las conductancias térmicas del modelo simplificado representado por (18).

$$G_s = \frac{1}{\left( \frac{1}{R_{cu,ir} + R_{st} + R_{sy1} + R_{sy2}} + \frac{1}{R_{ew,ec}} + \frac{1}{R_{ia,ec} + R_{ew,ia}} \right)^{-1} + \left( \frac{1}{R_0} + \frac{1}{R_{eca}} \right)^{-1}} \quad (23)$$

$$G_{sr} = \frac{1}{R_{r,ag} + R_{r,ag}} = \frac{1}{R_{airgap}} \quad (24)$$

$$G_r = \frac{1}{R_{shf} + \left( \frac{1}{R_0} + \frac{1}{R_{eca}} \right)^{-1}} \quad (25)$$

Las conductancias térmicas  $G_s, G_r, G_{sr}$  se calculan usando las tres ecuaciones anteriores y los datos del cuadro 2. Los valores calculados son  $G=[4.06 \ 0.026 \ 11.26]^T \text{ W/}^\circ\text{C}$ .

### Análisis de resultados

El cálculo de las capacitancias y conductancias térmicas realizadas en la sección anterior responden a una aproximación que utiliza las dimensiones del motor, sus materiales y constantes físicas, etc. Por lo tanto, es esperable que existan errores dado que no se conocían algunas constantes constructivas del motor y se debieron estimar según la literatura (Boglietti et al., 2003). Los parámetros obtenidos sirven para comparar el orden de las magnitudes con los resultados del algoritmo genético. Los errores obtenidos entre los parámetros del algoritmo genético y los parámetros calculados por las ecuaciones de la (23) a la (25) se muestra en el cuadro 3.

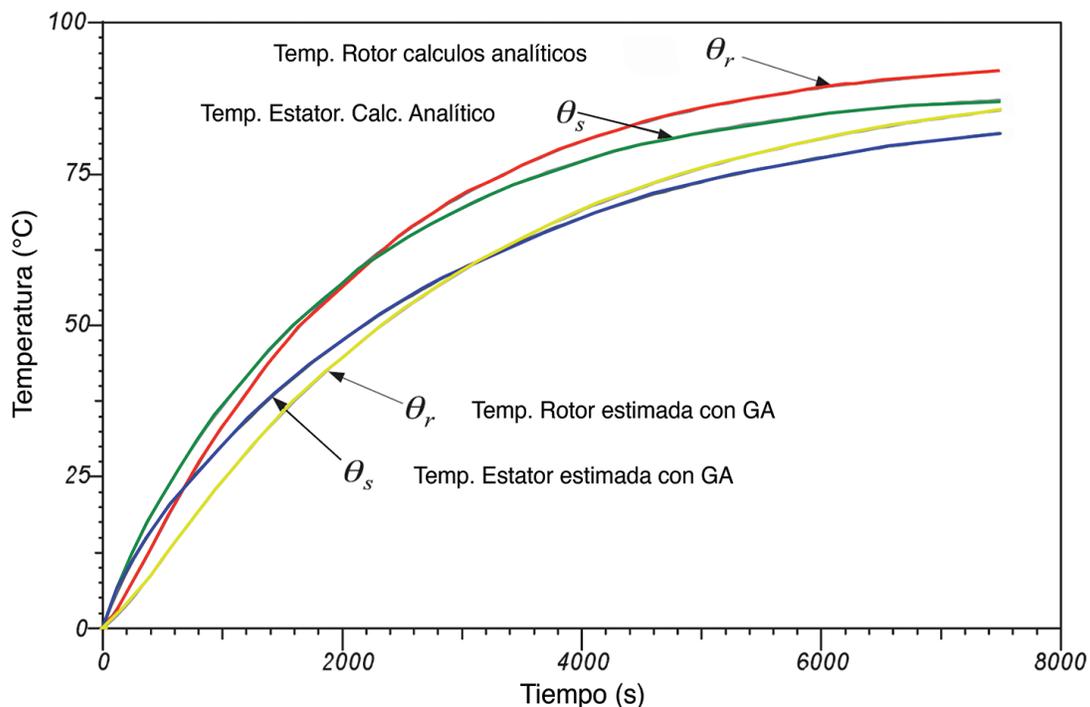
**Cuadro 3.** valor absoluto del error en los Parámetros térmicos

Símbolo	Error relativo al A.G
$C_s$	8.8%
$C_r$	39.8%
$G_s$	2.2%
$G_r$	100%
$G_{sr}$	7.4%

Se observa que existen dos parámetros con errores altos  $C_r$  y  $G_r$  ambos relacionados con el rotor. La capacitancia del rotor obtenida por el algoritmo genético es mayor que la obtenida por la ecuación (22), esto se debe principalmente a que esta ecuación no contempla el almacenamiento de calor del aire confinado en motores sellados como el motor en estudio, por lo que si se contemplara este hecho en dicha ecuación, la capacitancia tendería a ser mayor y, por tanto, a asemejarse a la obtenida por el GA. Por otra parte la conductancia térmica del rotor obtenida por el GA indica que es prácticamente cero, lo cual se interpreta como que el calor producido por el rotor fluye a través del entrehierro y no por el eje del motor hacia el ambiente. Este resultado si concuerda por el obtenido con la ecuación (25) que da un valor de  $0,026 \text{ W/}^\circ\text{C}$ , sin embargo cuando se calcula el error relativo este da un valor de 100%, esto se debe a que cualquier número respecto a un valor de casi cero tendrá un error muy alto como en este caso.

El modelo obtenido por las ecuaciones (19) es alimentado con los parámetros obtenidos por el GA cuadro 1 y por los parámetros analíticos obtenidos por las ecuaciones (21) a la (25). Ambos conjuntos de parámetros generan para el mismo modelo respuestas dinámicas distintas tal y como se aprecia en la figura 8, obteniéndose unas diferencias máximas de las respuestas de temperaturas de  $11^\circ\text{C}$  en el transiente y de  $5^\circ\text{C}$  en estado estable.

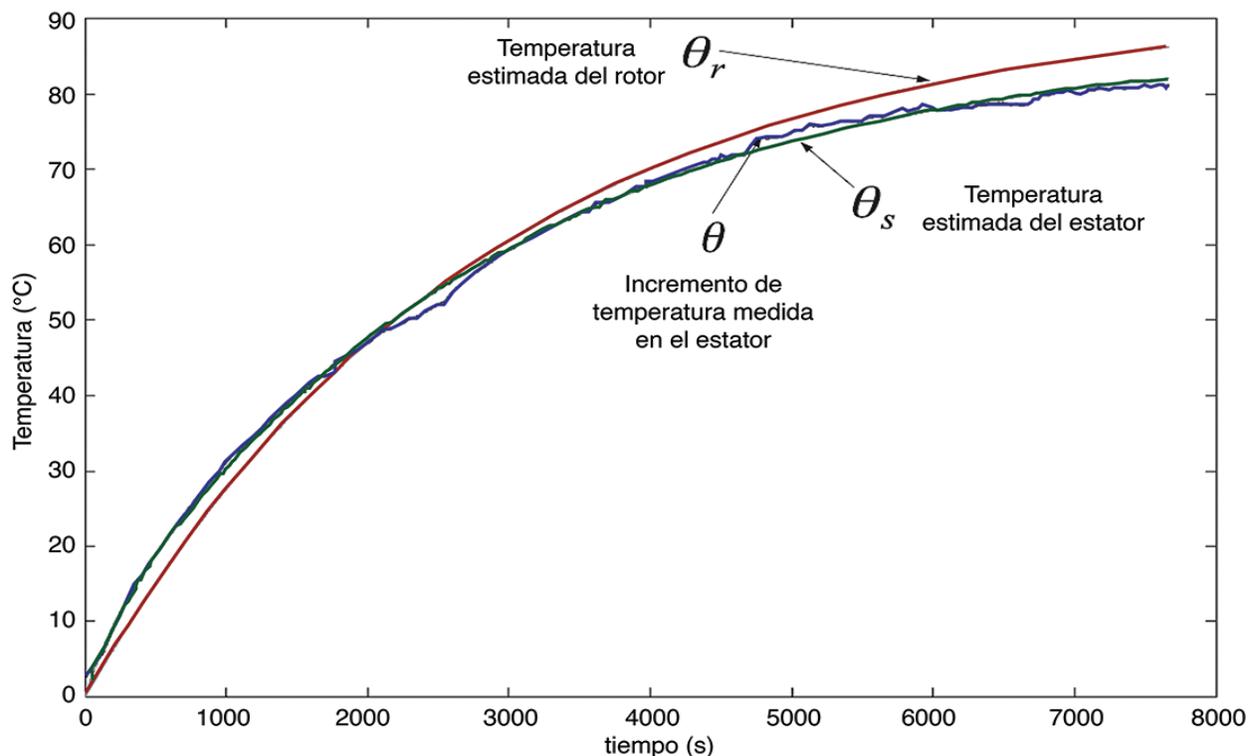
Finalmente, se rescata que el modelo térmico con los parámetros obtenidos por el GA presenta un mejor ajuste respecto a los datos experimentales  $\epsilon=0.5317$ , que el modelo con parámetros analíticos con ajuste de  $\epsilon=50.65$ , el ajuste del algoritmo genético se observa en la figura 9. Esto, como se dijo anteriormente se debe a que el algoritmo genera 80000 conjuntos de valores posibles y selecciona un conjunto con la mejor aptitud. Por lo tanto, el método para obtener los parámetros de un modelo térmico simple con algoritmos genéticos, es una opción eficiente en el sentido de que no se requiere conocer características físicas, materiales, etc., del motor en estudio.



**Figura 8.** Simulación de las temperaturas del rotor y estator para el modelo planteado (19) según los parámetros obtenidos por algoritmo genético y los obtenidos de forma analítica.

## Agradecimiento

Se agradece a la Vicerrectoría de Investigación del Instituto Tecnológico de Costa Rica por el financiamiento brindado al proyecto SEMAT, código VIE 5402-1341-1301, así como a los asistentes vinculados a este proyecto.



**Figura 9.** Ajuste de temperaturas entre la medida y la estimada.

## Bibliografía

- Boglietti, A., Cavagnino, A., Lazzari, M., & Pastorelli, M. (2003). A Simplified Thermal Model for Variable-Speed Self-Cooled Industrial Induction Motor. *IEEE Transactions on Industry Applications*, 39(4), 945–952. <http://doi.org/10.1109/TIA.2003.814555>
- Boglietti, A., Cavagnino, A., & Staton, D. (2008). Determination of Critical Parameters in Electrical Machine Thermal Models. *IEEE Transactions on Industry Applications*, 44(4), 1150–1159. <http://doi.org/10.1109/TIA.2008.926233>
- Boldea, I., & Nasar, S. A. (2002). *The Induction Machine Handbook*. CRC Press.
- Cho, K., & Seok, J. (2008). Induction Motor Temperature Estimation Based on High-Frequency Model of Rotor Bar. In *Industry Applications Society Annual Meeting, 2008. IAS '08. IEEE* (pp. 1–7). <http://doi.org/10.1109/08IAS.2008.208>
- Duran, M. J., & Fernandez, J. (2004). Lumped-Parameter Thermal Model for Induction Machines. *IEEE Transactions on Energy Conversion*. <http://doi.org/10.1109/TEC.2004.837272>
- Dutta, B., & Chowdhury, S. K. (2012). Steady state thermal model of TEFC induction machine. In *IEEE International Conference on Emerging Technologies and Factory Automation, ETFAPower Electronics, Drives and Energy Systems (PEDES), 2012* (pp. 1–6). Bengaluru. <http://doi.org/10.1109/PEDES.2012.6484344>
- Fraile-Mora, J. (2008). *Máquinas Eléctricas* (6th ed.). España: McGraw-Hill Interamericana de España.

- Gao, Z. (2006). *SENSORLESS STATOR WINDING TEMPERATURE ESTIMATION FOR INDUCTION MACHINES*. Georgia Institute of Technology.
- Gao, Z., Habetler, T. G., Harley, R. G., & Colby, R. S. (2005). An Adaptive Kalman Filtering Approach to Induction Machine Stator Winding Temperature Estimation Based on a Hybrid Thermal Model. *IEEE Industry Applications Conference*, 2–9. <http://doi.org/10.1109/IAS.2005.1518284>
- IEEE Standard Test Procedure for Polyphase Induction Motors and Generators. (2004). *IEEE Std 112-2004 (Revision of IEEE Std 112-1996)*, 0\_1–79. <http://doi.org/10.1109/ieeestd.2004.95394>
- Kim, D. J., Choi, J. H., Chun, Y. Do, Koo, D. H., & Han, P. W. (2014). The study of the stray load loss and mechanical loss of three phase induction motor considering experimental results. *Journal of Electrical Engineering and Technology*, 9, 121–126. <http://doi.org/10.5370/jeet.2014.9.1.121>
- Kylander, G. (1995). *Thermal modelling of small cage induction motors*. Chalmers University of Technology. Retrieved from <http://publications.lib.chalmers.se/records/fulltext/1335/1335.pdf>
- Moreno, J. F., Hidalgo, F. P., & Martínez, M. D. (2001). Realisation of tests to determine the parameters of the thermal model of an induction machine. *IEE Proceedings - Electric Power Applications*. <http://doi.org/10.1049/ip-epa:20010580>
- Murillo-Soto, L. D. (2014). Aplicación informática para el monitoreo de variables eléctricas , mecánicas y térmicas en motores trifásicos de inducción. In *Latin American and Caribbean Conference for Engineering and Technology (LACCEI'2014)*. Guayaquil: EBSCO. Retrieved from <http://www.laccei.org/LACCEI2014-Guayaquil/ExtendedAbstracts/EA026.pdf>
- Okoro, O. I., Weidemann, B., & Ojo, O. (2004). An efficient thermal model for induction machines. *Conference Record of the 2004 IEEE Industry Applications Conference, 2004. 39th IAS Annual Meeting.*, 4. <http://doi.org/10.1109/IAS.2004.1348823>
- Oraee, H. (2000). A quantitative approach to estimate the life expectancy of motor insulation systems. *IEEE Transactions on Dielectrics and Electrical Insulation*, 7(6), 790–796. <http://doi.org/10.1109/94.891990>
- Pillay, P., & Manyage, M. (2006). Loss of life in induction machines operating with unbalanced supplies. *IEEE Transactions on Energy Conversion*, 21(4), 813–822. <http://doi.org/10.1109/TEC.2005.853724>
- Rudolph, G. (1994). Convergence analysis of canonical genetic algorithms. *IEEE Transactions on Neural Networks*, 5, 96–101. <http://doi.org/10.1109/72.265964>
- Shenkman, a. L., & Chertkov, M. (2000). Experimental method for synthesis of generalized thermal circuit of polyphase induction motors. *IEEE Transactions on Energy Conversion*, 15(3), 264–268. <http://doi.org/10.1109/60.875491>
- Talbi, E. G. (2009). *Metaheuristics: From Design to Implementation*. *Metaheuristics: From Design to Implementation*. <http://doi.org/10.1002/9780470496916>
- Zocholl, S. E. (2004). Comparing motor thermal models. In *In 31st Annual Western Protective Relay Conference* (pp. 1–10). WA.
- Zocholl, S. E., Schweitzer, E. O., & Aliaga-Zegarra, A. (1984). Thermal Protection of Induction Motors Enhanced by Interactive Electrical and Thermal Models. *IEEE Transactions on Power Apparatus and Systems, PAS-103(7)*, 1749–1755. <http://doi.org/10.1109/TPAS.1984.318678>

# Análisis de un modelo SIS para el estudio de la dinámica de propagación de la enfermedad al aplicar medidas de control

Analysis of a SIS model to study the dynamics of disease spread when control measures are applied

Geisel Yajaira Alpízar-Brenes<sup>1</sup>

---

*Fecha de recepción: 19 de junio del 2015*

*Fecha de aprobación: 26 de setiembre del 2015*

Alpízar-Brenes, G. Análisis de un modelo SIS para el estudio de la dinámica de propagación de la enfermedad al aplicar medidas de control. *Tecnología en Marcha*. Edición especial. Matemática Aplicada, Mayo 2016. Pág 42-50.

---

<sup>1</sup> Escuela de Matemática, Instituto Tecnológico de Costa Rica. Correo electrónico: galpizar@itcr.ac.cr

## Palabras clave

Enfermedades infecciosas; modelos de transmisión de enfermedades; SIS; medidas de control.

## Resumen

En este artículo se analiza un modelo discreto metapoblacional SIS para determinar los parámetros que se pueden controlar y representen la aplicación de alguna medida de control, con el objetivo de estudiar su impacto en la dinámica de la enfermedad. Una vez definidos los parámetros controlables, se simulan casos específicos que se asemejen a diferentes escenarios que se pueden presentar en la vida real.

## Keywords

Infectious diseases; epidemic models; SIS; control measures.

## Abstract

In this paper a discrete-time metapopulation SIS model is analyzed to determine the parameters that can be controlled and they represent the implementation of some control measure, in order to study their impact on the dynamics of the disease. When the parameters are defined, different scenarios are simulated to represent various real-life cases.

## Introducción

La simulación a través de modelos matemáticos permite explorar el impacto de la aplicación de una o varias medidas de control en la dinámica de la transmisión de enfermedades infecciosas, proporcionando información valiosa para la toma de decisiones con el objetivo de controlarlas o erradicarlas.

Según Welte et al. (2010), hay dos maneras de lograr la eficacia general de un programa para controlar el brote de una enfermedad: un análisis retrospectivo de datos y la simulación utilizando modelos matemáticos. Diversas investigaciones demuestran que el uso de modelos matemáticos para describir la dinámica de transmisión de las enfermedades infecciosas son una herramienta necesaria para realizar un buen análisis de la relación costo-eficacia en la aplicación de medidas de control en la propagación de una enfermedad (Kretzschmar & Wallinga, 2010 y González et al., 2011).

La historia y eventos recientes demuestran que las enfermedades contagiosas constituyen una amenaza para la población mundial. Los países están cada vez más interconectados gracias a los modernos sistemas de transporte. Surgen enfermedades nuevas o reaparecen otras viejas que pueden atravesar fronteras y propagarse rápidamente por varias poblaciones. Así lo demostró la pandemia de gripe AH1N1, cuyo brote se detectó en abril de 2009 en México y al cabo de un mes se había extendido a 24 países (WHO, 2009). De estas y otras experiencias se puede concluir que un factor determinante en la propagación de una enfermedad es la movilidad intensa y rápida de la población. Es por esa razón que para este análisis se decidió usar un modelo discreto SIS metapoblacional. Conviene utilizar modelos en tiempo discreto, ya que los datos reales son dados de manera discreta y eso permite comparar los resultados (Brauer et al., 2014).

La propagación de enfermedades infecciosas y sus medidas de control han sido objeto de diversos estudios, la mayoría lo han hecho en modelos para la dinámica en una población. El objetivo de este trabajo es utilizar modelos metapoblacionales, estos modelos permiten conocer la influencia de los flujos migratorios en la propagación de una enfermedad y comprender las características de la propagación en subpoblaciones, cada una con su propia dinámica pero conectadas por el movimiento de personas entre sí.

## Modelo SIS metapoblacional

Los modelos SIS (susceptible-infectado-susceptible) son modelos epidemiológicos de compartimentos, que se definen a partir de clases y subclases. Como indican Brauer et al. (2014), las tasas de transición entre las clases para estos modelos se estiman a partir del conocimiento cualitativo y evidente en la historia natural de la enfermedad.

Para este estudio se eligió un modelo discreto metapoblacional de transmisión de enfermedades SIS. En este tipo de modelos, la población se divide en dos grupos de personas: las que han sido infectadas por la enfermedad y son infecciosas y las que son susceptibles de ser infectadas por la enfermedad.

Los modelos SIS se usan para enfermedades en las que no hay inmunidad, pues, una vez que las personas infectadas se recuperan, pasan a ser de nuevo susceptibles. Por lo tanto, la progresión de la enfermedad desde el punto de vista de un individuo es susceptible-infectado-susceptible.

El modelo SIS utilizado en este caso es una variante del planteado en Castillo-Chavez y Yakubu (2002) y además se ha considerado la aplicación de un tratamiento (Alpizar, 2012). En este modelo, la población se dividió en dos parches que, según Arino y Driessche (2006), pueden representar ciudades, regiones geográficas o comunidades que están interconectadas por los procesos de inmigración y emigración.

Se trabajó con el siguiente modelo SIS:

$$S_{k+1} = \gamma G_k S_k + \gamma(1-\sigma\tau)I_k$$

$$I_{k+1} = \gamma(1-G_k)S_k + \gamma\sigma\tau I_k$$

donde

$$G_k = \exp\left(\frac{-\alpha I_k}{N_k}\right)$$

es la probabilidad de seguir siendo susceptible en el tiempo  $k$  a  $k+1$  y  $\alpha$  es una constante que

pondera el papel de la prevalencia de  $\frac{I_k}{N_k}$  en la transmisión de la enfermedad en el momento  $k$ ,  $I_k$  son los infectados en el tiempo  $k$  y  $N_k$  es la población total en el tiempo  $k$ .

Supongamos que la población es constante  $N_k = S_k + I_k$  en cada momento  $k$ , es decir, no se considera la dinámica demográfica de la población. Por lo tanto, el cambio en el grupo de personas infectadas siempre es igual al cambio en el grupo de susceptibles.

Si  $\gamma$  es la probabilidad de sobrevivir a la enfermedad, entonces,  $\gamma G_k S_k$  es el grupo de personas que siguen siendo susceptibles en  $k+1$  y  $\gamma(1-G_k)S_k$  es la cantidad de personas que dejan de ser susceptibles y pasan a pertenecer al grupo de infectados.

Si  $(1-\sigma)$  es la probabilidad de recuperación natural (sin tratamiento) y  $(1-\tau)$  es la probabilidad de recuperación con tratamiento efectivo, entonces  $\gamma\sigma\tau I_k$  es el grupo de personas que siguen siendo infectadas en el tiempo  $k+1$  y  $\gamma(1-\sigma)I_k$  son los infectados que pasan a ser susceptibles por recuperación natural y  $\gamma(1-\tau\sigma)I_k$  es la cantidad de personas que no se recuperaron naturalmente pero sí por la aplicación del tratamiento; así, los infectados que pasan a ser susceptibles en  $k+1$  están dados por

$$\gamma(1-\sigma)I_k + \gamma(1-\tau\sigma)I_k = \gamma(1-\sigma\tau)I_k$$

Ahora consideremos una población con dos parches acoplados en donde la dinámica de cada uno de ellos está descrita por el modelo SIS anterior. Si denotamos con  $x$  cada parche, entonces  $x \in \{1,2\}$ . Sea  $S_k^x$  el grupo de susceptibles y  $I_k^x$  el grupo de infectados en el tiempo  $k$  y del parche  $x$ . En el parche  $x$  las personas sobreviven con probabilidad  $\gamma^x$ , mientras que los

infectados se recuperan naturalmente con probabilidad  $(1-\sigma^x)$ . Los susceptibles empiezan a ser infectados con probabilidad  $1-G_k^x$ . Supongamos que una fracción  $(1-\tau^x)$  de los individuos infectados que no se recuperó de forma natural es tratada y vuelve a la clase susceptible. En primer lugar, las ecuaciones que describen la dinámica de la enfermedad dentro de cada parche están dadas por

$$\widetilde{S}_k^x = \gamma^x G_k^x S_k^x + \gamma^x (1-\sigma^x \tau^x) I_k^x$$

$$\widetilde{I}_k^x = \gamma^x (1-G_k^x) S_k^x + \gamma^x \sigma^x \tau^x I_k^x$$

Ahora, se considera la dispersión de individuos entre los parches:  $D_s$  y  $D_I$  son las fracciones de individuos susceptibles e infectados de cada población que se intercambian en cada paso del tiempo, se les asignan superíndices para distinguir las fracciones asociadas a cada parche. Así, el sistema SIS para una metapoblación con dos parches se define como sigue:

$$S_{k+1}^1 = (1-D_S^1) \widetilde{S}_k^1 + D_S^2 \widetilde{S}_k^2$$

$$I_{k+1}^1 = (1-D_I^1) \widetilde{I}_k^1 + D_I^2 \widetilde{I}_k^2$$

$$S_{k+1}^2 = D_S^1 \widetilde{S}_k^1 + (1-D_S^2) \widetilde{S}_k^2$$

$$I_{k+1}^2 = D_I^1 \widetilde{I}_k^1 + (1-D_I^2) \widetilde{I}_k^2$$

$\widetilde{S}_k^x$  y  $\widetilde{I}_k^x$  representa la dinámica local de cada parche.

El control de una enfermedad infecciosa se hace mediante la intervención directa o la reducción de la susceptibilidad de una población. Al analizar el modelo anterior, observamos que se puede ejercer control sobre varios parámetros que representen la aplicación de estrategias para un control óptimo de la enfermedad. Se puede controlar el flujo de personas infectadas entre las dos poblaciones, capturado por los parámetros  $D_I^1$  y  $D_I^2$ . Se puede tomar como variable de control  $\tau^x$ , donde  $1-\tau^x$  es la proporción de la población a la que se le aplicará algún tipo de tratamiento en el parche  $x$ . El impacto de las medidas de intervención directa, como el distanciamiento social, produce cambios en el valor de  $\alpha$ . Más precisamente, se define  $\alpha^x = (1-f^x)\alpha$  como el parámetro de transmisión dentro del grupo  $x$  después de que las estrategias de intervención se han aplicado para reducir la transmisión, que se mide por el parámetro  $f^x$ . Después de introducir estos controles, las ecuaciones para cada parche son:

$$S_{k+1}^1 = (1-D_S^1) \widetilde{S}_k^1 + D_S^2 \widetilde{S}_k^2$$

$$I_{k+1}^1 = (1-D_I^1) \widetilde{I}_k^1 + D_I^2 \widetilde{I}_k^2$$

$$S_{k+1}^2 = D_S^1 \widetilde{S}_k^1 + (1-D_S^2) \widetilde{S}_k^2$$

$$I_{k+1}^2 = D_I^1 \widetilde{I}_k^1 + (1-D_I^2) \widetilde{I}_k^2$$

donde la dinámica local de cada parche estaría dada por el sistema

$$\widetilde{S}_k^x = \gamma^x G_k^x S_k^x + \gamma^x (1-\sigma^x \tau_k^x) I_k^x$$

$$\widetilde{I}_k^x = \gamma^x (1-G_k^x) S_k^x + \gamma^x \sigma^x \tau_k^x I_k^x$$

y

$$G_k = \exp\left(\frac{-(1-f_k^x)\alpha I_k}{N_k}\right)$$

Interesa estudiar los efectos de la aplicación de algunas medidas de control en la propagación de la enfermedad. Particularmente, se consideran tres medidas:

- Aplicar tratamiento: sea un programa de vacunación o un determinado fármaco. Si hay recursos suficientes, todas las personas infectadas pueden ser tratadas, de lo contrario, la finalidad del tratamiento es reducir al mínimo el total de infectados (Rowthorn et al., 2009).
- En ausencia de un tratamiento o vacuna, se puede utilizar el aislamiento de los individuos diagnosticados –al que se le llamará distanciamiento social– para controlar la propagación de la enfermedad. En el distanciamiento social, la persona enferma se separa de las que no están enfermas y puede ser atendida en un centro médico o en su hogar. Algunas medidas potencialmente eficaces, según WHO (2005), son las restricciones de las actividades sociales, por ejemplo, la cancelación de eventos públicos masivos (conciertos, deportivos, cine, obras de teatro, festivales), el cierre temporal de instalaciones de recreación (restaurantes, piscinas, clubes sociales, gimnasios), cierre de oficinas, escuelas y sistemas de transporte público.
- Restricciones sobre la dispersión: la imposición de controles sobre los parámetros que regulan el flujo migratorio entre los parches de una metapoblación, limitando la tasa de dispersión de individuos infectados, lo que sería equivalente a controles de frontera entre los parches (Rowthorn et al., 2009). Aunque tal vez en la realidad no sea posible restringir en su totalidad el flujo de personas infectadas entre los parches, sí es posible aplicar algún tipo de limitación. Para enfrentar el impacto de la dispersión en el desarrollo de la enfermedad se puede establecer mayor vigilancia en los puestos fronterizos con el fin de detectar individuos infectados, adoptar medidas tales como la inspección visual, aplicación de cuestionarios, medición de la temperatura (utilizando arcos térmicos), uso de imágenes térmicas y pruebas de laboratorio rápidas, entre otros métodos de inspección (Ali & Keil, 2007).

En resumen, se ejercerá control sobre los parámetros  $1-\tau_k^x$ ,  $f_k^x$  y  $D_k^x$ , en cada uno de los parches, que representan: la proporción de la población a la que se le aplicará tratamiento médico, la proporción de la población a la que se le aplicará distanciamiento social y proporción de individuos infectados que se dejarán pasar de un parche a otro, respectivamente.

## Resultados numéricos

Para las simulaciones se trabajó con las siguientes cotas para los controles:

$$f^x \in [0,0.2], 1-\tau^x \in [0,0.05] \text{ (tomado de González et al., 2011)}$$

$$D_i^x \in [0.0001, D_s^x]$$

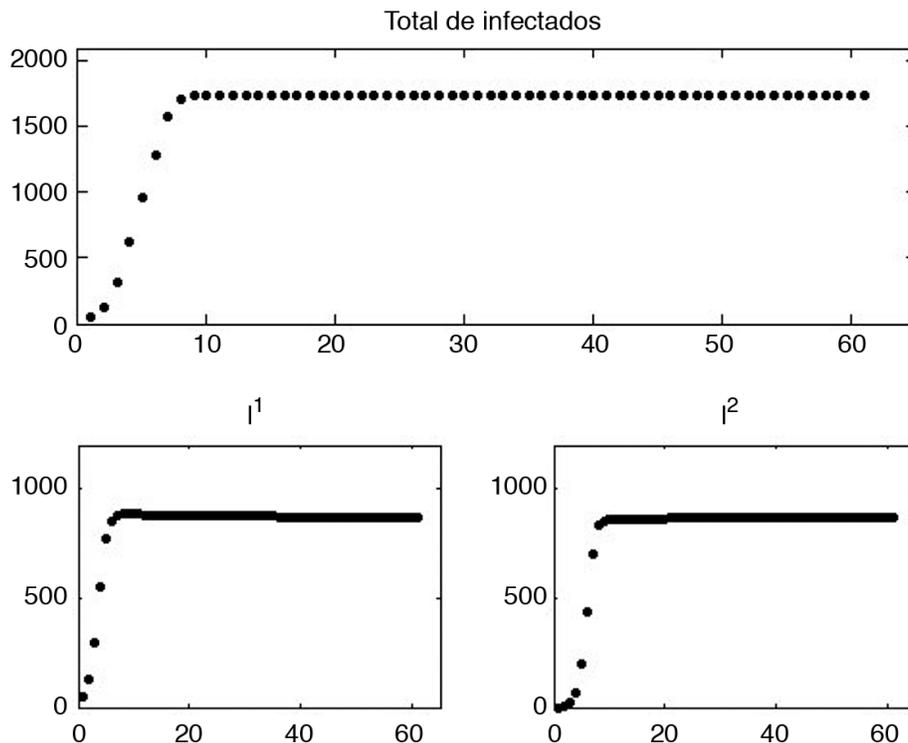
Se asignó arbitrariamente una cota inferior para  $D_i^x$  considerando que lograr una restricción completa en el flujo de personas de un parche a otro es realmente imposible. Además, se consideró  $D_s^x$  como el mayor valor que puede tomar  $D_i^x$ .

El valor para los demás parámetros del modelo se presenta en el cuadro 1.

Consideremos una las siguientes poblaciones iniciales para la clase de susceptibles  $S_1^1 = S_1^2 = 1000$ , además  $I_1^1 = 100$  y  $I_1^2 = 0$  suponiendo que se detecta el brote de la enfermedad en el parche 1 y se quiere ejercer control para evitar que se inicie un brote en el parche 2.

**Cuadro 1.** Valor de los parámetros del modelo SIS metapoblac

Parámetro	Valor	Definición	Fuente
$T$	60	Días transcurridos	
$\gamma$	1	Probabilidad de sobrevivir a la enfermedad	
$\beta$	1.94	Tasa de transmisión	González et al., 2011
$1-\sigma^x$	$\frac{1}{7}$	Fracción de recuperados naturalmente	González et al., 2011
$D_s^x$	0.03	Fracción de susceptibles que emigran del parche $x$ .	Herrera-Valdez et al., 2011



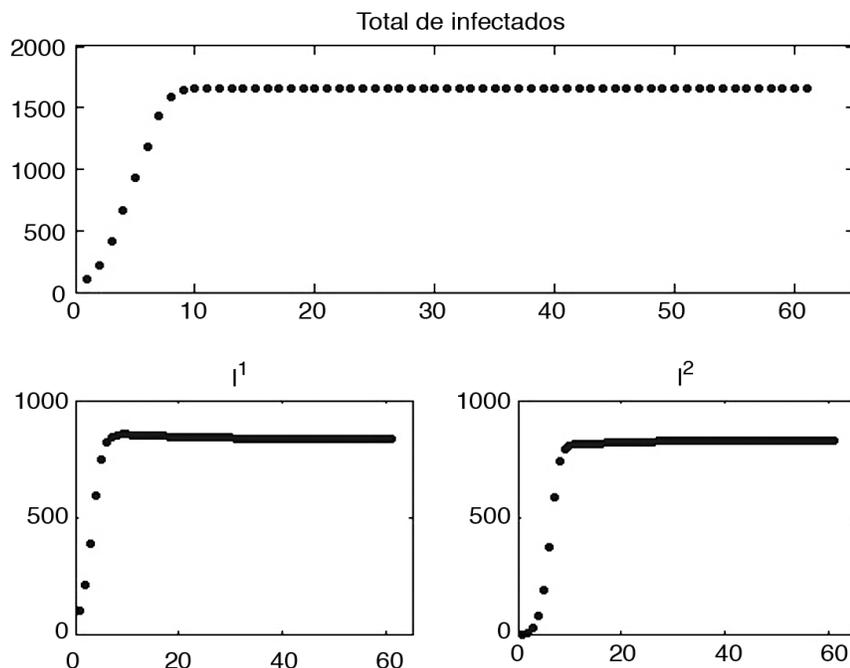
**Figura 1.** Total de infectados si no se aplica ninguna medida de control.

Lo que se quiere saber en la mayoría de los casos es la cantidad de personas infectadas que habrá en un período determinado de tiempo, suponiendo que no se aplica ninguna medida de control. Los resultados se muestran en la figura 1, donde se observa que en menos de 20 días se tienen más de 500 personas infectadas en el parche 2, aunque al finalizar los 60 días el total de infectados es mayor en la población en la que se inició la enfermedad.

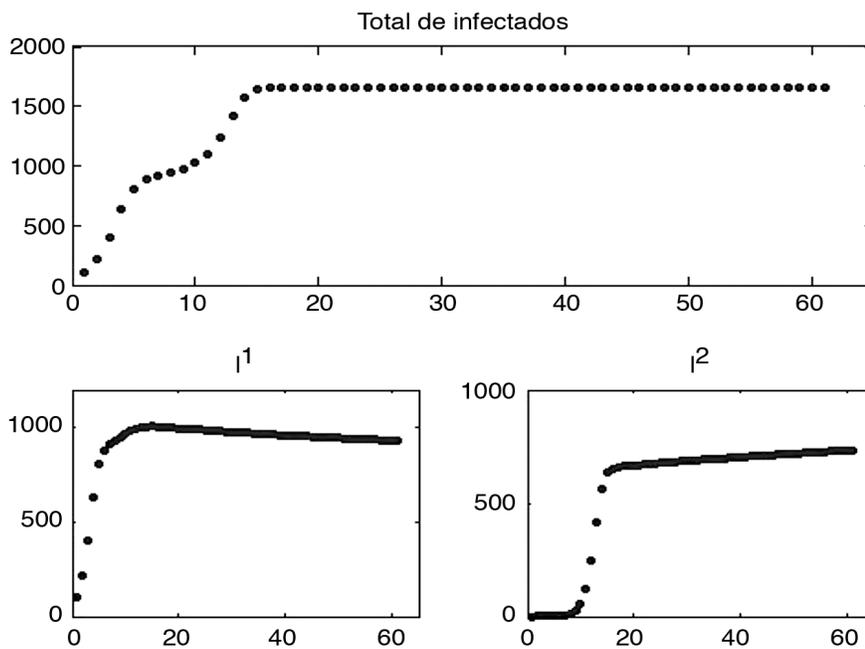
Supongamos que se aplique el máximo control para  $f^x$  y  $1-\tau^x$ , es decir, se aplicará distanciamiento social al 20% de la población y el tratamiento al 5% de la población. Para los valores anteriores consideremos dos casos, primero, que no se aplica ninguna restricción en el flujo de personas (figura 2), y segundo, que se aplica la máxima restricción, sin lograr la restricción total, para el flujo de personas (figura 3). Comparando estos dos casos, se puede observar que, aunque el número de infectados es similar, la aplicación de restricciones retrasa el máximo de enfermos al día 15; mientras que si no se aplica restricción alguna, el máximo se alcanza en el día 10. Además, aplicar alguna medida de control que regule el paso de personas de una población a otra produce una pequeña segunda ola de infectados.

En la figura 4 se ha considerado el caso de aplicar el máximo para las tres medidas de control en la población 1 y en la población 2 solamente la restricción máxima en el flujo de personas, es decir, no se ha aplicado distanciamiento social ni tratamiento en la población 2. La dinámica es muy similar al caso de la figura 3, pero se nota un aumento en el total de infectados de la población 2.

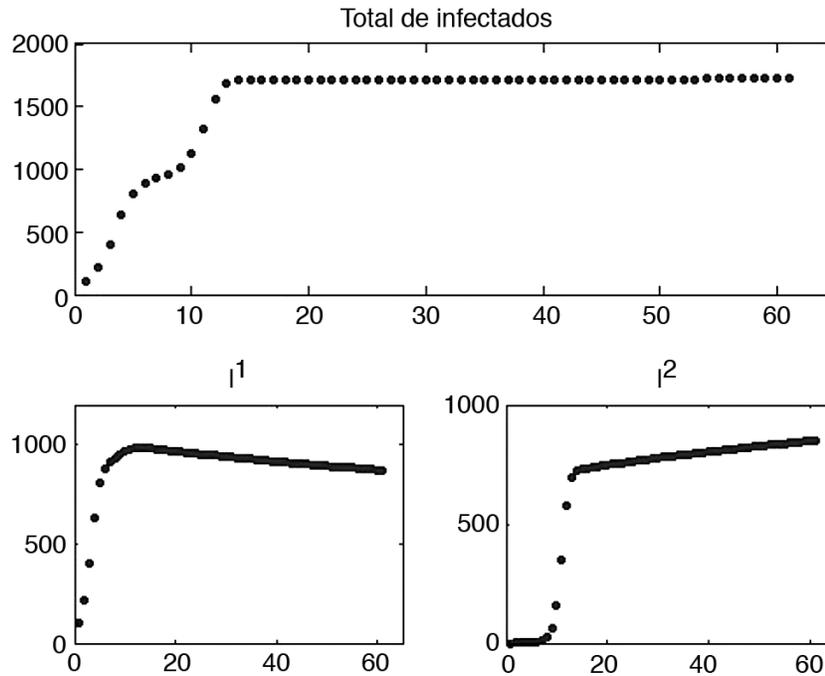
Hay que recordar que, aunque se está aplicando máxima restricción en el flujo de personas, se consideró una cota inferior diferente de cero para el mínimo en el flujo, dado que en la realidad es imposible la restricción total.



**Figura 2.** Total de infectados aplicando máximo control para  $f^x$  y  $1-\tau^x$ , y ninguno para  $D_T^x$



**Figura 3.** Total de infectados aplicando máximo control para  $f^x$  y  $1-\tau^x$ , y ninguno para  $D_T^x$ .



**Figura 4.** Total de infectados aplicando máximo control para  $f^1$  y  $1-\tau^1$ , y ninguno para  $D_i^1$

## Conclusiones

Se ha demostrado una vez más que las simulaciones con modelos matemáticos proporcionan información útil sobre las medidas de control que se deben tomar para lograr controlar una enfermedad a tiempo y evitar consecuencias desastrosas. Esta información pueden usarla los profesionales en salud para la planeación de programas de salud pública. Además, el modelo se puede utilizar para la evaluación de programas de control, trabajando otras simulaciones, constituyendo una opción económica y efectiva.

En los resultados obtenidos se hace evidente que el propósito principal de las restricciones en el flujo de personas infectadas es retrasar la propagación de la enfermedad hasta que se puedan desarrollar y aplicar otras intervenciones, o bien se pueden combinar con otras medidas de control (Herrera-Valdez et al., 2011). Ese retraso puede ser de gran ayuda en el desarrollo de la conciencia pública, la aplicación del distanciamiento social y la preparación de los centros de atención y de las medidas de prevención. Todo tiempo adicional que permita prepararse es una oportunidad para reducir los efectos potenciales de varias enfermedades.

## Agradecimiento

Agradezco a la Vicerrectora de Investigación del Instituto Tecnológico de Costa Rica por el apoyo brindado al proyecto de investigación: Propagación de enfermedades: Heurísticas aplicadas a la optimización de medidas de control, código VIE 540114404201.

## Bibliografía

- Ali, S. & Keil, R. (2007). Contagious cities. *Geography Compass*, 1, 1207-1226.
- Alpizar, G. (2012). *Estrategias de control óptimas en la propagación de enfermedades en poblaciones acopladas*. Tesis de Maestría, Universidad de Puerto Rico, Recinto Mayaguez.
- Alpizar, G. & Gordillo, L. (2013). Disease Spread in Coupled Populations: Minimizing Response Strategies Costs in Discrete Time Models. En *Discrete Dynamics in Nature and Society*.
- Arino, J. & Driessche, P. (2006). Disease spread in metapopulations. *Fields Institute Communications*, 48, 1-13.
- Brauer, F., Castillo-Chavez, C., De la Pava, E., Castillo-Garsow, C., Chowell, D., Espinoza, B., González, P., Hernández, C. & Moreno, V. (2014). *Modelos de la propagación de enfermedades infecciosas*. Universidad Autónoma de Occidente. Cali, Colombia.
- Castillo-Chavez, C. & Yakubu, A. (2002). Intraspecific competition, dispersal and disease dynamics in discrete-time patchy environments. En *Mathematical Approaches for Emerging and Reemerging Infectious Diseases: An Introduction to Models, Methods and Theory*, 125, 165-181. Castillo-Chávez C., van den Driessche P., Kirschner D. and Yakubu A.-A. (Eds.), Springer, New York, 2002.
- Ding, W. & Lenhart, S. (2010). Introduction to Optimal Control for Discrete Time Models with an Application to Disease Modeling. *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, 75, 109-119.
- González, P., Lee, S., Velázquez, L. & Castillo-Chavez, C. (2011). A note on the use of optimal control on a discrete time model of influenza dynamics. *Mathematical Biosciences and Engineering*, 8, 183-197.
- Herrera-Valdez, M., Cruz-Aponte, M. & Castillo-Chavez, C. (2011). Multiple outbreaks for the same pandemic: local transportation and social distancing explain the different waves of AH1N1PDM cases observed in Mexico during 2009. *Mathematical Biosciences and Engineering*, 8, 21-48.
- Kretzschmar, M. & Wallinga, J. (2010). Mathematical models in infectious disease epidemiology. En *Modern Infectious Disease Epidemiology* (214-226). Springer. Alexander Krämer, Mirjam Kretzschmar, Klaus Krickeberg (Eds.), New York, 2010.
- Lee, S., Chowell, G. & Castillo-Chavez, C. (2010). Optimal control of influenza pandemics: The role of antiviral treatment and isolation. *Journal Theoretical Biology*, 265, 136-150.
- Rowthorn, R., Laxminarayan, R. & Gilligan, C. (2009). Optimal control of epidemics in metapopulations. *The Royal Society*, 6(41), 1135-1144.
- Welte, R., Leidl, R., Greiner, W. & Postma, M. (2010). Health economics of infectious diseases. En *Modern Infectious Disease Epidemiology* (pp. 253-279). Springer. Alexander Krämer, Mirjam Kretzschmar, Klaus Krickeberg (Eds.), New York, 2010.
- WHO (World Health Organization). (2005). *Avian influenza: assessing the pandemic threat*. Obtenido de <http://www.who.int/csr/disease/influenza/H5N1-9reduit.pdf>.
- WHO (World Health Organization). (2009). *New Influenza A(H1N1): Number of laboratory confirmed cases and deaths as reported to WHO*. Obtenido de [http://www.who.int/csr/don/GlobalSubnationalMaster\\_20090507\\_1800a.png?ua=1](http://www.who.int/csr/don/GlobalSubnationalMaster_20090507_1800a.png?ua=1)

# Identificación de abejas sin aguijón (Apidae: Meliponini) a partir de la clasificación de los descriptores SIFT de una imagen del ala derecha anterior

## Identification of stingless bees beginning from the classification of SIFT descriptors of an image of the right forewing

Geovanni Figueroa-Mata<sup>1</sup>, Juan Pablo Prendas-Rojas<sup>2</sup>, Melvin Ramírez-Bogantes<sup>3</sup>, Ingrid Aguilar-Monge<sup>4</sup>, Eduardo Herrera-González<sup>5</sup>, Carlos Manuel Travieso-González<sup>6</sup>

---

*Fecha de recepción: 19 de junio del 2015*

*Fecha de aprobación: 26 de setiembre del 2015*

Figueroa-Mata G; Prendas-Rojas J; Ramírez-Bogantes M; Aguilar-Monge I; Herrera-González E; Travieso-González, C. Identificación de abejas sin aguijón (Apidae: Meliponini) a partir de la clasificación de los descriptores SIFT de una imagen del ala derecha anterior. *Tecnología en Marcha*. Edición especial. Matemática Aplicada, Mayo 2016. Pág 51-63.

---

1 Máster en Computación. Escuela de Matemática. Instituto Tecnológico de Costa Rica. Tel. (506) 25502012. Correo electrónico: gfigueroa@tec.ac.cr

2 Licenciado en Matemática. Escuela de Matemática. Instituto Tecnológico de Costa Rica. Tel. (506) 25502225. Correo electrónico: pprendas@tec.ac.cr

3 Licenciado en Matemática. Escuela de Matemática. Instituto Tecnológico de Costa Rica. Tel. (506) 25502. Correo electrónico: meramirez@tec.ac.cr

4 Doctora en Biología Tropical. Centro de Investigaciones Apícolas Tropicales de Costa Rica. Tel. (506) 2237-7043. Correo electrónico: ingrid.aguilar.monge@una.cr

5 Máster en Meliponicultura. Centro de Investigaciones Apícolas Tropicales de Costa Rica. Tel. (506) 2237-7043. Correo electrónico: apieduardo@gmail.com

6 Doctor en Cibernética y Telecomunicaciones. Universidad de las Palmas Gran Canaria. Tel. 928 452864. Correo electrónico: carlos.travieso@ulpgc.es

## Palabras clave

Abejas sin aguijón; clasificación de especies; procesamiento de imágenes; descriptores SIFT; agrupamiento aglomerativo jerárquico.

## Resumen

Se ha comprobado que la información morfométrica de un ala se puede usar para identificar la especie de algunos insectos, incluidas las abejas. En el presente trabajo se utiliza la información contenida en la fotografía de un ala para discriminar la especie de abejas nativas costarricenses sin aguijón. El método propuesto es un algoritmo discriminador basado en la clasificación de los descriptores SIFT de los puntos clave de la imagen del ala anterior derecha. Los especímenes usados en la investigación se recolectaron directamente de los nidos en diferentes zonas geográficas de Costa Rica y corresponden a 18 especies de 10 géneros. El algoritmo fue validado con 436 imágenes de alas con un resultado de 81% de éxito en la discriminación por género y de 76% a nivel de especie.

## Keywords

Stingless bees; classification; image processing; SIFT descriptors; clustering.

## Abstract

It has been found that morphological information contained in a wing can be used to identify the species of certain insects, including bees. This paper uses the information contained in the photograph of a wing to discriminate the specie of native stingless bees. A discriminator algorithm based on SIFT is proposed. Specimens used in the research were collected directly from the nests in different geographic areas of Costa Rica and correspond to 18 species of 10 genders. The proposed algorithm was validated with 436 images with 81% success in gender discrimination and 76% for the species.

## Introducción

Las abejas son insectos con alas de la familia Apoidea, orden Hymenoptera. Existen más de 20.000 especies documentadas alrededor del mundo (Michener, 2007). Se consideran el grupo de polinizadores más importantes, tanto de plantas silvestres como de cultivos (Quesada Euan, 2009). Existen numerosos estudios que demuestran su importancia económica en la industria agrícola, pero su aporte al ecosistema natural es difícil de cuantificar. Según proyecciones de la Organización de las Naciones Unidas (ONU), para el año 2050 la población mundial será de más de 9.000 millones de personas, lo que implicará una gran demanda de alimentos, y de ahí la importancia de las abejas para la supervivencia humana y la conservación de la biodiversidad. No obstante, la sobreexplotación de los bosques y el uso de pesticidas han causado la disminución en el número de colonias de *Apis mellifera* (Apidae: Apini) y efectos adversos no cuantificados en otras especies, entre ellas las abejas nativas sin aguijón (Apidae: Meliponini) (Meléndez, Meneses & Kevan, 2013; Thompson, 2012; Whitehorn, O'Connor, Walkers & Goulson, 2012).

Las abejas nativas sin aguijón tienen un aguijón atrofiado que no pueden utilizar para la defensa. Se conocen más de 400 especies, de las cuales 300 se encuentran en los trópicos y subtrópicos de América. En Costa Rica hay al menos 20 géneros y 59 especies (Camargo &

Pedro, 2007) y la mayoría habitan por debajo de los 1500 msnm. Las especies más utilizadas en Costa Rica para la meliponicultura son *Tetragonisca angustula* y *Melipona beecheii*.

El cuerpo de una abeja se compone de tres partes: cabeza, tórax y abdomen. El tórax está formado por el protórax, el mesotórax y el metatórax. Cada una de estas partes tiene un par de patas. Las alas de las abejas son membranosas y crecen en el tórax, un par en el mesotórax y otro par en el metatórax. Las membranas superior e inferior de las alas cubren una red de tubos endurecidos, llamados nervios o venas, que le dan rigidez a éstas.

## Trabajos relacionados

Tradicionalmente, para la identificación de especies, el experto –melitólogo– utiliza una clave dicotómica (clave taxonómica) que consta de un número de pasos organizados, cada uno de los cuales tiene dos opciones mutuamente excluyentes. Conforme se avanza en la identificación, estas opciones se van descartando. Las claves dicotómicas<sup>7</sup> utilizan características fáciles de observar en el espécimen y con pares de afirmaciones contrapuestas como, por ejemplo, “ojos labros” contra “ojos con pelos”.

Se ha demostrado que es posible identificar especies de insectos usando únicamente la morfometría del ala, ya que su variabilidad entre especies es mayor que dentro de la misma especie. Incluso las diferencias permiten discriminar entre subespecies (Hall, 2011; Tofilski, 2008a; Tofilski, 2008b; Tüzün & Yüksel, 2011). Las alas de los insectos son membranosas y poseen venas o nervaduras visibles, lo que permite la creación de un diagrama claro de su estructura. El uso de indicadores morfométricos ha cobrado relevancia recientemente (Breuker, Gibbs, Van Dongen, Merckx & Van Dyck, 2010; Hall, 2011; Hall, MacLeod & Wardhana, 2014), lo que ha motivado el desarrollo de sistemas de clasificación que utilizan las características morfométricas del ala (Abou Shara & Al Ghamdi, 2012; Arbuckle, Schröder, Steinhage & Wittmann, 2001; Breuker, Gibbs, Van Dongen, Merckx & Van Dyck, 2010). En general, estos sistemas identifican las especies mediante métodos morfométricos basados en el análisis multivariado de distancias y ángulos. En el cuadro 1 se muestra un resumen de algunos de estos sistemas y sus principales características.

Como se puede observar, existen varios sistemas capaces de clasificar especies de abejas de forma eficiente. Sin embargo, todos tienen limitaciones, entre ellas están que ninguno se ha desarrollado específicamente para la discriminación de abejas sin aguijón, por lo general discriminan pocas especies y géneros, en su mayoría no son completamente automáticos, requieren intervención activa de un usuario, requieren de una gran cantidad de información y algunos demandan muchas horas de entrenamiento.

Debido a esto, se considera que el desarrollo de un sistema automático de discriminación de abejas sin aguijón a partir de una fotografía del ala es relevante. En el presente trabajo se propone un sistema de fácil uso, ya que únicamente requiere una fotografía de un ala derecha anterior y con ella se logra identificar la especie más probable a la que pertenece el ala.

7 En <http://www.webbee.org.br/beetaxon/> pueden apreciarse ejemplos del uso de claves dicotómicas para la clasificación de especies de abejas brasileñas.

**Cuadro 1.** Sistemas para la clasificación automática de los insectos.

Sistema	Observaciones	Características
Fast Africanized Bee Identification System (FABIS)	Uno de los primeros sistemas, se desarrolló en la década de los 80 para clasificar abejas africanizadas.	Requiere alrededor de 25 medidas morfométricas y un proceso de entrenamiento.
Automated Bee Identification System (ABIS)	Propuesto en 1991, sin duda es el mejor sistema de identificación de abejas. Diseñado para las alas anteriores.	Reporta un 99,8% de exactitud en la identificación por género y un 95% de exactitud en la identificación por especies. Utilizaba una gran cantidad de información del ala con la que realizaba luego un análisis estadístico para la posterior identificación. Actualmente, el proyecto está inactivo (Arbuckle, Schröder, Steinhage & Wittmann, 2001).
MelittO Biotaxis System (MOBS)	Propuesto en 2011. A diferencia del ABIS, redujo considerablemente las características requeridas para la identificación, utilizando alrededor de 20 puntos clave.	Reporta una exactitud de 90% a 100% en la identificación de géneros. Utiliza las características anatómicas más relevantes y la venación del ala, junto con técnicas como ACP (Hall, 2011).
Draw Wing	Su uso fue impulsado en 2007. Se utiliza como un complemento de los sistemas anteriores. Diseñado para alas anteriores de insectos.	Extrae información de la imagen de un ala que se puede utilizar para la identificación de especies. Reporta 95% de exactitud. Requiere escanear la muestra. ( <a href="http://www.drawwing.org/">http://www.drawwing.org/</a> ).
tpsBias, tpsDig, tpsPLS, tpsPower, tpsRelw, tpsRegr	Conjunto de herramientas para el análisis morfométrico y estadístico de alas de insectos.	Desarrolladas por la Universidad de Stony Brook. ( <a href="http://life.bio.sunysb.edu/morph/">http://life.bio.sunysb.edu/morph/</a> )

## Base de datos y colección de especímenes

Las colecciones de especímenes de abejas son fundamentales para realizar estudios taxonómicos y constituyen la base de muchos tipos de estudio sobre biodiversidad y población. La falta de colecciones entomológicas apropiadas que cubran el inventario de todo el país es una de las debilidades que se presenta al realizar este tipo de estudios.

En esta investigación, la creación de la base de datos incluyó la preparación de tres colecciones: una de especímenes de abejas montadas en alfileres entomológicos, otra de alas derechas anteriores y posteriores montadas en un medio de preservación inerte y una de fotografías de estas placas.

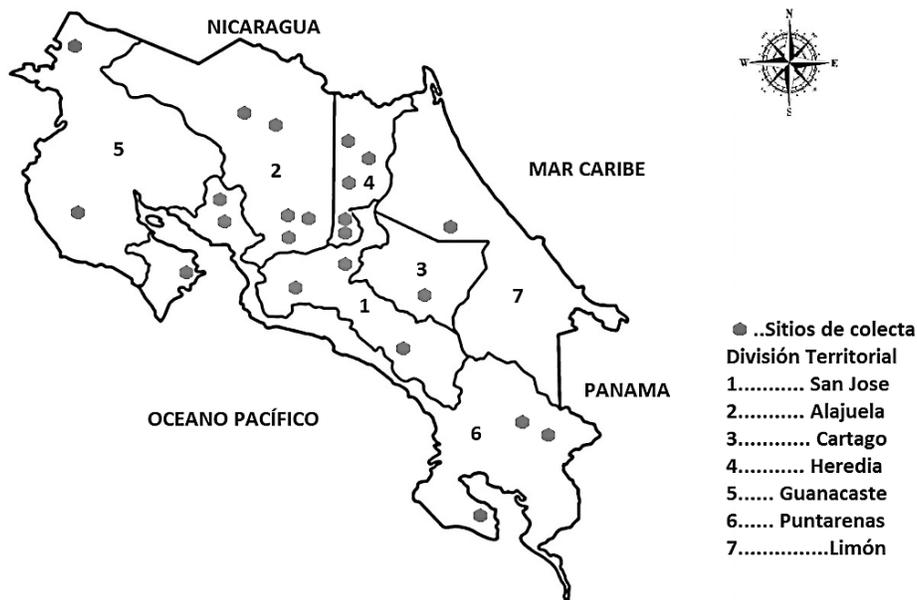
### Colección de especímenes

Parte de los especímenes utilizados se tomó de la colección de abejas nativas sin aguijón del Centro de Investigaciones Apícolas Tropicales (CINAT). Se revisaron todos los especímenes para seleccionar aquellos cuya ala derecha, anterior y posterior estuviera en perfectas condiciones. Para las especies de las que no se obtuvieron individuos con sus alas en perfectas condiciones, fue necesario realizar visitas de campo para ubicar nidos, de donde se recolectaron los especímenes necesarios para la investigación.

Las abejas se recolectaron directamente de los nidos en diferentes zonas geográficas de Costa Rica (figura 1), utilizando red entomológica y succionadores de plástico. Se usó una “cámara de muerte” con acetato de etilo para matar rápidamente al insecto y evitar daños en las alas. Las muestras se trasladaron al CINAT, donde se realizó el proceso de preservación de los ejemplares, el montaje y la identificación respectiva. Para cada ejemplar se registró la información necesaria para su identificación taxonómica: país, provincia, cantón, fecha, nombre del recolector, muestra en nido o en flor, etc.

### Colección de alas

Se extrajo el ala derecha anterior y posterior de las abejas, se cortó en la base del mesotórax con navajilla y tijeras de disección para facilitar el desprendimiento y no dañarla. Se quitó el residuo de la base del ala para su montaje en un portaobjetos con una gota de bálsamo de Canadá. Se usó una lámpara de 50 W para suavizar el bálsamo y permitir que la muestra quedara inmersa en el sustrato sin ser dañada. Se colocó el cubreobjetos con presión moderada para extraer las burbujas y evitar el desplazamiento del ala. Las alas en estudio se dejaron reposar sobre papel toalla por 24 horas antes de ser fotografiadas y almacenadas. La colección de alas comprende un total de 879 placas.



**Figura 1.** Zonas geográficas de los sitios de recolección.

### Colección de fotografías

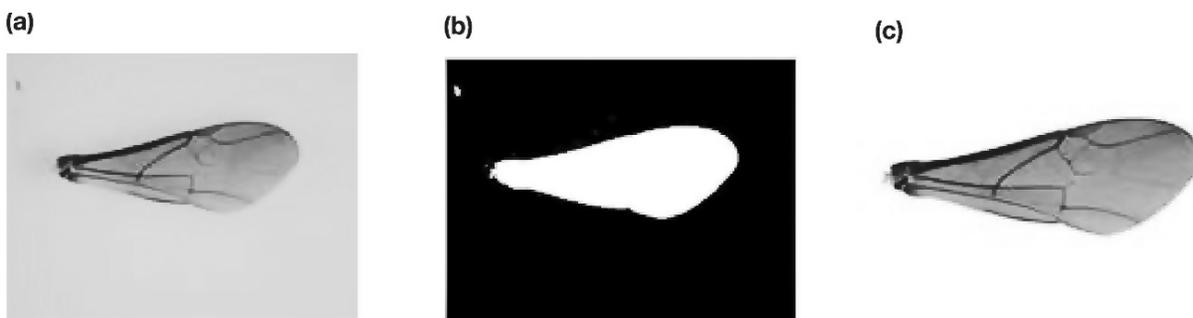
Para la toma adecuada de las fotografías se estableció un protocolo (Herrera et al., 2013). Las imágenes fueron tomadas con un estereoscopio Olympus modelo SZX16 con una cámara integrada marca Olympus modelo DP72, con una fuente de luz circular intralux 4000-1 con una intensidad de entre 1400 y 1600fc, dependiendo de la muestra. La resolución en el milimétrico fue de 2x con una apertura focal de 1/4. Para la captura de imágenes se utilizó un sistema TWAIN con exposición manual, tiempo de exposición 60%, sensibilidad ISO 200 y una resolución Live 1360 x 1024 y un *snap* 4140 x 3096.

Las fotografías se almacenaron en una base de datos, agrupadas por especie. La colección comprende un total de 1758 fotografías.

### Procesamiento de las imágenes

Para la sustracción del fondo de cada una de las fotografías se utilizaron histogramas, discriminación por umbrales y operadores morfológicos. Las fotografías presentaban dos tipos de fondos: uno con un tono gris y otro con presencia de azul. Para determinar el tipo de fondo de cada fotografía se utilizó el espacio de color HSV y una cota, calculada empíricamente. Una vez identificado el tipo de fondo, se hizo un filtrado mediante umbralización (Otsu) ajustado experimentalmente, para determinar cuáles píxeles se convertirían en blancos y cuáles en negros. En la figura 2 se muestra este proceso; la región en blanco de la fotografía (b) corresponde a lo que se identificó como ala y la fotografía (c) es el resultado del proceso de eliminación del fondo.

Algunas fotografías presentaban dos alas, siendo la de interés la ubicada más al centro de la imagen, según el protocolo establecido en Herrera et al. (2013). Para resolver esto, se calcularon los centros de masa de todas las regiones blancas y se eligió como el ala de interés aquella cuya distancia del centro de masa al centro de la imagen fuese menor.



**Figura 2.** (a) fotografía original de la colección, (b) fotografía obtenida luego del filtrado por umbral, conversión a blanco y negro y operaciones morfológicas, (c) resultado del proceso de eliminación de fondo.

### Aprendizaje

En la figura 3 se muestra el esquema del proceso de entrenamiento propuesto:

1. Para cada imagen de entrenamiento se extraen sus descriptores SIFT.
2. Se realiza un agrupamiento aglomerativo jerárquico de los descriptores extraídos.
3. Para cada descriptor en cada grupo se almacena su correspondiente especie.

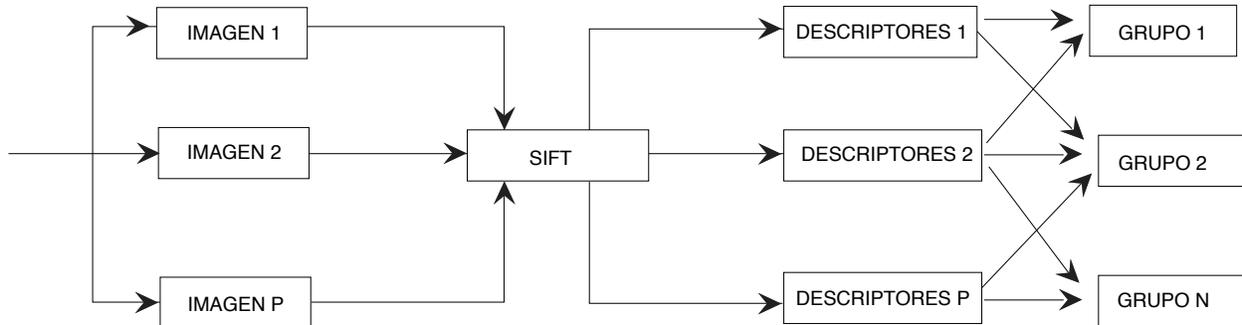
Se espera que cada grupo sea representativo de una única especie, pero la práctica demuestra que esto no siempre sucede, ya que existen grupos que comparten descriptores de varias especies. Por lo tanto, fue necesario resolver esta ambigüedad en la etapa de clasificación.

### Descriptores SIFT de puntos clave

El Scale Invariant Feature Transform (SIFT) es una transformación propuesta por David Lowe en 1999, cuya idea básica es representar una imagen mediante puntos clave (Lowe, 1999).

El método SIFT reduce la imagen a un conjunto de parámetros invariantes a la escala y rotación. Esto lo logra mediante la creación de pirámides de *scale-space* y la convolución con funciones

gaussianas. La imagen obtenida por diferencia de gaussianas logra resaltar los bordes y esquinas de la imagen.



**Figura 3.** Esquema del proceso de entrenamiento.

Una vez calculadas las diferencias gaussianas, se identifican los candidatos a máximos y mínimos entre los puntos que no se anularon. La comparación de cada punto con sus vecinos se hace en la escala próxima. Luego estos se discriminan por estabilidad.

Para asignar la orientación a los puntos clave  $(x, y)$  calculados en la etapa anterior, se utilizan la magnitud del gradiente y la orientación. Con la información obtenida se asigna a cada punto clave un conjunto de características, entre las que se destacan la posición, la orientación y las rotaciones del gradiente (para lograr invariancia respecto a rotaciones en 2D o cambios de puntos de perspectiva en 3D). Así, a cada punto clave se le asocia una matriz de  $4 \times 4 \times 8$ , es decir, un vector de tamaño 128, llamado descriptor del punto clave.

### Agrupamiento de los descriptores

Para construir los grupos de descriptores se usó el agrupamiento aglomerativo jerárquico. Según este método, dados  $F$  descriptores extraídos de las imágenes de entrenamiento, el agrupamiento se inicia con  $F$  grupos, cada uno conteniendo un único descriptor. En cada iteración, se unen los dos grupos con mayor proximidad.

Para determinar la proximidad entre grupos se utilizó la agregación de salto promedio con la métrica euclídea (Trejos, Castillo & González, 2014). Esta se define como la distancia promedio de todo descriptor de un grupo a todo descriptor del otro grupo:

$$d(G^p, G^q) = \frac{1}{M N} \sum_{i=1}^M \sum_{j=1}^N \|d_i^p - d_j^q\|$$

Aquí,  $M$  y  $N$  son el número de descriptores de cada uno de los grupos  $G^p$  y  $G^q$  respectivamente,  $d_i^p$  es el  $i$ -ésimo descriptor del grupo  $G^p$  y  $d_j^q$  es el  $j$ -ésimo descriptor del grupo  $G^q$ .

El agrupamiento aglomerativo en cada iteración va uniendo grupos hasta que un criterio de parada detiene el proceso, después de lo cual se obtienen grupos donde la proximidad entre cada par de grupos está por encima de un valor dado. Este valor determina el criterio de parada, que en nuestro caso fue determinado experimentalmente.

Las principales desventajas del agrupamiento aglomerativo son sus excesivos requerimientos en tiempo de ejecución y de memoria, que son del orden de  $\Theta(F^2)$ .

### Clasificación de una nueva imagen

Dada una nueva imagen, la clasificación se lleva a cabo de la siguiente forma:

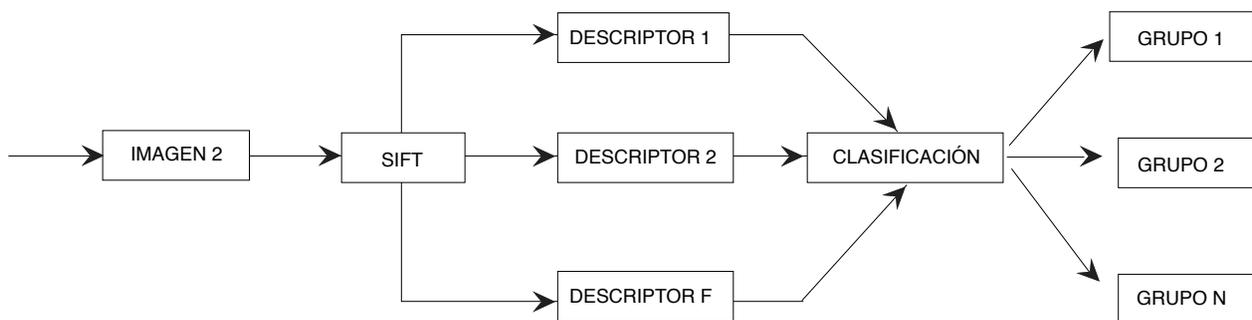
1. Primero se extraen los descriptores SIFT de la imagen.
2. Luego, cada uno de estos descriptores se clasifica en uno de los grupos del modelo aprendido.
3. Por último, se determina la especie más probable a la que pertenece la imagen de entrada.

En la figura 4 se muestra el esquema del proceso de discriminación propuesto. Primero se extraen los descriptores  $d_1, d_2, \dots, d_F$  de la imagen que se desea identificar. Luego, mediante análisis discriminante lineal (Johnson, 2000), se clasifica cada uno de estos descriptores  $d_i$  en uno de los grupos  $G_j$  obtenidos en el proceso de entrenamiento. Por último, se seleccionan las  $k$  especies más probables para la imagen de entrada. Para esto se calculan las sumas de las probabilidades de ocurrencia dado cada uno de los grupos seleccionados en la etapa anterior:

$$P(E_i) = \sum_j P(E_i|G_j)P(G_j)$$

Luego se eligen los  $k$  valores mayores para cada imagen de entrada. Aquí  $P(E_i|G_j)$  es la probabilidad de que la especie sea  $E_i$  dado el grupo  $G_j$  y  $P(G_j)$  es la probabilidad del grupo  $G_j$ .

El algoritmo fue programado en MatLab y se utilizó la biblioteca VLFeat<sup>8</sup> para la obtención de los descriptores SIFT de los puntos clave de una imagen.



**Figura 4.** Esquema del proceso de discriminación.

### Resultados

La obtención y análisis de los resultados se hizo de la siguiente forma:

1. Se usó una muestra pequeña de imágenes para calibrar los procesos de eliminación de fondo y la obtención de los descriptores SIFT.

<sup>8</sup> VLFeat es una biblioteca de código abierto que implementa algoritmos de visión por computadora. Se puede descargar en <http://www.vlfeat.org/>

2. Se entrenó el algoritmo con una muestra de 8 imágenes por especie y se utilizaron 18 especies para un total de 144 imágenes, como se muestra en el cuadro 2.
3. Se validó el algoritmo con una muestra de 436 imágenes totalmente diferentes a las utilizadas en las etapas anteriores.

**Cuadro 2.** Totalidad de especies utilizadas en la etapa de entrenamiento.

Cant.	Especie	Cant.	Especie
8	<i>Cephalotrigona zexmeniae</i>	8	<i>Scaptotrigona pectoralis</i>
8	<i>Geotrigona lutzi</i>	8	<i>Scaptotrigona subobscuripennis</i>
8	<i>Lestrimelitta danuncia</i>	8	<i>Tetragona zieglerei</i>
8	<i>Melipona costaricensis</i>	8	<i>Trigona corvina</i>
8	<i>Nannotrigona melleria</i>	8	<i>Trigona fulviventris</i>
8	<i>Nannotrigona perilampoides</i>	8	<i>Trigona fuscipennis</i>
8	<i>Partamona orizabaensis</i>	8	<i>Trigona muzoensis</i>
8	<i>Plebeia frontalis</i>	8	<i>Trigona nigerrima</i>
8	<i>Plebeia jatiformis</i>	8	<i>Trigona silvestriana</i>

Para la etapa de validación se utilizó una muestra de 436 fotografías que corresponden a 18 especies de 10 géneros. Debido a que no se pudo recolectar una cantidad suficiente de especímenes para 6 de las especies, el algoritmo se entrenó con 24 especies y se validó con 18.

Los resultados del proceso de validación con  $k=1$  se tabularon en una matriz de discriminación cruzada, como se muestra en el cuadro 3. Por ejemplo, de las 30 imágenes de la especie *Partamona orizabaensis*, 27 fueron identificadas correctamente, 2 fueron identificadas como de la especie *Trigona fulviventris* y 1 como de la especie *Geotrigona lutzi*. Se usaron colores para agrupar las especies de un género, por ejemplo, para las de *Trigona* se utilizó el color naranja.

En el cuadro 4 se resumen los resultados de la etapa de validación usando un valor de  $k=1$  (considerando únicamente la especie más probable). El porcentaje promedio de éxito para la discriminación por especie fue de 76,3%, mientras que para la discriminación por género alcanzó 80,7%.

En el cuadro 4 observamos que la discriminación por género para el caso de *Trigona* es del 100%, no así la discriminación por especie, que es de un 70,7%, pero es importante notar que ninguna de las imágenes de esta especie está clasificada como de otra especie (cuadro 3). Esto sugiere que las alas de las especies de este género son muy similares entre sí, pero tienen cierto grado de diferencia con las alas de las especies de otros géneros.

Esto lleva a pensar que, con el propósito de poder discriminar entre especies de ciertos géneros, como en el caso de *Trigona*, tal vez sea necesario considerar otros discriminadores que no se encuentran en las alas como, por ejemplo, color del abdomen, vellosoidad en las patas o antenas, etc.



**Cuadro 4.** Porcentaje de éxito considerando las especies más probables,  $k=1$ .

#	Género	Especie	Código	Muestras	Errores	Éxito por especie	Éxito por género
1	<i>Cephalotrigona</i>	<i>zexmeniae</i>	Ce.ze	10	0	100	100
2	<i>Geotrigona</i>	<i>lutzi</i>	Ge.lu	12	0	100	100
3	<i>Lestrimelitta</i>	<i>danuncia</i>	L.da	32	2	93.75	93.75
4	<i>Melipona</i>	<i>costaricensis</i>	Me.co	15	10	33.33	33.33
5	<i>Nannotrigona</i>	<i>melleria</i>	Na.me	33	0	100	97.37
6		<i>perilampoides</i>	Na.pe	5	1	80	
7	<i>Partamona</i>	<i>orizabaensis</i>	Par.or	30	3	90	90
8	<i>Plebeia</i>	<i>frontalis</i>	Pl.fr	15	0	100	100
9		<i>jatiformis</i>	Pl.ja	25	0	100	
10	<i>Scaptotrigona</i>	<i>pectoralis</i>	Scp.pe	35	1	97.14	75.6
11		<i>subobscuripennis</i>	Scp.su	31	19	38.7	
12	<i>Tetragona</i>	<i>ziegleri</i>	Te.zi	24	20	16.7	16.7
13	<i>Trigona</i>	<i>corvina</i>	Tri.co	29	19	34.48	100
14		<i>fulviventris</i>	Tri.ful	34	1	97.06	
15		<i>fuscipennis</i>	Tri.fus	31	22	29.03	
16		<i>muzoensis</i>	Tri.mu	32	6	81.25	
17		<i>nigerrima</i>	Tri.ni	14	2	85.71	
18		<i>silvestriana</i>	Tri.si	29	1	96.55	
Totales				436	107	76.32	80.69

**Cuadro 5.** Porcentajes de éxito considerando las dos especies más probables,  $k=2$ .

#	Género	Especie	Código	Muestras	Errores	Éxito por especie	Éxito por género
1	<i>Cephalotrigona</i>	<i>zexmeniae</i>	Ce.ze	10	0	100	100
2	<i>Geotrigona</i>	<i>lutzi</i>	Ge.lu	12	0	100	100
3	<i>Lestrimelitta</i>	<i>danuncia</i>	L.da	32	1	96.88	96.88
4	<i>Melipona</i>	<i>costaricensis</i>	Me.co	15	9	40	40
5	<i>Nannotrigona</i>	<i>melleria</i>	Na.me	33	0	100	97.37
6		<i>perilampoides</i>	Na.pe	5	1	80	
7	<i>Partamona</i>	<i>orizabaensis</i>	Par.or	30	1	96.67	96.67
8	<i>Plebeia</i>	<i>frontalis</i>	Pl.fr	15	0	100	100
9		<i>jatiformis</i>	Pl.ja	25	0	100	
10	<i>Scaptotrigona</i>	<i>pectoralis</i>	Scp.pe	35	1	97.14	75.6
11		<i>subobscuripennis</i>	Scp.su	31	15	51.61	
12	<i>Tetragona</i>	<i>ziegleri</i>	Te.zi	24	15	37.5	37.5
13	<i>Trigona</i>	<i>corvina</i>	Tri.co	29	5	82.76	100
14		<i>fulviventris</i>	Tri.ful	34	0	100	
15		<i>fuscipennis</i>	Tri.fus	31	19	38.71	
16		<i>muzoensis</i>	Tri.mu	32	1	96.88	
17		<i>nigerrima</i>	Tri.ni	14	1	92.86	
18		<i>silvestriana</i>	Tri.si	29	0	100	
Totales				436	69	83.95	84.41

## Conclusiones

Hemos podido comprobar que es posible identificar el género y la especie de abejas nativas sin aguijón a partir de la fotografía del ala derecha anterior, con porcentajes de éxito que rondan el 84%.

Con base en la información dada por expertos en estudios apícolas, existen especies cuyas alas son muy similares entre sí, por lo que su identificación se realiza considerando otros aspectos como: color de ciertas partes del cuerpo, vello, tamaño de las patas o antenas, etc. Por ejemplo, las alas de las especies *Trigona silvestriana* y *Trigona fulviventris* son muy similares pero se diferencian por la coloración del abdomen. Por eso, al incorporar algunos de estos discriminadores, los porcentajes de éxito podrían mejorar.

Aún quedan algunas tareas pendientes, como investigar las razones de que los porcentajes de discriminación para las especies *Melipona costaricensis* y *Tetragona zieglerei* sean los más bajos; evaluar otras técnicas para el proceso de clasificación de descriptores e incorporar más especies en los procesos de entrenamiento y validación.

## Agradecimiento

Agradecemos a la Vicerrectora de Investigación del Instituto Tecnológico de Costa Rica por el apoyo brindado al proyecto de investigación: Sistema automático de clasificación de abejas sin aguijón (Apidae: Meliponini) basado en el contorno y venación de sus alas, código VIE 5402-1701-0101.

## Bibliografía

- Abou Shara, H. & Al Ghamdi, A. (2012). Studies on wings symmetry and honey bee races discrimination by using standard and geometric morphometrics. *Biotechnology in Animal Husbandry*, 28(3), 575-584.
- Arbuckle, T., Schröder, S., Steinhage, V., & Wittmann, D. (2001). Identification and Monitoring of Bee Species using ABIS. *Proceedings of the 15th International Symposium Informatics for Environmental Protection* (pp. 425-430). Zurich: EMPA.
- Breuker, C., Gibbs, M., Van Dongen, S., Merckx, T. & Van Dyck, H. (2010). The Use of Geometric Morphometrics in Studying Butterfly Wings in an Evolutionary Ecological Context. En A. Elewa (Ed.), *Morphometrics for Nonmorphometricians* (pp. 271-287). Berlin: Springer.
- Camargo, J. & Pedro, S. (2007). Meliponini Lepeletier. En J. Moure y D. Urban (Eds.), *Catalogue of bees (Hymenoptera, Apoidea) in the neotropical region* (pp. 272-578). Curitiba: Sociedade Brasileira de Entomologia.
- Francoy, T., Silva, R., Nunes-Silva, P., Menezes, C. & Imperatriz-Fonseca, V. (2007). Gender identification of five genera of stingless bees (Apidae, Meliponini) based on wing morphology. *Genetics and Molecular Research*, 207-214.
- Francoy, T., Wittman, D., Muller, S., Steinhage, V., Bezerra-Laure, V., De Jong, D. & Goncalves, L. (2008). Identification of africanized honey bees through wing morphometrics: two fast and efficient procedures. *Apidologie*, 488-494.
- Hall, C. (2011). *Automated approach to bee identification from wing venation*. Madison: Universidad de Wisconsin.
- Hall, M., MacLeod, N. & Wardhana, A. (2014). *Use of wing morphometrics to identify populations of the Old World screwworm fly* (pp. 549-555). Elsevier.
- Herrera, E., Aguilar, I., Prendas, P., Figueroa, G., Travieso, C. & Ramírez, M. (2013). Photograph capture protocol for the creation of an automatic classification. En *VIII Congreso Mesoamericano de Abejas Nativas* (p. 337). Heredia: Universidad Nacional de Costa Rica.
- Johnson, D. (2000). *Métodos multivariados aplicados al análisis de datos*. México: International Thomson Editores.
- Lowe, D. (1999). Object recognition from local scale-invariant features. *Proc. International Conference on Computer Vision*, 1150-1157.

- Lowe, D. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 91-110.
- Meléndez, R., Meneses, C. & Kevan, P. (2013). Effects of human disturbance and habitat fragmentation on stingless bees. En P. Vit, P. Silvia y D. Roubik (Eds.), *Pot-Honey: A legacy of stingless bees*. New York: Springer.
- Michener, C. (2007). *The bees of the world*. New York: Johns Hopkins University Press.
- Quesada Euan, J. (2009). Potencial de las abejas nativas en la polinización. *Acta Biológica Colombiana*, 14(2), 169-172.
- Thompson, H. (2012). Behavioural Effects of Pesticides in Bees. Their Potential for Use in Risk Assessment. *Ecotoxicology*, 317-330.
- Tofilski, A. (2008a). Automatic measurements of honeybee wings. En N. MacLeod (Ed.), *Automated Taxon Identification in Systematics: Theory, Approaches and Applications* (pp. 289-298). Florida: CRC Press.
- Tofilski, A. (2008b). Using geometric morphometrics and standard morphometry to discriminate three morphometry to discriminate three honeybee subspecies. *Apidologie*, 39(5), 558-563.
- Trejos, J., Castillo, W. & González, J. (2014). *Análisis multivariado de datos métodos y aplicaciones*. San José: Universidad de Costa Rica.
- Tüzün, A. & Yüksel, S. (December 2011). Use of wing morphometry for the discrimination of some *Cercheris* (Insecta: Hymenoptera:Crabronidae Species). *African Journal of Biotechnology*, 10(80), 18569-18580.
- Whitehorn, P., O'Connor, S., Walkers, F. & Goulson, D. (2012). Neonicotinoid pesticide reduces bumble bee colony growth and queen production. *Science*, 336, 351-352.

# Movimiento de centroides y transferencias: alternativas para construir vecinos en sobrecalentamiento simulado

## Centroid movement and transferences: alternatives for generating neighbors in simulated annealing

Jeffry Chavarría-Molina<sup>1</sup>, Juan José Fallas-Monge<sup>2</sup>

---

*Fecha de recepción: 11 de junio del 2015*

*Fecha de aprobación: 17 de setiembre del 2015*

Chavarría-Molina, J; Fallas-Monge. J. Movimiento de centroides y transferencias: alternativas para construir vecinos en sobrecalentamiento simulado. *Tecnología en Marcha*. Edición especial. Matemática Aplicada, Mayo 2016. Pág 65-77.

---

1 MSc, Profesor de la Escuela de Matemática, Instituto Tecnológico de Costa Rica. Tel. (506)2550-2225, correo electrónico: jchavarría@itcr.ac.cr

2 Juan José Fallas Monge. MSc, Profesor de la Escuela de Matemática, Instituto Tecnológico de Costa Rica. Tel. (506)2550-2225, correo electrónico: jfallas@itcr.ac.cr

## Palabras clave

Heurísticas; optimización; sobrecalentamiento simulado; particionamiento de datos.

## Resumen

En este artículo se comparan dos estrategias alternativas para la generación de vecinos en el algoritmo de sobrecalentamiento simulado. La primera corresponde a la transferencia de objetos de una clase a otra. La segunda, realiza el movimiento de individuos artificiales (llamados centroides) que representan a las clases. La comparación se realizó en el contexto del problema de optimización combinatoria de clasificación de datos cuantitativos. Este problema fue planteado como una minimización de  $W(P)$ , que representa la inercia intraclases como función de una partición  $P$ . Finalmente, esto permitió comparar el rendimiento de los algoritmos en diversos conjuntos de datos.

## Keywords

Heuristics; optimization; simulated annealing; data clustering.

## Abstract

In this paper, two different strategies to generate neighbors in the simulated annealing algorithm were compared. The first idea is based on transferring objects between classes. The other strategy moves artificial objects (called centroids) which represent the clusters. A comparison was developed using the combinatorial optimization problem of quantitative data clustering. This problem is presented as the minimization of  $W(P)$ , which represents the within-inertia as a function of the partition  $P$ . Finally, this allowed comparison of the algorithms' performance in several data sets.

## Introducción

El particionamiento de datos corresponde a un problema de optimización combinatoria en el que se desea efectuar una distribución de individuos en grupos, regido por algún criterio mínimo de costo, el cual se fundamenta en el grado de similitud que presenten los individuos asignados a un mismo grupo.

Primero se considera el conjunto de individuos  $X=\{x_1, \dots, x_n\}$  y se quiere construir un agrupamiento de dichos objetos en  $K$  grupos,  $K$  conocido a priori. Dicho agrupamiento se denomina una partición  $P=\{C_1, \dots, C_K\}$  del conjunto  $X$ , la cual debe satisfacer que  $C_l \subset X$  y  $C_l \neq \emptyset$  para cada  $l=1, \dots, n$ ;  $l \neq l' \Rightarrow C_l \cap C_{l'} = \emptyset$  y  $X = \cup_{l=1}^K C_l$ . Se parte de la tabla de datos de tamaño  $n \times p$ , caracterizada por  $n$  individuos y  $p$  variables cuantitativas independientes, tal como se muestra en el cuadro 1.

La fila  $i$  del cuadro 1 contiene las entradas del  $i$ -ésimo individuo que puede interpretarse como un vector en  $\mathbb{R}^p$ . Cada individuo se asume con peso constante  $p = \frac{1}{n}$ , y  $x_{ij}$  corresponde al valor que toma  $x_i$  en la  $j$ -ésima variable cuantitativa  $v_j$ .

**Cuadro 1.** Tabla de datos de tamaño  $n \times p$

Ind/Var	$v_1$	$v_2$	...	$v_p$
$x_1$	$x_{11}$	$x_{12}$	...	$x_{1p}$
$x_2$	$x_{21}$	$x_{22}$	...	$x_{2p}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$x_n$	$x_{n1}$	$x_{n2}$	...	$x_{np}$

Si se considera el conjunto  $X=\{x_1, \dots, x_n\}$  como representación matricial de la cuadro 1, con  $x_i \in \mathbb{R}^p$  y  $P$  una partición de  $X$  en  $K$  clases  $C_1, \dots, C_K$ , entonces el problema de particionamiento de los individuos  $x_1, \dots, x_n$  en  $K$  clases puede formularse como la minimización de la función

$$W(P) = \frac{1}{n} \sum_{l=1}^K \sum_{x_i \in C_l} \|x_i - g_l\|^2,$$

donde  $\|\cdot\|^2$  corresponde a la norma inducida por la métrica euclídea clásica, tal que

$$\|x_i - g_l\|^2 = (x_i - g_l)^T (x_i - g_l).$$

La función  $W(P)$  se denomina la inercia intraclases asociada a la partición  $P$  y permite cuantificar el agrupamiento de los individuos en todas las clases a la vez. Cuanto menor sea el valor de  $W(P)$ , los individuos pertenecientes a una misma clase están más agrupados entre sí, indicando, por ende, una mayor similitud a lo interno de las clases. Lo anterior en contraposición de la disimilitud que presentan los individuos pertenecientes a clases diferentes.

### Sobrecalentamiento simulado (SS)

La heurística de sobrecalentamiento simulado (SS) (llamado en inglés *simulated annealing*) también denominada recocido simulado, es una técnica de búsqueda y optimización propuesta por primera vez en 1983 por Kirkpatrick, Gellat y Vecchi, relacionando los campos de la optimización combinatoria y la estadística termodinámica (Kirkpatrick et al., 1983).

Este método está basado en la idea de un proceso de metalurgia denominado *annealing*, que está conformado por dos etapas. En la primera, un sólido es calentado a altas temperaturas hasta que se funde. De esta manera, las partículas que conforman la materia se mueven y se reorganizan en forma aleatoria. En la segunda etapa, la materia recalentada y deformada entra en un enfriamiento. Para ello, la temperatura suministrada al sólido empieza a disminuir lentamente, de modo que este encuentre un estado de equilibrio para cada paso de dicho decrecimiento (Babu & Murty, 1994). El objetivo de este proceso es generar materiales, como el caso del vidrio, de mayor dureza.

El equilibrio térmico del sistema, cuando la temperatura es  $T$ , se rige mediante la distribución de Boltzmann (Moyano, 2011; Pauling, 1988). Según esta distribución, si  $\Omega$  es un espacio de probabilidad que posee todos los estados de la sustancia y si  $Y$  es una variable aleatoria sobre  $\Omega$  que indica el estado actual de la sustancia, entonces la probabilidad de que  $Y=i$  está dada por:

$$P_T\{Y = i\} = \frac{1}{Z(T)} \text{Exp}\left(\frac{-E_i}{k_B T}\right),$$

donde  $E_i$  denota la cantidad de energía del estado  $i$  y sometido a una temperatura  $T$ ,  $k_B$  es una constante física conocida como constante de Boltzmann, que relaciona temperatura absoluta y energía, y  $Z_T$  es una constante de normalización calculada por la relación:

$$Z(T) = \sum_{j \in \Omega} \text{Exp}\left(\frac{-E_j}{k_B T}\right)$$

y denominada función de partición canónica. Además, se cumple que  $P_T(W=i) \geq 0$  y

$$\sum_{j \in \Omega} P_T(W = j) = \frac{1}{Z(T)} \sum_{j \in \Omega} \text{Exp}\left(\frac{-E_j}{k_B T}\right) = 1.$$

Por lo que  $P_T(Y=i)$  es una función de densidad de probabilidad para  $Y$ . De este modo es notorio, para cualquier temperatura  $T$ , la existencia de estados en  $\Omega$  con probabilidad positiva de ocurrir. Así, cuando la temperatura  $T$  es cercana a cero, los estados con probabilidad positiva serán aquellos de muy baja energía (Bertsimas & Tsitsiklis, 1993; de los Cobos et al., 2010).

Mediante la técnica de Monte Carlo para la simulación de variables aleatorias es posible simular el proceso físico del *annealing*, generando así una secuencia de estados de la materia calentada a una temperatura  $T$ . Si la materia calentada se encuentra en el estado  $i$  con energía  $E_i$ , es posible generar un nuevo estado  $j$  mediante una perturbación causada por la transferencia o desplazamiento de una partícula de la materia. La energía de este nuevo estado es  $E_j$ . De esta manera, si  $E_i - E_j \leq 0$ , el estado  $j$  se acepta como el estado actual. En caso contrario, la aceptación del estado  $j$  se hace solo bajo la probabilidad (Abbass et al., 2002; Babu & Murty, 1994; de los Cobos et al., 2010)

$$P = \text{Exp}\left(\frac{E_i - E_j}{k_B T}\right).$$

Este método se conoce como el criterio de aceptación de Metropolis. El valor  $k_B T$  puede verse como una sola cantidad, que en el algoritmo de SS se le denomina *parámetro de control* o simplemente temperatura. Este valor se debe ajustar para cada problema por resolver y se denotará con  $T^*$ .

### Sobrecalentamiento simulado en optimización combinatoria

Si se considera un problema de optimización combinatoria en el que se quiere determinar el valor que *minimice* la función objetivo  $f$  dentro del espacio de búsqueda, el algoritmo de SS puede verse como una iteración del algoritmo de Metropolis, si se toman valores decrecientes del parámetro de control  $T^*$  (de los Cobos et al., 2010), donde el estado  $i$  corresponde a una solución factible y la energía  $E_i$  es el valor de la función objetivo en dicho estado, es decir,  $E_i = f(i)$ .

El criterio de aceptación de Metropolis establece la regla para aceptar la solución  $j$  dada la solución  $i$ , la cual indica que la probabilidad de aceptación está dada por (Aarts, 1990):

$$P_{T^*}(\text{aceptar } j) = \begin{cases} 1 & \text{si } f(j) \leq f(i) \\ \text{Exp}\left(\frac{f(i)-f(j)}{T^*}\right) & \text{si } f(j) > f(i). \end{cases}$$

Al igual que en el proceso de *annealing* en metalurgia, el éxito del algoritmo depende de la velocidad del enfriamiento (control del parámetro *temperatura*). Existen muchas formas de realizar el enfriamiento, algunas de ellas pueden ser consultadas en Talbi (2009) y Osman y Christofides (1994). En particular, la fórmula más usada para realizar la actualización de la temperatura para la iteración  $k + 1$  es la fórmula geométrica dada por  $T_{k+1}^* = \alpha T_k^*$ , donde  $\alpha \in ]0, 1[$ . Se recomienda un valor para  $\alpha$  cercano a uno, debido a que valores altos de  $\alpha$  generan un decrecimiento más lento del modelo geométrico y, por ende, el sistema tendrá más oportunidad de alcanzar un equilibrio en cada uno de los estados de la temperatura. Este modelo fue el seleccionado para el presente trabajo.

En el algoritmo de SS, el sistema debe alcanzar su estabilidad térmica para cada valor de la temperatura  $T^*$ , antes de hacer que esta decrezca por alguno de los modelos de enfriamiento. Por lo tanto, se torna necesario la inserción de un nuevo parámetro LongCade que representa un largo de truncamiento de la cadena de Markov (Mesa, 2007). Este parámetro indica el número de soluciones generadas para una temperatura  $T^*$  fija. Como consecuencia, en la implementación del algoritmo, para un estado de la temperatura se ejecuta un ciclo de longitud LongCade, que corresponde a la longitud tomada para la cadena de Markov. Finalizada dicha cadena, se enfría el sistema (se pasa a un nuevo estado) para proceder nuevamente con la ejecución del ciclo correspondiente. Se mantiene este comportamiento de manera sucesiva hasta que el sistema se haya enfriado lo suficiente. El algoritmo completo de SS se muestra en el Algoritmo 1, y en particular en la línea 6 se puede observar el ciclo asociado a la cadena de Markov.

### ALGORITMO 1 Sobrecalentamiento simulado

Entrada: Parámetros  $T_f$  (temperatura final),  $M$  (número máximo de iteraciones) y LongCade.

- 1:  $s \leftarrow$  Solución inicial generada aleatoriamente.
- 2:  $T^* \leftarrow T_{inicial}$ .
- 3:  $k \leftarrow 0$ .
- 4: MIENTRAS No se dé el criterio de parada HACER
- 5:  $k \leftarrow k + 1$ .
- 6: PARA  $j \leftarrow 1$  HASTA LongCade HACER
- 7:  $s' \leftarrow$  Se genera una nueva solución mediante una perturbación de  $s$ .
- 8:  $\Delta E \leftarrow f(s') - f(s)$ .
- 9: SI  $\Delta E \leq 0$  ENTONCES
- 10:  $s \leftarrow s'$ .
- 11: SI NO
- 12: SI aleatorio  $[0, 1] \leq \text{Exp}\left(\frac{-\Delta E}{T^*}\right)$  ENTONCES
- 13:  $s \leftarrow s'$ .

- 14: FIN SI
- 15: FIN SI
- 16: FIN PARA
- 17:  $T^* \leftarrow g(T^*)$ , con  $g$  la función de enfriamiento.
- 18: FIN MIENTRAS
- 19: Retornar: La mejor solución encontrada en el proceso.

Para el caso del valor de  $T_{inicial}$  (temperatura inicial), existen muchas referencias en la literatura sobre formas de cómo definir este parámetro para la ejecución del algoritmo de SS. Por ejemplo, pueden consultarse Kirkpatrick et al. (1983), Talbi (2009), Yang (2010) y Ben-Ameur (2004). Para efectos de la implementación se siguió una estrategia similar a la propuesta en Ben-Ameur (2004) que se basó en la regla de aceptación de Metropolis. En particular, la probabilidad de aceptación en el caso que  $f(j) > f(i)$ , se implementa mediante la comparación del valor de  $\text{Exp}\left(\frac{f(i)-f(j)}{T^*}\right)$  con un número aleatorio generado con una distribución uniforme en el intervalo  $]0,1[$  (Aarts & Korst, 1990). Dicho número aleatorio se puede interpretar como una tasa de aceptación  $\chi_0$  (Trejos & Murillo, 2004), con  $0 < \chi_0 < 1$ , y visualizarlo como el porcentaje que se quiere para aceptar (bajo una probabilidad) las primeras soluciones en el algoritmo de sobrecalentamiento simulado. En síntesis, si  $f(j) > f(i)$  y dada una tasa de aceptación  $\chi_0$ , se tiene que  $\chi_0 = \text{Exp}\left(\frac{f(i)-f(j)}{T^*}\right)$ . Así, despejando  $T^*$  en la igualdad anterior se obtiene  $T^* = \frac{f(i)-f(j)}{\ln(\chi_0)}$ . Por lo tanto, para el cálculo de la temperatura inicial se generaron de manera aleatoria  $L$  soluciones factibles del problema de optimización y a cada una de ellas se le construyó un vecino (la generación de vecinos se tratará en detalle en la siguiente sección) tal que tuviera una inercia intraclases mayor que la inercia de la solución factible a partir de la cual se generó. Es decir, si  $W_i$  denota la inercia intraclases de la  $i$ -ésima solución generada y  $W_{vec(i)}$  denota la inercia intraclases de su vecino, para  $i = 1, \dots, L$ , entonces debe darse que  $W_{vec(i)} > W_i$ . Finalmente, se promediaron las diferencias positivas  $W_{vec(i)} - W_i$  y se dividió entre la expresión  $\ln(\chi_0)$ . Por lo tanto,

$$T_{inicial} = \frac{\sum_{i=1}^L (W_{vec(i)} - W_i)}{L \cdot \ln(\chi_0)}$$

En este estudio se seleccionó  $\chi_0 = 0,96$ , con el objetivo de favorecer una alta variabilidad para el cálculo de la temperatura inicial.

### Implementación en particionamiento

Para efectos de la implementación computacional del problema de particionamiento con el algoritmo de SS se construyó un vector, denominado *VClasificacion* (vector de clasificación), que representa la forma en la que se manejan las posibles particiones de  $X$ . Dicho vector posee entradas enteras y es de dimensión  $1 \times n$ . Además, tiene la forma

$$VClasificacion = (c_1, c_2, \dots, c_n),$$

de tal manera que la  $i$ -ésima entrada,  $c_i$ , de *VClasificacion* satisface que  $c_i \in \{1, 2, \dots, K\}$  y denota la clase a la que se asigna el individuo  $x_i$  de  $X$ . Por lo tanto, se quiere determinar la combinación de entradas enteras para *VClasificacion* que minimice el valor de la inercia intraclases  $W(P)$  y que represente de manera computacional la partición  $P$  buscada. Por lo anterior, en la implementación se entiende como una solución factible del problema a cada combinación posible tomada por *VClasificacion*. Los algoritmos diseñados parten de una solución inicial

generada de manera aleatoria. Esto es, una combinación dada sobre *VClasificación*, en la que cada entrada de dicho vector es seleccionada de manera aleatoria del conjunto  $\{1,2,\dots,K\}$ .

En el presente estudio se analizan y comparan dos variantes del algoritmo de SS. Dichas variantes consisten en dos metodologías alternativas para la construcción de los vecinos en el algoritmo de SS (refiérase a la línea 7 del Algoritmo 1). La primera de ellas, que se denotará SS-T, representa la metodología clásica de construcción de vecinos mediante las transferencias de objetos entre las clases. En efecto, se entiende como vecino de la solución factible presente en una determinada iteración del algoritmo (SolActual) a otra clasificación, denotada Vecino, generada al cambiar la entrada  $c_k$  de SolActual a un valor  $c_j$ , respetando las condiciones  $c_j \in \{1,2,\dots,K\}$ ,  $c_j \neq c_k$ . De la definición de *VClasificación* se nota que esta forma de generar un vecino corresponde a la transferencia del individuo  $x_i$  de  $X$  de la clase  $i$  a la clase  $j$ .

Para efectos de mejorar el rendimiento del algoritmo SS-T, en términos de los tiempos de ejecución, se utilizaron las fórmulas de actualización que se enuncian y demuestran en Trejos et al. (2014), que indican la forma en la que varían los centros de gravedad y la inercia intraclases, al realizar la transferencia de un objeto de una clase a otra. En este sentido, al hacer la transferencia de  $x$  de la clase  $C_r$  a la clase  $C_l$ , los centros de gravedad se modifican de la siguiente manera:

$$g(C_r - \{x\}) = \frac{1}{\mu_r - p_x} (\mu_r g_r - p_x x),$$

$$g(C_l \cup \{x\}) = \frac{1}{\mu_l + p_x} (\mu_l g_l + p_x x),$$

donde  $p_x$  es el peso del individuo  $x$ . Además, la inercia intraclases presenta la variación

$$\Delta W = \frac{\mu_r p_x}{\mu_r - p_x} \|g_j - x\|^2 - \frac{\mu_l p_x}{\mu_l + p_x} \|g_l - x\|^2.$$

Por otra parte, la segunda metodología consiste en generar un vecino mediante el movimiento del centroide de una de las clases que conforman la partición generada en SolActual. En este algoritmo, que se denotará SS-CG, cada solución factible de la forma  $(c_1, c_2, \dots, c_n)$  del problema de optimización tratado tiene asociada la estructura  $(g_1, g_2, \dots, g_K)$ , que corresponde a la matriz de centros de gravedad de la partición  $P = \{C_1, C_2, \dots, C_K\}$  y relativa a dicha solución factible. De tal manera que  $g_l$  representa el centro de gravedad de la clase  $C_l$ , para  $l \in \{1,2,\dots,K\}$  y, en caso de que se asuman todos los individuos en  $X$  con el mismo peso  $p = \frac{1}{n}$ , se define como  $g_l = \frac{1}{|C_l|} \sum_{x_i \in C_l} x_i$ . Así, los centros de gravedad, al igual que los individuos  $x_i$ , corresponden a vectores de  $\mathbb{R}^p$  y, por ende, tienen la forma  $g_l = (g_{l1}, g_{l2}, \dots, g_{lp})$ , para  $l \in \{1,2,\dots,K\}$ . A partir de lo anterior, en el algoritmo SS-CG se siguen los siguientes pasos para la construcción de un vecino:

- De manera aleatoria se selecciona cuál de los  $K$  centroides (centros de gravedad artificiales) relativos a la solución actual será el que se moverá de posición.
- De las  $p$  posiciones del centroide seleccionado, se escoge aleatoriamente la posición que se variará.

Si se ha seleccionado el centroide  $l$ , con  $l \in \{1,2,\dots,K\}$ , y la posición  $r$ , con  $r \in \{1,2,\dots,p\}$ , entonces la posición  $(l,r)$  de dicho vector se actualiza para generar el nuevo centroide asociado a la clase  $l$  del vecino, como  $g_{lr}^* = g_{lr} + 2 \cdot \sigma_r$  o  $g_{lr}^* = g_{lr} - 2 \cdot \sigma_r$ . Esto es, se suma o resta (bajo una

probabilidad) el tamaño de paso  $2 \cdot \sigma_r$ , donde  $\sigma_r$  denota la desviación estándar de los valores que toman los  $n$  individuos de  $X$  en la variable número  $r$ . El movimiento realizado siempre respeta las condiciones  $g_{ir} \pm \Delta_r \geq \text{mín} \{x_{1r}, \dots, x_{nr}\}$  y  $g_{ir} \pm \Delta_r \leq \text{máx} \{x_{1r}, \dots, x_{nr}\}$ . Es decir, se restringe el que los movimientos en la dimensión  $r$  se realicen entre el valor máximo y el valor mínimo que toman los individuos por clasificar en la variable  $r$ . Esta estrategia lo que hace es delimitar el espacio de búsqueda, que en un principio es  $\mathbb{R}^p$ , al hiperrectángulo dado por el producto de intervalos

$$\prod_{r=1}^p [\text{mín}\{x_{1r}, \dots, x_{nr}\}, \text{máx}\{x_{1r}, \dots, x_{nr}\}]$$

Teniendo actualizada la matriz de centroides, debido al movimiento generado en una única dimensión de uno de sus centroides, se recalculan las entradas de VClasificación. Esto es, se reasignan los  $n$  individuos de  $X$  al centroide más cercano, para generar nuevamente la partición que será representada computacionalmente en VClasificación. En este punto culmina la generación del vecino para este algoritmo.

En ambos algoritmos se utilizó como complemento la técnica de  $k$ -medias, la cual corresponde a un algoritmo de búsqueda local que en términos globales consta de tres etapas (Trejos et al., 2014):

- Se recorren secuencialmente los  $n$  individuos de  $X$  y cada uno de ellos se asigna a la clase más cercana (en el contexto del artículo se entiende como la menor distancia en el sentido euclídeo). Esto es, el individuo  $x_i$  se asigna a la clase  $C_r$ , si el centro de gravedad  $g_r$  de dicha clase satisface que el valor de  $\|x_i - g_r\|$  sea mínimo, para  $r \in \{1, 2, \dots, K\}$ .
- Posteriormente al proceso de transferencias, se calculan de nuevo los centros de gravedad de las clases.
- Se repiten los dos primeros pasos hasta que haya convergencia del algoritmo.

Se utilizó el parámetro KM para controlar la aplicación respectiva de esta técnica. El Algoritmo 2 muestra la adaptación realizada en la investigación al algoritmo de SS para estudiar el problema de particionamiento con las dos variantes de generación de vecinos ya explicadas. En la línea 12 se indica el punto en el que se construye el vecino. Si se hace referencia al algoritmo SS-T, este proceso se ejecuta mediante la transferencia de un individuo de una clase a otra. Por su parte, en el caso de SS-CG se realiza mediante el movimiento de un centroide, tal y como se explicó previamente. En la línea 13 se indica la condición bajo la cual se desarrolla la aplicación del algoritmo de  $k$ -medias. En la línea 32 se muestra el uso de la fórmula geométrica como modelo de enfriamiento. Como estrategia de aceleración se determinó abortar el bucle PARA asociado a un valor de la temperatura, si ha transcurrido más del 50% de iteraciones en ese bucle, sin que se haya presentado una mejora de la mejor solución que el algoritmo ha encontrado hasta ese momento (ver líneas 28 y 29). Por último, con el objetivo de favorecer la exploración del espacio de soluciones factibles y evadir la optimalidad local, se implementó la idea de reiniciar aleatoriamente SolActual si se han realizado 10 enfriamientos consecutivos en los que el algoritmo no ha reportado ninguna mejora (refiérase a las líneas 8 y 9).

### Análisis del parámetro $\alpha$

Es bien conocido la alta sensibilidad que tienen las heurísticas, y los algoritmos en general, ante la selección que se realice para los parámetros propios de cada algoritmo. Por ejemplo, el parámetro  $\alpha$  que se utiliza para controlar el modelo geométrico de enfriamiento en

sobrecalentamiento simulado influye en el rendimiento de este algoritmo para poder encontrar buenas soluciones.

### ALGORITMO 2 SS en particionamiento

Entrada: Parámetros  $T_f$ ,  $M$ , LongCade,  $\alpha$  (para decrecer la temperatura) y KM.

- 1: SolActual  $\leftarrow$  Solución inicial generada aleatoriamente.
- 2: TempActual  $\leftarrow T_{inicial}$
- 3: Iteraciones  $\leftarrow 0$
- 4: ContadorSinMejoras  $\leftarrow 0$
- 5: MIENTRAS TempActual  $> T_f$  y Iteraciones  $< M$  HACER
- 6: Iteraciones  $\leftarrow$  Iteraciones + 1.
- 7: ContadorSinMejoras  $\leftarrow$  ContadorSinMejoras + 1
- 8: SI ContadorSinMejoras mod 10 = 0 ENTONCES
- 9: Reiniciar la solución actual.
- 10: FIN SI
- 11: PARA  $j \leftarrow 1$  HASTA LongCade HACER
- 12: Vecino  $\leftarrow$  Generar vecino de SolActual
- 13: SI  $j \bmod KM = 0$  ENTONCES
- 14: Vecino.AplicarKMedias.
- 15: FIN SI
- 16:  $\Delta E \leftarrow$  Vecino.Inercia - SolActual.inercia
- 17: SI  $\Delta E \leq 0$  ENTONCES
- 18: SolActual  $\leftarrow$  Vecino.
- 19: SI SolActual.inercia  $<$  MejorSolucion.inercia ENTONCES
- 20: MejorSolucion  $\leftarrow$  SolActual.
- 21: ContadorSinMejoras  $\leftarrow 0$
- 22: FIN SI
- 23: SI NO
- 24: SI  $\text{aleatorio}[0,1] \leq \text{Exp}\left(\frac{-\Delta E}{T^*}\right)$  ENTONCES
- 25: SolActual  $\leftarrow$  Vecino.
- 26: FIN SI
- 27: FIN SI
- 28: SI ContadorSinMejoras  $> 0,5 \cdot \text{LongCade}$  ENTONCES
- 29: Abortar cadena actual (abortar el PARA).
- 30: FIN SI
- 31: FIN PARA
- 32: TempActual  $\leftarrow \alpha \cdot \text{TempActual}$ .

33: FIN MIENTRAS

34: Retornar: MejorSolucion.

Si se escoge un valor muy cercano a la unidad, como, por ejemplo,  $\alpha = 0,999$ , entonces el sistema se enfriará de manera muy lenta, generando, en términos computacionales, tiempos de ejecución muy elevados. Recíprocamente, si se selecciona, por ejemplo,  $\alpha = 0,5$ , entonces el sistema se enfría tan rápidamente que el algoritmo no tiene la oportunidad de explorar adecuadamente el espacio de soluciones factibles en búsqueda de buenas soluciones. Como consecuencia, se hace necesario buscar un equilibrio en los valores asignados a cada parámetro.

Para soporte del proceso se generaron experimentalmente dos tablas de datos, que se denominarán T105 ( $n = 105$ , que es la cantidad de objetos por clasificar) y T525 ( $n = 525$ ), siguiendo una distribución normal de números pseudoaleatorios. El agrupamiento se desarrolló, en cada tabla, considerando  $K = 7$  (número de clases), tal que seis clases tienen varianzas  $\sigma^2 = 1$  y la clase restante tiene varianzas  $\sigma^2 = 3$ . Además, T105 fue construida con una clase “grande” de cardinalidad 51 y las seis clases restantes con cardinalidad 9. De manera similar, T525 tiene una clase de tamaño 261 y las seis restantes de 44 objetos. Dado que el diseño es controlado, se pudo determinar a priori el valor de  $W(P)$  (inercia intraclases) que representa, en cada caso, el agrupamiento de referencia en 7 clases. Esto con el objetivo de controlar la respuesta del algoritmo ante tablas experimentales. El cuadro 2 muestra los valores de referencia de  $W(P)$  para T105 y T525.

**Cuadro 2.** Valores de referencia de  $W(P)$

Tabla	$W(P)$ de referencia
T105	7,6247
T525	7,4561

Posteriormente, se realizó un proceso de calibración del parámetro  $\alpha$ . Para ello se tomó como base lo expuesto en Talbi (2009) y Trejos y Murillo (2004), en cuanto a que valores de  $\alpha$  cercanos a la unidad generan mejores resultados. Por lo tanto, en el análisis se consideró variar  $\alpha$  desde 0,9 hasta 0,99 a paso de 0,1, habiendo analizado entonces 10 posibles valores para este parámetro. Después del análisis se decidió tomar  $\alpha = 0,99$  para la comparación de los algoritmos.

## Datos utilizados y resultados

Para la prueba de los algoritmos se utilizaron ocho tablas extraídas de repositorios internacionales disponibles en <http://archive.ics.uci.edu/ml/> (University of California) y <http://cs.joensuu.fi/sipu/datasets/> (University of Eastern Finland), las cuales se describen a continuación.

### Tabla de los Iris de Fisher

Tabla de 150 objetos (150 flores de tres especies: *Iris setosa*, *Iris versicolor* e *Iris virginica*), que son caracterizadas en cuatro variables cuantitativas: largo y ancho del sépalo y largo y ancho del pétalo.

### Tablas de Wine Quality

Constan de dos tablas sobre las variantes roja y blanca del *vinho verde* (vino producido en Minho, zona al noroeste de Portugal). La primera, *winequality-red* (WQ-red), consta de 1599 muestras de vino rojo caracterizadas en 11 atributos cuantitativos. Por su parte, la tabla *winequality-white* (WQ-white) se compone de 4898 muestras de vino blanco, descritas en esos mismos atributos.

### Tabla Glass

Está compuesta por 214 instancias, que corresponden a muestras de 6 clases de vidrios caracterizadas en 9 atributos cuantitativos (cantidad presente en cada muestra de Mg, K, Ca, Ba, entre otros elementos químicos).

### Tablas de S-Sets

Corresponde a un conjunto de cuatro tablas de datos sintéticos, denominadas S1, S2, S3 y S4. Cada tabla tiene 5000 individuos y 15 clases. La diferencia entre ellas es el grado de solapamiento entre las clases.

El cuadro 3 resume las principales características de los conjuntos de datos anteriores. En particular, se indica el número  $n$  de individuos, el número  $p$  de variables, el número  $K$  de clases y el valor  $W(P)$  de mínima inercia intraclases que se logró determinar para cada tabla, y que se tomó como valor de referencia para calcular los porcentajes de atracción.

**Cuadro 3.** Características de las tablas de datos.

Tabla	$n$	$p$	$K$	W(P) referencia
<i>Iris</i>	150	4	3	0,5214
WQ-red	1599	11	3	247,2075
WQ-white	4898	11	3	560,4186
Glass	214	9	6	1,5704
S1	5000	2	15	1783523123,37346
S2	5000	2	15	2655821898,14594
S3	5000	2	15	3377914369,87141
S4	5000	2	15	3140628447,25202

El cuadro 4 muestra los resultados obtenidos al aplicar los algoritmos de SS a las tablas de datos. En ambos casos se consideró un número máximo de iteraciones  $M = 400$ ,  $KM = 5$  y  $T_f = 0,001$ . En cada caso, los porcentajes indicados representan la proporción de veces que el algoritmo atinó el valor de referencia para la inercia intraclases, de un total de 500 ejecuciones. En cada corrida se registró el tiempo en segundos, y en el cuadro 4 se reportan los tiempos promedios para cada caso. Además, también se muestra el valor del parámetro LongCade, que se utilizó en las diferentes corridas.

Finalmente, el equipo utilizado para las pruebas corresponde a una computadora de escritorio HP Compaq Elite 8300 MT, con procesador Intel(R) Core(TM) i7-3770 CPU @3.40 GHz, con memoria RAM instalada de 8 GB.

## Conclusiones

En función de los resultados expuestos en el cuadro 4, se muestra un claro dominio del algoritmo SS-T sobre el algoritmo SS-CG. Excepto para la tabla WQ-red, los tiempos promedio en todos los demás casos son menores para SS-T y los porcentajes de atracción son mayores o iguales que los generados por SS-CG.

Por otra parte, con base en el experimento se reforzó la intuición de que el solapamiento entre las clases es un factor que incide significativamente para poder determinar la clasificación óptima de un conjunto de datos. En particular, la tabla S4 tiene un alto grado de intersección entre las clases y, por ende, resultó ser la tabla más complicada de analizar.

Además, otro aspecto relacionado con la complejidad de los datos, que se pudo ver como parte del análisis, es que el número  $K$  de clases tiene mayor influencia en el hecho de que una tabla de datos sea difícil de analizar, en comparación con otras características como el número  $n$  de individuos y el número  $p$  de variables. Por ejemplo, se pueden comparar las tablas WQ-white y S1, las cuales tienen casi el mismo número de individuos, y además WQ-white tiene 11 variables cuantitativas, en contraposición a S1 que son solo 2. Sin embargo, la diferencia en la cantidad de clases es lo que provoca que para S1 se genere un aumento drástico en el tiempo promedio y una disminución en la calidad del rendimiento de los algoritmos, medido en términos de los porcentajes de atracción.

**Cuadro 4.** Tiempos promedio en segundos y porcentajes de atracción de los algoritmos.

Tabla	SS-T			SS-CG		
	%	Tiempo en segundos	LongCade	%	Tiempo en segundos	LongCade
T105	100%	0,078	30	100%	0,488	100
T525	92%	3,181	300	11%	3,507	150
<i>Iris</i>	100%	0,017	10	100%	0,168	60
WQ-red	93%	3,558	100	100%	1,924	30
WQ-white	100%	1,778	10	98%	5,758	30
Glass	100%	0,071	10	100%	0,498	45
S1	78%	22,159	40	0%	-	-
S2	100%	10,841	40	0%	-	-
S3	93%	7,998	40	0%	-	-
S4	32%	37,328	300	0%	-	-

## Reconocimientos

Los autores agradecen a la Vicerrectoría de Investigación y Extensión (VIE) del Instituto Tecnológico de Costa Rica. Parte de los resultados expuestos en el artículo se obtuvo en el marco del proyecto Heurísticas de optimización combinatoria para la clasificación de datos, inscrito en la VIE bajo el código 5402-1440-3901.

## Bibliografía

- Aarts, E. & Korst, J. (1990). *Simulated Annealing and Boltzmann Machines: A Stochastic Approach to Combinatorial Optimization and Neural Computing*. Chichester, Inglaterra: John Wiley & Sons.
- Abbass, H., Sarker, R. & Newton, C. (2002). *Data Mining: A Heuristic Approach*. Hershey, PA, EE.UU.: Idea Group Publishing.
- Babu, P. & Murty, N. (1994). Simulated annealing for selecting optimal initial seeds in the k-means algorithm. *Indian Journal Pure and Applied Mathematics*, 25(1-2), 85-94.
- Ben-Ameur, W. (2004). Computing the initial temperature of simulated annealing. *Computational Optimization and Applications*, 29, 369-385.
- Bertsimas, D. & Tsitsiklis, J. (1993). Simulated annealing. *Statistical Science*, 8(1), 10-15.
- University of Eastern Finland, School of Computing. (s.f.). *Clustering datasets: Speech and Image Processing Unit*  
[Repositorio de datos]. Recuperado desde <http://cs.joensuu.fi/sipu/datasets/>
- de los Cobos, S., Goddard, J., Gutiérrez, M. & Martínez, A. (2010). *Búsqueda y Exploración Estocástica*. México: Editorial CIBI.
- Kirkpatrick, S., Gelatt, C. & Vecchi, M. (1983). Optimization by simulated annealing. *Science*, 220(4598), 671-680.
- Lichman, M. (2013). *UCI Machine Learning Repository*. Irvine, CA: University of California, School of Information and Computer Science. Disponible en <http://archive.ics.uci.edu/ml>
- Mesa, G. (2007). Cadenas de Markov, una sencilla aplicación. *Revista Memorias*, 5(9).
- Moyano, G. (2011). *Cálculos básicos de termodinámica estadística*. Obtenido de [http://aprendeenlinea.udea.edu.co/lms/moodle/file.php/539/mod\\_02/03\\_statmech/guia\\_03\\_statmech.pdf](http://aprendeenlinea.udea.edu.co/lms/moodle/file.php/539/mod_02/03_statmech/guia_03_statmech.pdf)
- Osman, I. & Christofides, N. (1994). Capacitated clustering problems by hybrid simulated annealing and tabu search. *International Transactions in Operational Research*, 3(1), 317-336.
- Pauling, L. (1988). *General Chemistry*. New York: Dover Publications.
- Talbi, E. (2009). *Metaheuristics: from design to implementation*. New Jersey: John Wiley & Sons.
- Trejos, J. & Murillo, A. (2004). Heuristics of Combinatorial Optimization and Applications to Data Analysis. En *Memorias del I Summer School on Optimization and Numerical Analysis*, Berlín, Alemania.
- Trejos, J., Castillo, W. & González, J. (2014). *Análisis multivariado de datos: métodos y aplicaciones*. San José: Editorial de la Universidad de Costa Rica.
- Yang, X. (2010). *Engineering optimization*. New Jersey: John Wiley & Sons.

# Modelo de corrección de errores y pronósticos de inflación

## Error Correction Model and Inflation Forecast

David Lowell Lovelady

---

*Fecha de recepción: 19 de junio del 2015*  
*Fecha de aprobación: 26 de setiembre del 2015*

Lowell-Lovelady, D. Modelo de corrección de errores y pronósticos de inflación. *Tecnología en Marcha*. Edición especial. Matemática Aplicada, Mayo 2016. Pág 78-82.

## Palabras clave

Granger-Engle; pronóstico de inflación; regresión; corrección de errores.

## Resumen

Este estudio investiga la aplicabilidad de las ideas de cointegración y corrección de errores de Granger-Engle a los pronósticos de inflación en Costa Rica. Se muestra que estas ideas, aplicadas a una regresión en movimiento contra la inflación de Estados Unidos, tuvieron éxito con la inflación de Costa Rica, incluso a través de la ruptura económica de 2008.

## Keywords

Granger-Engle; inflation forecasting; regression; error correction.

## Abstract

This study investigates the applicability of Granger-Engle cointegration and error correction for predicting inflation in Costa Rica. It is shown that these ideas, applied to a sliding regression against U.S. inflation, successfully forecast Costa Rica's inflation, even through the rupture of 2008.

## Introducción

Se conoce (Engle & Granger, 1987) que los modelos de corrección de errores (ECM, por sus siglas en inglés) son útiles cuando se tienen datos igual e idénticamente distribuidos (i.i.d.). En el estudio, estas ideas se aplican al problema de pronosticar los precios a los consumidores de un país pequeño con respecto a los precios al consumidor en EE.UU.

El estudio demuestra que los ECM pueden proporcionar pronósticos de alta calidad para los precios al consumidor en Costa Rica después de una regresión con respecto a los precios al consumidor de Estados Unidos. Durante el período de estudio, que va de 1996 a 2011, la evidencia disponible indica que los parámetros de regresión de los precios al consumidor de Costa Rica con respecto a los precios al consumidor de Estados Unidos cambiaron alrededor de octubre de 2008. Esta nota no trata de explicar este giro aparentemente brusco, pero se demostrará que el cambio no tiene que perturbar los pronósticos.

La situación es la siguiente: Hay dos series de tiempo, datos que dependen del tiempo,  $x$  y  $y$ , y una hipótesis es que para cada  $t$ , el valor  $x_t$  llega antes del valor  $y_t$ . La hipótesis de regresión es que hay dos constantes  $c$  y  $\beta$  tal que la serie  $y_t - (c + \beta x_t)$  es estacionaria. Es más común decir que hay una serie  $\{\delta_t\}$  tal que  $\{\delta_t\}$  es estacionaria y

$$y_t = c + \beta x_t + \delta_t$$

Para todo  $t$ . Por supuesto, los parámetros  $c$  y  $\beta$  indican una relación entre  $x$  y  $y$ , y tales relaciones pueden cambiar. Un cambio brusco en  $c$  y  $\beta$  puede convertir un procedimiento bueno para pronosticar en uno malo. En su origen, los MCE no fueron propuestos específicamente para este problema, pero es posible utilizarlos para pronosticar con exactitud durante el tiempo que los algoritmos de estimación de parámetros se están poniendo al día con los cambios. En ese sentido, en este trabajo se utiliza la naturaleza de corta duración de los MCE para compensar la naturaleza a largo plazo de las interrupciones de regresión.

## Modelos de corrección de errores

Un modelo básico para corregir errores mejora la estimación del valor  $y_t$  basado en los valores de  $y$  de tiempo  $t-1$  y antes y, por supuesto, los valores de  $x$  de tiempo  $t$  y antes. Es posible que haya lectores que no estén familiarizados con este conjunto de ideas, por lo que se hará una breve descripción.

Supongamos que hemos encontrado, por un procedimiento de mínimos cuadrados, los valores óptimos  $\hat{c}$  y  $\hat{\beta}$  para minimizar

$$\sum_t (y_t - (c + \beta x_t))^2$$

Es decir, estamos estudiando una regresión de  $y$  contra  $x$ . Nuestras hipótesis dicen que

$$y_t = c + \beta x_t + \delta_t$$

con la serie  $\{\delta_t\}$  estacionaria.

Ahora, tenemos

$$y_t - (y_{t-1} - y_{t-1}) = \hat{c} + \hat{\beta} x_t + ((\hat{c} + \hat{\beta} x_{t-1}) - (\hat{c} + \hat{\beta} x_{t-1})) + \delta_t$$

y

$$y_t - y_{t-1} = -(y_{t-1} - (\hat{c} + \hat{\beta} x_{t-1})) + \hat{\beta}(x_t - x_{t-1}) + \delta_t$$

Esta última ecuación sugiere que debemos definir nuestro MCE como la resolución de la regresión en dos variables

$$(\Delta y)_t = \alpha \xi_{t-1} + \gamma (\Delta x)_t + \delta_t,$$

Con  $(\Delta y)_t = y_t - y_{t-1}$ ,  $(\Delta x)_t = x_t - x_{t-1}$ , y  $\xi_s = y_s - (\hat{c} + \hat{\beta} x_s)$ . Después de que se encuentran los  $\hat{\alpha}$  y  $\hat{\gamma}$  óptimos, podemos pronosticar  $y_t$  en términos de  $x_t$ ,  $x_{t-1}$ ,  $y_{t-1}$ , y  $\xi_{t-1}$ .

### Un ejemplo sintético

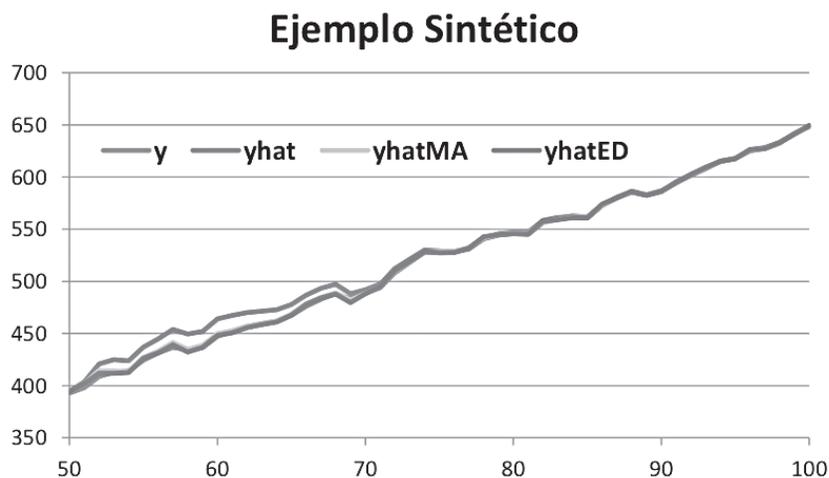
Para ilustrar mejor este enfoque, consideramos una simple ilustración del problema y nuestros resultados (ver figura 1). Supongamos que  $\{x_t\}_{t=1}^{100}$  y  $\{y_t\}_{t=1}^{100}$  son dos series tal que, para cada  $t \in \{1, 2, 3, \dots, 100\}$ ,  $x_t = t + \delta_t$  y

$$y_t = 8x_t + \epsilon_t \text{ si } 1 \leq t \leq 50, y_t = y_{t-50} + 5(x_t - x_{t-50}) + \epsilon_t \text{ si } 51 \leq t \leq 100$$

Con  $\{\delta_t\}$  y  $\{\epsilon_t\}$  resultados de una transformación de tipo  $MA(3)$  de  $N(0, 1)$  (distribución normal, promedio 0, varianza 1). Por ejemplo, podemos generar 102 muestras de  $\{\eta_t\}_{t=1}^{102}$  de  $N(0, 1)$  y definir  $\delta_t = (\eta_t + \eta_{t+1} + \eta_{t+2})/\sqrt{3}$ . Definimos  $\{\epsilon_t\}$  en la misma manera. Notamos que las dos series son estacionarias pero NO son independientes.

El problema sencillo de pronóstico es predecir el valor  $y_t$  solamente con los valores  $\{y_1, y_2, \dots, y_{t-1}\}$  y  $\{x_1, x_2, \dots, x_{t-1}, x_t\}$ . Si una persona considera la posibilidad de que hay una regresión quebrada, pero no sabe si es la verdad, un procedimiento razonable es hacer, para

cada  $t \geq 30$ , una regresión de 30 datos que utilice las 30 muestras antes de  $t$ . Es decir, por ejemplo, cuando tenemos  $\{x_t\}_{t=1}^{66}$  y  $\{y_t\}_{t=1}^{65}$ , y se quiere pronosticar  $y_{66}$ , se hace una regresión básica de  $\{y_t\}_{t=1}^{65}$  con respecto a  $\{x_t\}_{t=1}^{65}$ , y obtenemos el intercepto  $\hat{c}$  y la pendiente  $\hat{\beta}$ , y pronosticamos  $y_{66} = \hat{c} + \hat{\beta}x_{66}$ . Esto hicimos, y hubo errores visibles después del punto del cambio estructural (y y  $\hat{y}$  en el dibujo de abajo) con cuadrático medio mínimo de 70 pronósticos de 6.24. Los dos algoritmos que se describen abajo nos dan 1.34 y 1.59.



**Figura 1.** Ejemplo sintético del problema

## Los algoritmos

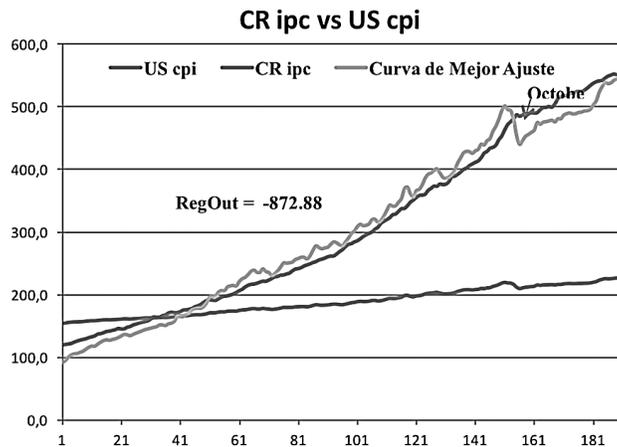
Tenemos dos versiones de nuestros algoritmos; una versión con promedios móviles (MA) y otra con disminución exponencial de los pesos (ED). El procedimiento comienza con una regresión básica de  $y$  contra  $x$ , por 30 muestras. Luego se actualizan los valores  $\hat{c}$  y  $\hat{\beta}$  muestra por muestra para hacer el primer pronóstico. Calculamos los errores y los  $\hat{\alpha}$  y  $\hat{\gamma}$ , también muestra por muestra. Luego, mejoramos el pronóstico y hacemos el procedimiento completo para la siguiente muestra. Hay dos maneras de evitar las primeras muestras que tienen demasiado efecto.

Se pueden calcular las sumas de la regresión con una ventana móvil o con un filtro exponencial con factor 1.

## Pronósticos de inflación del país

Aplicamos nuestro procedimiento al problema de pronosticar la inflación de Costa Rica. Separamos los efectos de la inflación en dos partes, efectos exteriores y efectos interiores y modelamos los efectos exteriores de acuerdo con la inflación de Estados Unidos.

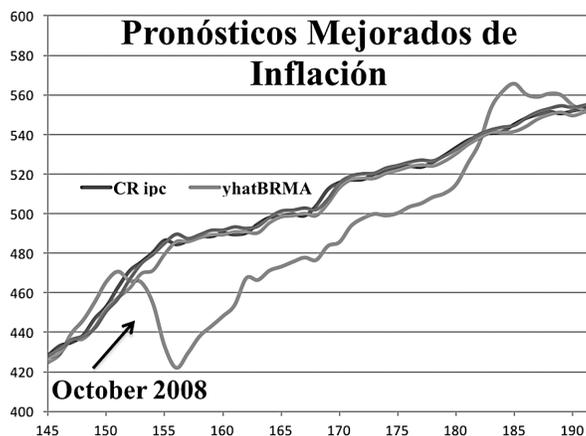
Un gráfico de una regresión básica del Índice de Precios al Consumidor (IPC) de Costa Rica con respecto al pci (siglas en inglés de IPC) de Estados Unidos nos da el resultado que se muestra en la figura 2.



**Figura 2.** Gráfico de regresión básica del Índice de Precios al Consumidor (IPC) de Costa Rica con respecto al pci de Estados Unidos.

Se puede notar en la figura 2 que en octubre de 2008 se registró una importante ruptura económica.

Luego, aplicamos las ideas de esta nota y se obtiene lo que se muestra en la figura 3.



**Figura 3.** Pronósticos Mejorados de Inflación

En el gráfico de la figura 3, la curva azul representa el IPC de Costa Rica, la curva roja muestra el resultado de la regresión básica con respecto al pci de Estados Unidos sin corrección. Las curvas verde y marrón corresponden a los dos procedimientos: ventana móvil y filtro exponencial.

## Bibliografía

Engle, R. & Granger, C. (1987). *Co-Integration and Error Correction: Representation, Estimation, and Testing*. Obtenido de: [http://www.ntuzov.com/Nik\\_Site/Niks\\_files/Research/papers/stat\\_arb/EG\\_1987.pdf](http://www.ntuzov.com/Nik_Site/Niks_files/Research/papers/stat_arb/EG_1987.pdf)

La revista *Tecnología en Marcha* es publicada por la Editorial Tecnológica de Costa Rica, con periodicidad trimestral. Su principal temática es la difusión de resultados de investigación en áreas de Ingeniería. El contenido de la revista está dirigido a investigadores, especialistas, docentes y estudiantes universitarios de todo el mundo.

1. Los artículos deberán ser originales, inéditos y no pueden participar simultáneamente en otros procesos de publicación.
2. La extensión de los trabajos debe oscilar entre 10 y 20 páginas de 21,59 x 27.94 cm (8,5 x 11 pulgadas). Se debe presentar en un documento de Microsoft Word, con interlínea de espacio y medio, en una columna, en letra Times 12 pts.
3. Los títulos de los artículos deben ser sencillos, claros, cortos y estar en español e inglés.
4. Es necesario indicar claramente el nombre y los dos apellidos del autor, nacionalidad, profesión, teléfonos, correo electrónico, dirección exacta, lugar de trabajo y país de origen de dicha entidad.
5. Las palabras clave deben presentarse en español y en inglés. Además, el resumen debe estar compuesto por 250 palabras y aparecer en ambos idiomas.
6. Las imágenes se deben enviar en un archivo aparte del documento principal. En caso de ser escaneadas, la resolución mínima es de 300 ppi. Los formatos permitidos son: .jpg, .tiff, .eps, .psd y .ai.
7. Las fórmulas y ecuaciones matemáticas deben realizarse con el editor de ecuaciones de Word.
8. En lo pertinente, se usará el Sistema Internacional de Unidades.
9. La bibliografía debe aparecer al final del documento, ordenada según su aparición en el documento y utilizar el formato IEEE.
10. Los documentos deberán enviarse a las direcciones electrónicas [revistatm@itcr.ac.cr](mailto:revistatm@itcr.ac.cr) o [alamirez@itcr.ac.cr](mailto:alamirez@itcr.ac.cr)
11. La Comisión editorial no dará trámite de edición al artículo que no cumpla con estos requisitos.

### **Nota importante**

Los originales serán sometidos a un proceso editorial que se desarrollará en varias fases. En primer lugar, serán objeto de una evaluación preliminar por parte de los miembros del Comité Editorial, quienes determinarán la pertinencia de su publicación. Una vez establecido que cumple con los requisitos temáticos y formales indicados en estas instrucciones, será enviado a dos pares académicos externos para decidir en forma anónima (doble ciego) si debe publicarse, si necesita cambios o si se rechaza. En caso de que ambos llegaran a discrepar, el artículo será enviado a un tercer evaluador, para tomar la decisión. Los resultados del dictamen académico serán inapelables en todos los casos. En caso de que el artículo sea aprobado para su publicación, el autor autoriza a la Editorial Tecnológica de Costa Rica para que lo incluya en la revista y pueda editarlo, reproducirlo, distribuirlo, exhibirlo y comunicarlo en el país y en el extranjero mediante medios impresos y electrónicos bajo la licencia *Creative Commons*.

# Instructions to publish in **TECNOLOGÍA** *en marcha*

The journal *Tecnología en Marcha* is published by the Editorial Tecnológica de Costa Rica every three months. It focuses mainly in disseminating the results of research of engineering areas. The journal's contents are intended for researchers, experts, teachers and university students around the world.

1. All articles must be originals, unpublished, and cannot be simultaneously used in other processes.
2. Papers may be 10-20 pages long (8.5 x 11 in, or 21.59 x 27.94 cm). All documents must be submitted in MS Word, 1.5 line spacing, using Times 12 pts. font and in one column.
3. Article titles must be simple, clear, short, and be included in both Spanish and English.
4. The name and (two) last names of the author should be clearly indicated, along with their profession, telephone numbers, email, physical address, place of work (organization, department, school), and where the organization is based.
5. The key words must be included in both Spanish and English. Furthermore, the abstract must be 250 words long, and should also be submitted in both languages.
6. The images must be delivered in a separate document. If scanned, the minimum resolution is 300 ppi. The formats allowed are .jpg, .tiff, .eps, .psd, and .ai.
7. Mathematical equations and formulas must be done with MS Office's Equation Editor.
8. Where necessary, use the International System of Units.
9. Bibliography will be included at the end of the document, arranged order based on IEEE format.
10. Papers should be sent to the following emails: [revistatm@itcr.ac.cr](mailto:revistatm@itcr.ac.cr), or [alamirez@itcr.ac.cr](mailto:alamirez@itcr.ac.cr)
11. The Editorial Committee will only consider for publication the articles meeting the above requirements.

## Important note

All originals will be subject to an editorial process consisting of several phases. First, a preliminary assessment will be done by members of the Editorial Committee, the Director, and the editors, who will jointly determine whether the article would make a relevant publication. After determining that an article meets the thematic and formal requirements established in these Instructions, it will be sent to two outside academic peers who will decide anonymously (double-blind) whether it should be published, if it needs any changes, or whether it should be turned down. In case of disagreement between these peers, the article will be sent to a third evaluator in order to reach a decision. In no case may the results of this decision be appealed. If the article is accepted for publication, the author authorizes the Editorial Tecnológica de Costa Rica to edit, reproduce, distribute, exhibit and communicate at the country and abroad through print media and electronic equipment under the *Creative Commons* license.

## Cronograma 2016

	<b>Vol. 29-1</b>	<b>Vol. 29-2</b>	<b>Vol. 29-3</b>	<b>Vol. 29-4</b>	<b>Vol. 30-1</b>	<b>Vol. 30-2</b>	<b>Vol. 30-3</b>
Recepción de artículos	Mayo - julio 2015	Agosto - octubre 2015	Noviembre 2015 - enero 2016	Febrero - abril 2016	Mayo - julio 2016	Agosto - octubre 2016	Noviembre 2016 - enero 2017
Evaluación de expertos y aprobación	Agosto - setiembre 2015	Noviembre - diciembre 2015	Febrero - marzo 2016	Mayo - Junio 2016	Agosto - setiembre 2016	Noviembre - diciembre 2016	Febrero - marzo 2017
Revisión de estilo y corrección	Octubre 2015	Febrero 2016	Abril 2016	Julio 2016	Octubre 2016	Enero 2017	Abril 2017
Diagramación y correcciones finales	Enero 2016	Marzo 2016	Mayo - junio 2016	Agosto - setiembre 2016	Noviembre - diciembre 2016	Febrero - marzo 2017	Mayo - junio 2017
Publicación del número	Febrero 2016	Mayo 2016	Julio 2016	Octubre 2016	Febrero 2017	Abril 2017	Julio 2017