

## Influencia de los hiper-parámetros en algoritmos basados en Evolución Diferencial para el ajuste de controladores del tipo PID en procesos SISO

Paúl Nicolai Martínez-Luzuriaga<sup>a</sup>, Gilberto Reynoso-Meza<sup>b,\*</sup>

<sup>a</sup>Universidad Politécnica Salesiana. Calle Vieja 12-30 y Elia Liut, Cuenca, Ecuador.

<sup>b</sup>Programa de Pós-graduação em Engenharia de Produção e Sistemas (PPGEPS), Pontifícia Universidade Católica do Paraná (PUCPR), Rua Imaculada Conceição, 1155,80215-901 Curitiba, PR, Brazil.

**To cite this article:** Martínez-Luzuriaga, P., Reynoso-Meza, G. 2023. Hyper parameter influence in Differential Evolution based algorithms for PID controller tuning applications. Revista Iberoamericana de Automática e Informática Industrial 20, 44-55. <https://doi.org/10.4995/riai.2022.16517>

### Resumen

Los controladores PID se mantienen como una solución confiable de primera línea en sistemas de control retroalimentado. Incluso cuando su sencillez es una de las principales razones de ello, un correcto ajuste de sus parámetros es fundamental para garantizar un rendimiento satisfactorio. Como consecuencia, se encuentran disponibles varios métodos de ajuste. En la actualidad, realizar un proceso de ajuste mediante optimización estocástica es una solución atractiva para controlar procesos complejos. No obstante, la solución obtenida con estos métodos de optimización es muy sensible a los hiper-parámetros utilizados. En este artículo proponemos a los diseñadores un conjunto de hiper-parámetros para configurar diferentes algoritmos basados en Evolución Diferencial en sistemas de una entrada y una salida (SISO). Los resultados obtenidos muestran varios aspectos a considerar sobre los valores más prometedores para varias instancias de optimización facilitando la transferencia de conocimiento para nuevas instancias de optimización.

*Palabras clave:* Ajuste de controladores PID, Algoritmos evolutivos, Ajuste de hiper-parámetros, Optimización

### Influence of hyper-parameters in algorithms based on Differential Evolution for the adjustment of PID-type controllers in SISO processes through mono and multi-objective optimisation

#### Abstract

PID Controllers remain as the reliable front-line solution in feedback control loops. Even when their simplicity is one of the main reasons for this, the right tuning of their parameters is essential to guarantee their performance. As consequence, several tuning methods are available. Nowadays performing a tuning process via stochastic optimisation is an attractive solution for complex processes. Nevertheless, the solution obtained using such optimisation methods is very sensitive to the hyper-parameters used. In this paper, we propose to designers a set of hyper-parameters for different algorithms based on Differential Evolution in SISO processes. Obtained results show several aspects to consider regarding the most promising values for several optimisation instances, facilitating knowledge transfer for new optimisation instances.

*Keywords:* PID tuning, Evolutionary algorithms, Hyper-parameters tuning, Optimisation

### 1. Introducción

Por lo general en un sistema de control es de vital importancia el diseño y ajuste de sus estructuras de control, denominadas controladores. Los controladores del tipo Proporcional - In-

tegral - Derivativo (PID) son los más utilizados, debido principalmente a su eficiencia y sencillez de implementación (Åström and Hägglund, 1995). Además, se ha demostrado su confiabilidad y robustez para controlar diferentes procesos que involucran variables tales como: temperatura; nivel, presión, flujo,

\*Autor para correspondencia: g.reynosomeza@pucpr.br

nivel de acidez, entre otras (Saad et al., 2012). Se estima que los controladores de tipo PID se encuentran presentes en alrededor del 90 % de lazos de control automático en la industria (Kozák, 2014).

Para alcanzar el desempeño óptimo de un sistema de control, es necesario realizar la sintonización de los parámetros del controlador PID. Para efectuar esta tarea tenemos distintos métodos (Borase et al., 2021). Entre los métodos de mayor aplicación se tienen aquellos basados en reglas de ajuste tales como el Método de Ziegler-Nichols (ZN) propuesto por Ziegler and Nichols (1993) y el de Chien, Hrones y Reswick (CHR), expresado en Åström and Hägglund (1995). Actualmente son populares también los basados en optimización (Ang et al., 2005).

La existencia de no-linealidades y características de fase no mínima presentes en algunos sistemas dificultan la aplicación de métodos basados en reglas de ajuste (Singh et al., 2017). Por otra parte, los requerimientos actuales en cuanto a desempeño de los sistemas de control demandan soluciones capaces de responder a varias especificaciones de diseño. Se han planteado distintas formas de abordar este tipo de problemas como problemas de optimización evolutiva (Reynoso-Meza et al., 2012). Una de las estrategias recomendadas para responder ante estas exigencias es el planteamiento de problemas de optimización multiobjetivo (MOP, por sus siglas en inglés *Multi-Objective Optimization Problems*), ya que permiten satisfacer un conjunto de especificaciones y restricciones. Los MOP se pueden solucionar mediante optimización multiobjetivo (MOO, por sus siglas en inglés *Multiobjective Optimization*), que ha demostrado ser una herramienta capaz de ofrecer soluciones a este tipo de problemas de ingeniería de una forma relativamente sencilla (Rodríguez-Molina et al., 2020).

Además, en la actualidad la mayoría de los controladores PID se encuentran implementados en *software*, permitiendo la inclusión de algoritmos complejos. De esta manera, se han obtenido mejoras significativas mediante la implementación de algoritmos evolutivos (EA por sus siglas en inglés) para el ajuste de controladores. De estos EA se destaca la aplicación de : enjambre de partículas (Dashti et al., 2010; Alhanjouri, 2017); algoritmo de forraje de bacterias (Konstantinov and Baryshnikov, 2017); algoritmos genéticos (Saad et al., 2012; Urrea-Quintero et al., 2018); colonia de hormigas (Alhanjouri, 2017), e incluso evolución diferencial (DE por su nombre en inglés *Differential Evolution*) (Chiha et al., 2012; Lakshmi et al., 2019).

En todos los EA's se define un conjunto de hiper-parámetros configurables (Montero et al., 2014). El desempeño de un EA se encuentra estrechamente relacionado con la configuración adoptada para dichos hiper-parámetros (Tanabe and Fukunaga, 2015). Sin embargo, el encontrar un conjunto óptimo de valores para configurar el EA es una tarea que en muchas ocasiones requiere cierta pericia acerca del algoritmo, sus hiper-parámetros y el problema en particular (Neumüller et al., 2012). Por consiguiente, se requiere efectuar una sintonización o ajuste de los mismos.

Lograr un adecuado ajuste de hiper-parámetros en los EA es una tarea compleja, considerándose incluso uno de los grandes desafíos dentro del campo de la computación evolutiva (Smit and Eiben, 2010), ya que generalmente requiere de una exploración exhaustiva de los espacios de búsqueda de hiper-parámetros. Por otra parte, una configuración adecuada puede verse

condicionada por múltiples características que influyen sobre el rendimiento general del EA, tales como: propiedades de la función de costo, restricciones de tiempo, restricciones computacionales, entre otras (Ugolotti et al., 2019). En consecuencia, el problema de ajuste de hiper-parámetros cobra una gran relevancia debido a su alto impacto en el desempeño de los EA y la frecuencia con la que se debe efectuar esta tarea de ajuste (De Landgraaf et al., 2007).

El proceso de ajuste de los controladores PID, juega un importante rol en el comportamiento de los sistemas de control (Jaen-Cuellar et al., 2013). Trabajos como el realizado por Konstantinov and Baryshnikov (2017) confirman la eficiencia de los EA's para la resolución de problemas de optimización global, tales como el ajuste de controladores PID. Dentro de la literatura, encontramos múltiples estudios acerca de esta temática bajo enfoques de optimización mono y multi-objetivo, y aplicando distintos EA's (Urrea-Quintero et al., 2018; Rodríguez-Molina et al., 2020; Singh et al., 2017; Kozák, 2014; Reynoso-Meza et al., 2012). Sin embargo, no se han encontrado estudios acerca de la influencia de los hiper-parámetros en el desempeño de dichos algoritmos. En este trabajo se pretende determinar conjuntos óptimos de hiper-parámetros para configurar variantes de algoritmos basados en DE en tareas relacionadas con el ajuste de controladores PID en procesos SISO, con la finalidad de brindar una herramienta útil a los nuevos desarrolladores de control que utilizan optimización evolutiva.

El trabajo de investigación se organiza tal como se describe a continuación: en la sección 2 se presenta un breve referencial teórico acerca de control PID, EAs, optimización mono y multi-objetivo y ajuste de hiper-parámetros en algoritmos evolutivos. En la sección 3 se proponen los métodos y herramientas a ser utilizados en la fase experimental, mientras que en la sección 4 se discuten y analizan los resultados obtenidos. Finalmente en la sección 5 se presentan las conclusiones y propuestas para estudios futuros.

## 2. Referencial Teórico

En esta sección se comentarán los aspectos teóricos esenciales que se involucran en el desarrollo de este trabajo. De forma ordenada se presentan los fundamentos teóricos referentes a: controladores PID, algoritmos evolutivos, optimización mono y multi-objetivo y técnicas de ajuste de hiper-parámetros en algoritmos evolutivos.

### 2.1. Controladores PID

La forma más común de implementación de un lazo de control retroalimentado SISO se muestra en la Figura 1. La diferencia entre el punto de consigna  $r(t)$  y la salida del sistema  $y(t)$  ingresa al sistema expresada como una señal de error  $e(t)$ . Dicha señal ingresa al controlador  $C$  que determina la magnitud de la señal de control  $u(t)$  a enviar hacia la planta  $P$  para ajustar la salida  $y(t)$ .

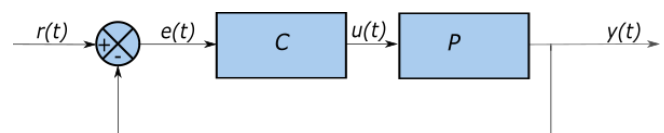


Figura 1: Lazo de Control de un sistema SISO.

Los controladores PID se componen de tres parámetros, conocidos como ganancias: Proporcional, Integral y Derivativa, denotadas por:  $K_p$ ,  $K_i$  y  $K_d$ . Mediante un correcto ajuste del valor de las mismas, se puede mejorar el desempeño del sistema. Existen distintas estructuras para la implementación de los controladores, siendo la más común la presentada en la Ecuación 1. La señal de error  $E(s)$  se somete a las acciones proporcional, integral y derivativa y se obtiene como respuesta la señal de control  $U(s)$  que se enviará a la planta a ser controlada. De esta manera, su función de transferencia viene definida por:

$$C(s) = K_p + \frac{K_i}{s} + K_d s \quad (1)$$

Para ajustar un controlador PID existen diferentes enfoques. Según Urrea-Quintero et al. (2018), la implementación de EA's posibilita sintonizar controladores PID en: a) procesos de fase no mínima, b) sistemas oscilatorios, c) sistemas no lineales y d) sistemas de segundo orden con tiempos muertos. Este trabajo se centrará en aquellos basados en optimización evolutiva.

## 2.2. Problema de optimización multi-objetivo

En concordancia con Miettinen (1998) un problema de optimización, sin pérdida de generalidad<sup>1</sup>, puede declararse como:

$$\min_{\theta \in \mathfrak{R}^n} J(\theta) \quad (2a)$$

$$J(\theta) = [J_1(\theta), \dots, J_m(\theta)] \in \mathfrak{R}^m \quad (2b)$$

$$\text{sujeto a} \quad K(\theta) \leq 0 \quad (2c)$$

$$L(\theta) = 0 \quad (2d)$$

$$\underline{\theta}_i \leq \theta_i \leq \bar{\theta}_i, i = [1, \dots, n] \quad (2e)$$

En donde:  $J(\theta)$ , se denomina vector objetivo; Cuando  $m = 1$ , se habla de optimización mono-objetivo; cuando  $m > 1$  de optimización multi-objetivo.  $K(\theta)$ ,  $L(\theta)$ , se denominan vectores de restricciones de desigualdad e igualdad respectivamente.  $\underline{\theta}_i, \bar{\theta}_i$ , son los límites inferior y superior en el espacio de decisión para la variable  $\theta_i$  en ambos casos.

De forma general, no existe una única solución en un problema multi-objetivo, sino que se presenta un conjunto de alternativas de solución denominado Conjunto de Pareto (PS, por sus siglas en inglés *Pareto Set*) y denotado por  $\Theta_P$ . Cada alternativa de solución en el PS define un vector objetivo en el Frente de Pareto (PF por su nombre en inglés *Pareto Front*), denotado por  $J_P$ . Todas las soluciones en el PF, son pareto-óptimas y no dominadas (Reynoso-Meza et al., 2014a). A continuación se presentan las definiciones teóricas concernientes al PF.

**Definición 1.** Dominancia de Pareto (Coello et al., 2007) Dado un vector  $J(\theta^1)$  se dice que domina a un vector  $J(\theta^2)$  ( $J(\theta^1) \leq J(\theta^2)$ ), si y solo si:

$$\{\forall i \in [1, \dots, m], J_i(\theta^1) \leq J_i(\theta^2)\}$$

$\wedge$

$$\{\exists q \in [1, \dots, m] : J_q(\theta^1) < J_q(\theta^2)\}$$

**Definición 2.** Optimalidad de Pareto (Miettinen, 1998) Dado un vector decisión  $\theta^1$  se dice que es pareto-óptimo si no existe otro  $\theta^2$  que ejerza dominancia sobre el mismo, es decir:

$$\nexists J(\theta^2) : J(\theta^2) \leq J(\theta^1)$$

Por medio de técnicas MOO podemos encontrar una aproximación discreta  $\Theta_P^*$  del PS  $\Theta_P$  (Reynoso-Meza et al., 2012). Esta aproximación proporciona una descripción  $J_P^*$  del PF  $J_P$ .

**Definición 3.** Conjunto Óptimo de Pareto (Miettinen, 1998) Representado por  $\Theta_P^*$ , se compone de todos los posibles vectores decisión óptimos, es decir:

$$\Theta_P^* = \{ \theta^1 | \nexists \theta^2, J(\theta^2) \leq J(\theta^1) \}$$

**Definición 4.** Frente de Pareto (Miettinen, 1998) El PF se compone de los resultados de evaluar todos los vectores decisión en  $\Theta_P^*$  según su correspondiente función objetivo, de modo que:

$$J_P^* = \{ J(\theta) | \theta \in \Theta_P^* \}$$

## 2.3. Algoritmos de Evolución Diferencial para optimización

El algoritmo de Evolución Diferencial (DE por sus siglas en inglés) (Storn and Price, 1997; Das and Suganthan, 2011; Bilal et al., 2020) es ampliamente utilizado debido principalmente a su flexibilidad, versatilidad (Bilal et al., 2020) y su bajo número de parámetros de control (Červenka and Boudna, 2018). Este algoritmo toma como población de cada generación ( $G$ ) un conjunto de vectores de decisión ( $x^i$ ) de dimensión igual al tamaño total de la población ( $N_p$ ). En su versión original, este algoritmo (Storn and Price, 1997) utiliza tres operadores: mutación, recombinación y selección.

**Mutación.-** Para cada vector de decisión  $x^i |_G$  (vector padre) se calcula un vector mutante  $v^i |_G$  en cada generación  $G$ , según distintas estrategias, tales como:

$$\text{“DE/rand/1”}: v^i |_G = x^{r_1} |_G + F(x^{r_2} |_G - x^{r_3} |_G)$$

$$\text{“DE/best/1”}: v^i |_G = x^{best} |_G + F(x^{r_1} |_G - x^{r_2} |_G) \quad (3)$$

$$\text{“DE/rand/2”}:$$

$$v^i |_G = x^{r_1} |_G + F(x^{r_2} |_G - x^{r_3} |_G) + F(x^{r_4} |_G - x^{r_5} |_G)$$

En donde:

$r_1, r_2, r_3, r_4, r_5$  son enteros aleatorios delimitados entre 1 y  $N_p$ .

$r_1 \neq r_2 \neq r_3 \neq r_4 \neq r_5 \neq i \in [1, N_p]$ .

$F \in [0, 1]$ , se conoce como factor de escala y controla la amplificación de las variaciones diferenciales.

$x^{best}$  representa el vector individual con el mejor valor de desempeño en la generación  $G$ .

**Recombinación.-** Luego de la mutación, se aplica a cada par conformado por un vector padre  $x^i |_G$  y un mutante  $v^i |_G$  respectivamente, para generar un hijo  $u^i |_G = [u_1^i |_G, u_2^i |_G, \dots, u_n^i |_G]$ , con base en distintas estrategias, tales como:

- **Recombinación Binomial.-** El número de parámetros que el vector hijo hereda desde el mutante tiene una distribución cercana a la binomial, expresada en la ecuación 4.

$$u_j^i |_G = \begin{cases} v_j^i |_G, & \text{si } rand(0, 1) \leq C_r \\ x_j^i |_G, & \text{de otra forma} \end{cases} \quad (4)$$

<sup>1</sup>Un problema de maximización puede convertirse en uno de minimización. Para cada uno de los objetivos que deban ser maximizados, se puede emplear la transformación  $\max(J_i(x)) = -\min(-J_i(x))$

- **Recombinación Aritmética.**- Propuesta en (Price et al., 2005). El hijo se genera como una combinación convexa entre padre y mutante, tal como se expresa en la ecuación 5.

$$\mathbf{u}^i |G = \mathbf{x}^i |G + F_i(\mathbf{v}^i |G - \mathbf{x}^i |G) \quad (5)$$

En donde:

$C_r \in [0, 1]$ , se conoce como tasa de cruce y representa la probabilidad de efectuar la recombinación binomial.

$F_i$  se conoce como factor de escalamiento para la recombinación aritmética. Si el valor de  $F_i$  se mantiene constante durante todos los valores que toma  $i$ , se conoce como **recombinación lineal**.

**Selección.**- Si los valores de los candidatos exceden los máximos o mínimos permisibles, se debe adecuar los mismos dentro del rango permitido, tomando el valor más próximo dentro de dicho rango. Luego, se evalúan los hijos y se efectúa una comparación por pares para evaluar el desempeño. Los mejores individuos se seleccionan para avanzar a la siguiente generación.

$$\mathbf{x}^i |G+1 = \begin{cases} \mathbf{u}^i |G, & \text{si } f(\mathbf{u}^i |G) \leq f(\mathbf{x}^i |G) \\ \mathbf{x}^i |G, & \text{de otra forma} \end{cases} \quad (6)$$

En el algoritmo 1 se presenta el pseudocódigo correspondiente a un DE en su forma básica.

```

1 Generar población inicial  $P |_0$  con  $N_p$  individuos
2 Inicializar contador de generaciones  $G = 0$ 
3 Evaluar  $P |_0$ 
4 mientras  $G < \text{máx. gen o criterio de finalización no alcanzado}$  hacer
5   para cada  $x_i \in P_G$  hacer
6     Generar un vector mutante (según Ecuación 3)
7     Generar un vector hijo (según Ecuación 4 o 5)
8     Efectuar una comparación por pares (según Ecuación 6)
9    $G = G + 1$ 

```

Algoritmo 1: DE Básica

Siendo que este algoritmo ha sido utilizado con éxito en varios trabajos para el ajuste de controladores PID, tales como: Saad et al. (2012); Lakshmi et al. (2019); Alhanjouri (2017); Fister et al. (2016), entre otros se plantea un estudio para la selección del mejor conjunto de hiper-parámetros.

#### 2.4. Ajuste de los hiper-parámetros en algoritmos meta-heurísticos

De acuerdo a Wolpert and Macready (1997) y su teorema de "No free lunch", no existe ningún algoritmo meta-heurístico cuyo desempeño sea el mejor para cualquier problema de optimización. Es necesario efectuar un correcto ajuste de los hiper-parámetros del EA en cuestión para lograr un desempeño adecuado según el problema a abordar. La tarea de ajuste no es trivial, incluso para los expertos (Eiben et al., 2007). De acuerdo a Smit et al. (2009) una perspectiva básica del problema se puede resumir como un diseño de 3 capas, presentado en la Figura 2.

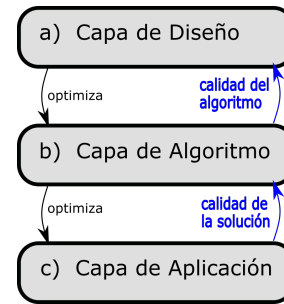


Figura 2: Esquema de concepto general de ajuste de hiper-parámetros. Adaptado de Smit et al. (2009).

En la capa de diseño, un meta-optimizador (optimizador de un optimizador) busca la mejor configuración para el algoritmo. No requiere mayor información acerca de la aplicación (problema o problemas bajo tratamiento). Únicamente requiere conocer los dominios de los hiper-parámetros a ajustar (espacio de búsqueda) y el criterio de finalización. Este proceso genera una alta demanda de recursos computacionales, debido a que se ejecuta el algoritmo para cada configuración candidata generada por el meta-optimizador.

El problema de ajuste de hiper-parámetros numéricos o categóricos, puede afrontar distintas dificultades (Veček et al., 2016). Entre estas dificultades se puede mencionar: requiere altas cantidades de tiempo; los mejores hiper-parámetros para el algoritmo varían según el problema de optimización; existen interacciones entre los parámetros; y la aleatoriedad propia de los algoritmos meta-heurísticos puede ocasionar distintos resultados incluso cuando se usan los mismos valores de hiper-parámetros.

#### 2.5. Clasificación de los Métodos de Ajuste

Según Eiben et al. (2007) el ajuste de hiper-parámetros en algoritmos evolutivos se puede abordar desde dos enfoques distintos: sintonización y control.

1. **Sintonización.** Es un enfoque fuera de línea, en el cual se busca encontrar valores óptimos para los hiper-parámetros. Posteriormente se ejecuta el algoritmo con estos valores, sin efectuar ninguna modificación.
2. **Control.** Es un enfoque en línea. El algoritmo se inicializa con valores base en los hiper-parámetros, que se auto-ajustan durante la ejecución del algoritmo. El proceso de auto-ajuste puede basarse en métodos determinísticos, adaptativos o auto-adaptativos, los cuales se diferencian por las formas en las que se modifican los parámetros.

En nuestro estudio usamos el enfoque de sintonización, que permite efectuar búsquedas exhaustivas de las combinaciones de hiper-parámetros disponibles dentro del espacio de búsqueda y así determinar las más adecuadas para la tarea en cuestión.

### 3. Estudio experimental

En esta sección se presentan las definiciones correspondientes al estudio experimental propuesto.

### 3.1. Problema de optimización

Como modelos bajo estudio, se plantea un miembro representativo de cada una de las nueve familias de procesos de control propuestas como *test batch* en Åström and Häggglund (2004). Estos procesos se presentan en la Tabla 1. Según sus autores estas familias consisten en una muestra representativa de la mayoría de procesos encontrados en la teoría de control.

Cada uno de los modelos ha sido implementado en el entorno de MATLAB R2019a, en un ordenador portátil con procesador *Intel Core i7* de 9ª generación y 8 GB de memoria RAM. En dicho lazo de control, se establece un controlador  $C(s)$  que responde a la ecuación (1) para un controlador PID con ganancias  $K_p, K_i, K_d$  (variables de decisión  $\theta$ ). En conformidad con lo planteado en el trabajo de Åström and Häggglund (2004), la señal de entrada será una señal de tipo escalón unitario de amplitud 1. Sobre estos modelos y con las condiciones establecidas, se determina para cada proceso un conjunto de ganancias de referencia  $\theta_R$  por medio de la aplicación de las técnicas de Ziegler and Nichols (1993),

Tabla 1: Valores de coeficientes según familia seleccionados para estudio.

Familia	Modelo
$P_1(s)$	$f_1(s) = \frac{e^{-s}}{1+s}$
$P_2(s)$	$f_2(s) = \frac{e^{-s}}{(1+1.3s)^2}$
$P_3(s)$	$f_3(s) = \frac{1}{(s+1)(1+0.1s)^2}$
$P_4(s)$	$f_4(s) = \frac{1}{(s+1)^5}$
$P_5(s)$	$f_5(s) = \frac{1}{(1+s)(1+0.5s)(1+0.25s)(1+0.125s)}$
$P_6(s)$	$f_6(s) = \frac{e^{-0.3s}}{s(1+0.7s)}$
$P_7(s)$	$f_7(s) = \frac{2e^{-0.9s}}{(1+2s)(1+0.1s)}$
$P_8(s)$	$f_8(s) = \frac{1-0.6s}{(s+1)^3}$
$P_9(s)$	$f_9(s) = \frac{1}{(s+1)(0.16s^2+0.56s+1)}$

Los objetivos de diseño seleccionados para el presente trabajo de investigación son:

- **La integral del error absoluto (IAE).**- Es una medida de desempeño que nos indica cuán alejada se encuentra la respuesta del sistema frente a la entrada de referencia .

$$J_{IAE}(\theta) = \int_{t=t_0}^{t_f} |e(t)| dt \quad (7)$$

- **La medida de la función máxima de sensibilidad ( $M_s$ ).**- Es una medida de robustez que nos da información sobre el peor escenario en caso de presentarse una perturbación.

$$J_{M_s}(\theta) = \max_{\omega} |S(i\omega)| = \max_{\omega} \left| \frac{1}{1 + P(i\omega)C(i\omega)} \right| \quad (8)$$

El primero de ellos será aproximado discretamente via simulación en MATLAB con tasa de muestreo en base al tiempo de establecimiento (*settling time*) del modelo bajo estudio. El espacio de búsqueda para las variables de decisión será definido a partir del cubo unitario, como fue propuesto por Reynoso-Meza et al. (2014c) con base en el trabajo de Tan et al. (2006).

Lo anterior garantiza el muestreo de controladores estables durante todo el proceso evolutivo. Por consiguiente, el MOP planteado se puede expresar como:

$$\min_{\theta} J(\theta) = [J_{IAE}(\theta), J_{M_s}(\theta)] \quad (9)$$

en donde:

$$\theta = [\tilde{K}_p, \tilde{K}_i, \tilde{K}_d] \quad (10)$$

sujeto a:

$$\begin{aligned} \tilde{K}_p &\in [0, 1] \\ \tilde{K}_i &\in [0, 1] \\ \tilde{K}_d &\in [0, 1] \\ 1.2 &\leq M_s \leq 2 \end{aligned} \quad (11)$$

Donde  $\tilde{K}_p, \tilde{K}_i, \tilde{K}_d$  representan el equivalente en base al cubo unitario, de los parámetros  $K_p, K_i, K_d$  según Reynoso-Meza et al. (2014a), Tan et al. (2006).

Para propósitos de interpretabilidad, los objetivos de diseño se normalizan con respecto al conjunto de ganancias de referencia  $\theta_R$ , para lo cual se introducen las ecuaciones 12 y 13.

$$\hat{J}_{IAE}(\theta) = \frac{IAE(\theta)}{IAE(\theta_R)} \quad (12)$$

$$\hat{J}_{M_s}(\theta) = \frac{M_s(\theta)}{M_s(\theta_R)} \quad (13)$$

### 3.2. Indicadores de desempeño de los algoritmos utilizados

Tal como se menciona en el trabajo de Eiben et al. (2007), dentro del proceso de calibración de los hiper-parámetros se necesita conocer el desempeño de los optimizadores para poder establecer una medida de su utilidad. En el presente trabajo, se aborda el problema de optimización desde dos enfoques distintos para la medición del desempeño, el primero de ellos mediante la medida del Hipervolumen del PF; mientras que en el segundo se genera un enfoque de función objetivo agregada (AOF, por sus siglas en inglés *Aggregate Objective Function*) de los objetivos bajo estudio. Para esto último se combinan los índices de desempeño  $\hat{J}_{IAE}(\theta)$  y  $\hat{J}_{M_s}(\theta)$  en un índice de utilidad  $J_{GPP}(\phi)$  mediante técnicas de *Global Physical Programming*, planteadas por Sanchis et al. (2010) y que nos permiten expresar como un “valor físico” las preferencias del diseñador con rangos que van desde “altamente deseable” hasta “altamente indeseable”.

Estas técnicas se basan en el trabajo de Messac (1996) y responden a una regla denominada “uno contra otros” (OVO, por sus siglas en inglés *One vs Others*). Esta regla es una forma de preferencia entre criterios en la cual se prefiere un objetivo con respecto a los demás si se encuentra un grado de preferibilidad mejor; de modo que se procura realizar la minimización del peor objetivo en base a la mejor minimización (Messac, 1996). Por ejemplo: se puede asumir cierta degradación al pasar de “deseable” a “tolerable” en  $J_2(\theta)$ , siempre que ello signifique una mejora de una región de “indeseable” a “tolerable” de  $J_1(\theta)$ ; con la finalidad de alcanzar un mejor desempeño. (Messac, 1996; Reynoso-Meza et al., 2014b). Siguiendo las recomendaciones de Reynoso-Meza et al. (2016) se estructura la Tabla 2 que presenta las preferencias establecidas para nuestro estudio.

Tabla 2: Preferencias para índices de desempeño en tarea de ajuste de controladores. Se ha establecido cinco rangos: Altamente Deseable (AD), Deseable (D), Tolerable (T), Indeseable (I), Altamente Indeseable (AI).

	Conjunto de Preferencias														
	←	AD	→	←	D	→	←	T	→	←	I	→	←	AI	→
Objetivo	$\hat{J}_i^0$		$\hat{J}_i^1$		$\hat{J}_i^2$		$\hat{J}_i^3$		$\hat{J}_i^4$		$\hat{J}_i^5$				
$\hat{J}_{IAE}(\theta)$	0.2		0.5		0.7		1		2		10				
$\hat{J}_{M_s}(\theta)$	0.7143		0.8095		0.9048		1.0000		1.0952		1.1905				

De esta manera, determinamos como índices de utilidad:

- Para el enfoque basado en hipervolumen, el valor del mismo.
- Para el enfoque basado en AOF, el valor de  $J_{GPP}(\phi)$ .

### 3.3. Algoritmo REVAC para el ajuste de hiper-parámetros

Luego de la revisión de la literatura, dentro de los distintos algoritmos disponibles para tareas de ajuste de hiper-parámetros se seleccionó REVAC para el proceso de ajuste. REVAC según Eiben et al. (2007) es un algoritmo de estimación de distribución (EDA por sus siglas en inglés), que realiza la calibración de forma sistemática y automatizada para medir la importancia de los parámetros. Este algoritmo parte de una población de configuraciones y durante su ejecución las mejora constantemente (Montero et al., 2014). Además, según Eiben et al. (2007) en la formulación de este algoritmo no encontraron diferencias significativas al efectuar replicaciones del mismo. Siendo REVAC un algoritmo en sí mismo, posee sus propios parámetros de ejecución, los cuales han sido tomados por defecto según recomiendan sus autores y se enlistan a continuación:

- **Número máximo de ejecuciones ( $maxEjec$ ).**- Con un valor de 1000.
- **Tamaño de la población  $M$ .**- Indica el número de vectores solución generados de forma inicial, con un valor de 100.
- **Tamaño del operador de cruzamiento ( $N$ ).**- Indica el número de vectores seleccionados para el proceso de cruzamiento. Toma un valor de  $M/2$ .
- **Tamaño del operador de mutación ( $H$ ).**- Toma un valor de  $N/10$ .

La versión utilizada de REVAC ha sido adecuada para efectuar la calibración de los parámetros de variantes de la familia DE. En el Algoritmo 2, se presenta el pseudocódigo correspondiente:

```

1 Generar un conjunto  $C_p$  compuesto por  $M$ 
  configuraciones de parámetros al azar
2 Evaluar cada una de las configuraciones  $M \in C_p$ 
  Inicializar contador de ejecuciones  $ejec = 0$ 
3 mientras  $ejec < maxEjec - M$  hacer
4   Generar una configuración hijo  $c_{hijo}$  por medio de
    cruzamiento uniforme sobre las  $N$  mejores
    configuraciones de parámetros según su utilidad.
5   para cada parámetro  $p_i \in c_{hijo}$  hacer
6     Asignar a  $p_i$  un valor aleatorio comprendido
      entre el  $H$ -ésimo peor y  $H$ -ésimo mejor vecino
      de  $p_i$ .
7   Evaluar la configuración  $c_{hijo}$ .
8   Reemplazar la configuración más antigua  $\in C_p$  con
     $c_{hijo}$ .
9   Calcular la entropía de cada parámetro  $C_p$ .

```

Algoritmo 2: REVAC

En el contexto del Algoritmo 2, el término entropía en la línea 9 hace referencia a la cantidad mínima de información para especificar que un valor pertenece a una determinada distribución según mencionan los autores de REVAC en Nannen and Eiben (2007). Este algoritmo se basa en el cálculo de la entropía diferencial de Shannon ( $H$ ), de una distribución  $\mathcal{D}$  definida sobre un intervalo  $[a, b]$ ; misma que se expresa como:

$$\mathcal{D}_{[a,b]} = \int_a^b \mathcal{D}(x) \log_2 \mathcal{D}(x) dx \quad (14)$$

Los algoritmos basados en DE analizados en el presente estudio son:

- Algoritmo DE básico, descrito en la sección 2.
- Algoritmo de evolución diferencial auto-adaptativa (SA-DE, por sus siglas en inglés) (Qin and Suganthan, 2005; Brest et al., 2006), mismo que incorpora una estrategia de aprendizaje para seleccionar de entre 2 alternativas de mutación; implementación propia.
- El algoritmo de Evolución Diferencial Multi Objetivo Extendida basado en poda esférica (spMODEx) propuesto por Reynoso-Meza et al. (2017), se basa en técnicas de programación física (PP por sus siglas en inglés *Physical Programming*) y poda esférica para generar soluciones diversas.
- El algoritmo de Evolución Diferencial Multi Objetivo basado en Indicador (ibMODE), permite incluir indicadores

<sup>2</sup><https://es.mathworks.com/matlabcentral/fileexchange/65145>

<sup>3</sup><https://es.mathworks.com/matlabcentral/fileexchange/87262>

de desempeño en los mecanismos de selección del algoritmo.

Para la realización del presente estudio se ha adecuado las versiones de spMODEx<sup>2</sup> e ibMODE basado en preferencias<sup>3</sup> disponibles en MATLAB Central para poder trabajar en conjunto con REVAC.

#### 4. Análisis y discusión de resultados

En esta sección comparamos los resultados obtenidos al evaluar los mejores conjuntos de hiper-parámetros sobre los algoritmos basados en DE bajo estudio. Antes de proceder a la evaluación efectuamos el proceso de calibración de los mismos, para lo cual se sigue el enfoque planteado en la Figura 2. Es decir, se propone el proceso de ajuste basado en el esquema planteado en la Tabla 3.

Tabla 3: Diseño de experimentos para ajuste de parámetros.

<b>Capa de diseño</b>	REVAC
<b>Capa de algoritmo</b>	DE/SADE/spMODEx/ibMODE
<b>Capa de aplicación</b>	Controlador PID en sistemas SISO

##### 4.1. Configuración de los experimentos

###### 4.1.1. Delimitación de hiper-parámetros a ajustar

Dentro de los hiper-parámetros categóricos se ha seguido distintas recomendaciones de la literatura, en las cuales se puede evidenciar que la estrategia de mutación con mayor aplicación es: “DE/rand/1” (Storn and Price, 1997; Qin and Suganthan, 2005; Bilal et al., 2020) siendo incluso considerada como estrategia principal para la formulación del algoritmo SADE. Sin embargo, en otros trabajos como el de Qin et al. (2009), se incluye también como estrategia principal a “DE/rand/2” dentro de la formulación de SADE. Por tal motivo, se toma estas dos estrategias bajo estudio. Por otra parte las alternativas para la estrategia de recombinación serán: binomial y lineal. Las combinaciones de las distintas estrategias organizadas como variantes se han establecido tomando en cuenta las recomendaciones de Červenka and Boudna (2018), tal como se presentan en la Tabla 4.

Tabla 4: Tratamientos según hiper-parámetros categóricos.

Denominación	Est. Mutación	Est. Recombinación
Bin1	DE/rand/1	Binomial
Bin2	DE/rand/2	Binomial
Lin1	DE/rand/1	Lineal
Lin2	DE/rand/2	Lineal

Por otra parte, como hiper-parámetros numéricos se realizó la calibración de: tamaño de la población ( $N_p$ ), factor de escala ( $F$ ) y tasa de cruce ( $C_r$ ). Para establecer la delimitación de los rangos de los espacios de búsqueda correspondientes, se ha considerado:

- **Para  $N_p$** , de acuerdo a las recomendaciones de Storn and Price (1997), se sugiere  $N_p = 10 \cdot d$  ( $d$  es la dimensionalidad del espacio de búsqueda). Para un controlador tipo

PID,  $d = 3$ . Por propósito de generar soluciones más diversas, se establece un rango de  $10 \cdot d \leq N_p \leq 100$

- **Para  $F$** , de acuerdo a lo establecido en Storn and Price (1997) se recomienda un rango de  $0.5 \leq F \leq 1$
- **Para  $C_r$** , según los estudios de Storn and Price (1997) se recomienda un rango de  $0.5 \leq C_r \leq 1$

De este modo, se realizó una corrida de REVAC por cada uno de los algoritmos, siguiendo las combinaciones de estrategias presentadas en la Tabla 4 sobre cada una de las plantas de forma individual.

##### 4.1.2. Configuración de escenarios para determinar diferencias estadísticas

Luego de determinar los mejores hiper-parámetros para las distintas alternativas de algoritmos basados en DE bajo estudio, se procedió a ejecutar 25 pruebas con cada una de ellas. Cada prueba consiste en una ejecución de la variante del algoritmo en cuestión configurada con sus correspondientes hiper-parámetros realizando 2000 evaluaciones de la función objetivo por cada prueba. El criterio de finalización de los algoritmos se establece por número de evaluaciones de la función objetivo.

##### 4.2. Experimento I: Análisis de influencia de los hiper-parámetros en el desempeño de algoritmos DE bajo enfoque basado en AOF

Este experimento considera las variantes de algoritmo DE bajo enfoque basado en AOF seleccionadas para estudio. Como punto de partida, se efectúa el proceso de calibración de hiper-parámetros mediante la aplicación de REVAC. Trabajamos bajo un enfoque de calibración de forma puntual a cada uno de los problemas. Posteriormente se determina los mejores valores resultantes de cada una de estas tareas de calibración. En este punto, es relevante mencionar que dado que el algoritmo SADE trabaja únicamente con recombinación binomial, no se incorpora para su análisis la recombinación lineal.

Una vez encontradas las mejores configuraciones, efectuamos pruebas según los lineamientos establecidos en la subsección 4.1.2 A modo de resumen del presente experimento, se condensa la información más relevante en la Figura 3 por medio de un diagrama de diferencias críticas. Este tipo de diagramas se basa en la determinación de la diferencia crítica y su representación según establece Demšar (2006). Mediante la aplicación del test de Nemenyi, se establece la existencia de diferencias significativas entre dos o más algoritmos, de la siguiente manera: cualquier par de algoritmos cuya diferencia en cuanto a desempeño se establece como significativamente diferente (Calvo and Santafé, 2016).

De forma gráfica, la inexistencia de diferencias estadísticamente significativas se representa por medio de la unión de los de algoritmos bajo estudio mediante una barra horizontal. Es decir, todos aquellos algoritmos en los que no se encuentren diferencias superiores a la diferencia crítica se unirán por medio de esta barra. Por otra parte, aquel algoritmo con el mejor desempeño de entre todo el conjunto se ubica primero desde la izquierda del diagrama.

Por ejemplo, en la Subfigura 3(g) el diagrama de diferencias críticas está identificando un ranking de acuerdo al desempeño de las seis estrategias bajo análisis, e identifica que en un

primer sitio tendría que estar “DE\_Lin2” y en un último sitio “SADE\_Bin1”. Sin embargo la barra horizontal que une a las soluciones “DE\_Lin2” y “DE\_Lin1” nos indica que no se tiene evidencia estadística suficiente para sustentar que sus desempeños son diferentes. Por otra parte, dado que “DE\_Lin2” con respecto a “DE\_Bin2” no se encuentran unidas por una barra, quiere decir que hay suficiente evidencia estadística para justificar que el desempeño de “DE\_Lin2” es mejor que el de las otras 4 estrategias.

De este modo, según se observa en la Figura 3 la variante de algoritmo DE básico tiene una predominancia marcada, presentando un mejor desempeño en el 88.9 % de los procesos estudiados, en contraparte la variante SADE tiene presencia en el 11.1 % restante. Con respecto a las estrategias de mutación, se observa que “DE/rand/2” se presenta en 55.55 % de los casos y “DE/rand/1” en el 44.45 % restante. Finalmente, en lo que respecta a estrategias de recombinación, lineal sobresale en el 55.55 % de los procesos bajo análisis mientras que la binomial se presenta en el 44.4 % restante.

Por medio del análisis del diagrama de diferencias críticas se determinan las mejores configuraciones para cada uno de los problemas, tal como se muestra en la Tabla 5. Es relevante mencionar que, en base a los resultados obtenidos se presenta para  $f_2(s)$  como recomendado el algoritmo SADE, mismo que en su configuración no contempla los parámetros  $F$  y  $C_r$ .

Acerca de los resultados en cuanto a  $N_p$  se observa que existe una tendencia hacia el límite inferior del intervalo planteado dentro del espacio de búsqueda, lo cual nos hace creer que apuntan hacia lo que plantean las recomendaciones de (Storn and Price, 1997).

#### 4.3. Experimento II: Análisis de influencia de los hiperparámetros en el desempeño de algoritmos DE bajo enfoque basado en hipervolumen

En el experimento II se presenta el análisis de las variantes de algoritmo DE bajo enfoque basado en hipervolumen seleccionadas para estudio. De igual manera que en el experimento anterior, realizamos inicialmente el proceso de ajuste de parámetros mediante la aplicación de REVAC. Los algoritmos bajo estudio han sido adecuados para incluir las consideraciones planteadas en la Tabla 4 y Subsección 4.1.2.

Siguiendo el mismo enfoque que en el estudio para el enfoque basado en AOF, efectuamos las pruebas requeridas para determinar la existencia de diferencias estadísticamente significativas. En la Figura 4 se presenta el diagrama de diferencias críticas como resumen de los experimentos desarrollados para evaluar el desempeño de los algoritmos al efectuar las pruebas bajo enfoque basado en hipervolumen.

Tal como se observa en la Figura 4 “ibMODE” presenta una clara predominancia sobre la variante “spMODEx”, en el 100 % de los casos estudiados. En lo que respecta a las estrategias de mutación, se puede notar que en el 88.9 % de los procesos estudiados obtuvo mejor desempeño la estrategia “DE/rand/1”, completando el 11.1 % restante la estrategia “DE/rand/2”. Para concluir, en cuanto a estrategias de recombinación, la Lineal sobresale en el 55.55 % de los procesos bajo análisis mientras que la Binomial se presenta en el 44.4 % restante; teniendo en este último aspecto un comportamiento similar al obtenido en el

análisis bajo enfoque basado en hipervolumen. Las recomendaciones de las mejores configuraciones se presentan en la Tabla 5.

Acerca de los resultados en cuanto a  $N_p$  se observa que existe una tendencia hacia el límite superior del intervalo planteado dentro del espacio de búsqueda, lo cual nos hace creer que dichos tamaños de población pueden generar soluciones con mayor diversidad y pertinencia dentro del PF. Esto hace sentido con el hecho de que en este tipo de análisis se prefieren soluciones más diversas dentro de los conjuntos.

En la figura 5 se ilustra el comportamiento de distintos controladores calculados para la planta  $f_7(s)$ , respecto a una señal de entrada del tipo escalón unitario. Las constantes  $K_p$ ,  $K_i$ ,  $K_d$  correspondientes han sido determinadas por medio de las recomendaciones establecidas en la tabla 5 para los enfoques abordados en este estudio. Por medio de la evolución de la salida  $y(t)$  con respecto al tiempo, en función de los distintos controladores, se evidencia el desempeño de los mismos.

#### 4.4. Tiempos de cómputo empleados

A la complejidad computacional del REVAC, se añade la complejidad de la versión de DE bajo estudio, así como la complejidad derivada del modelo bajo estudio. Esta última, puede incrementar la cantidad de recursos computacionales necesarios para el proceso de calibración. Es así que en cuanto a los tiempos de cómputo empleados en base a nuestros recursos descritos anteriormente; se cuentan con registros de procesos que requirieron desde 12 hasta cerca de 36 horas por cada ejecución de este proceso.

## 5. Conclusiones

En el presente estudio hemos comparado las capacidades de cuatro variantes de algoritmos de la familia DE presentes en la literatura al efectuar la tarea de ajuste de controladores PID en sistemas SISO; tanto desde un enfoque basado en AOF como en un enfoque basado en el valor del hipervolumen del PF generado. Posteriormente, hemos desarrollado un análisis de las mejores variantes y conjuntos de hiperparámetros con la finalidad de presentar una herramienta que posibilite un mejor aprovechamiento de las capacidades de estos algoritmos al resolver este problema de ingeniería.

#### 5.1. Acerca de las variantes de algoritmo DE bajo estudio con mejor rendimiento

Bajo orientación basada en AOF, el algoritmo DE básico predomina en alrededor del 88.9 % de los procesos bajo estudio. Por otra parte, cuando se emplea un análisis basado en hipervolumen se encuentra una clara dominancia del algoritmo ib-MODE, ya que presenta un mejor desempeño en todos los procesos bajo estudio.

#### 5.2. Acerca de estrategias con mejor rendimiento

En ambos tipos de análisis realizados se observa una fuerte presencia de la estrategia de recombinación lineal. En cuanto a estrategias de mutación, para las versiones basadas en AOF se presenta con mayor frecuencia la estrategia “DE/rand/2” y en versiones multi-objetivo se presenta la estrategia “DE/rand/1” en un 88.9 % de los casos estudiados.



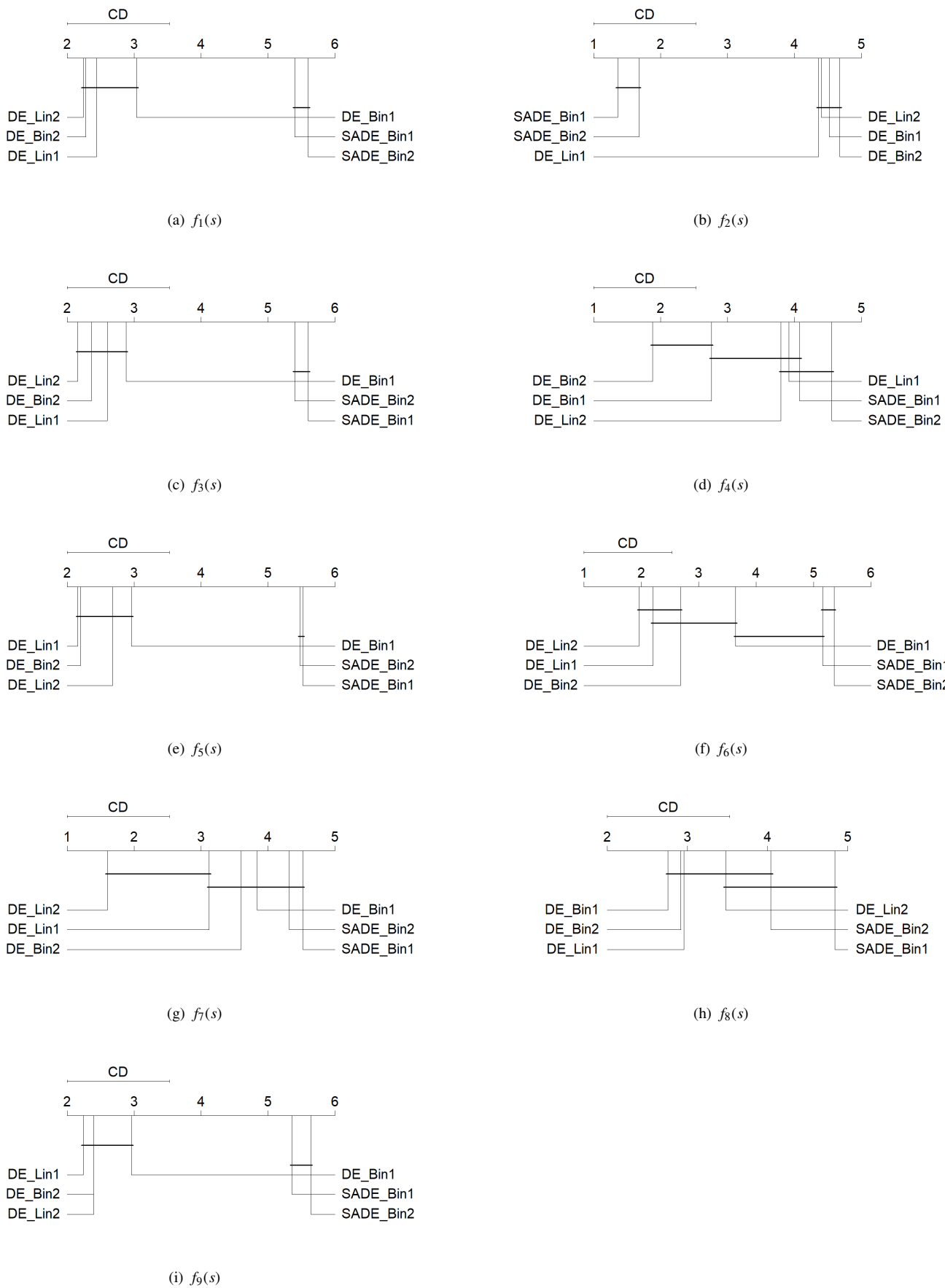
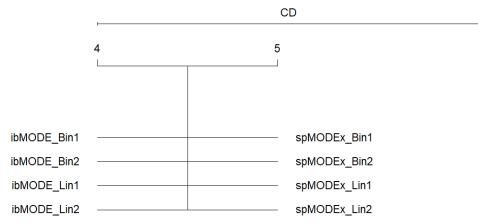
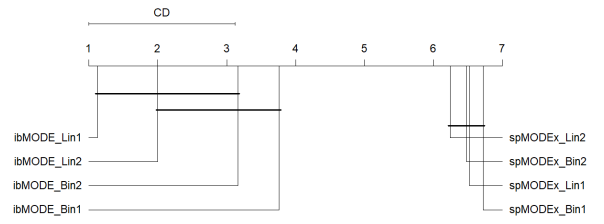


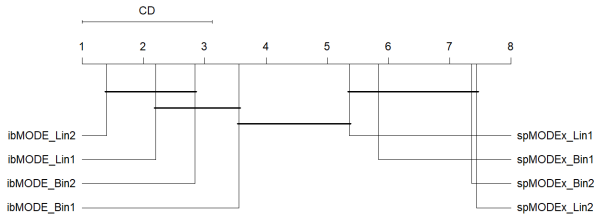
Figura 3: Diagrama de Diferencias Críticas para enfoque basado en AOF.



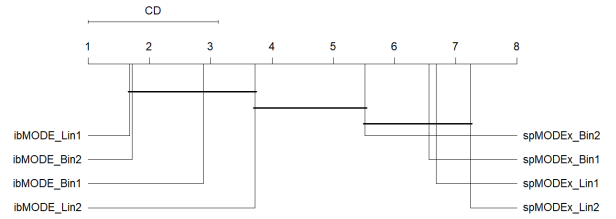
(a)  $f_1(s)$



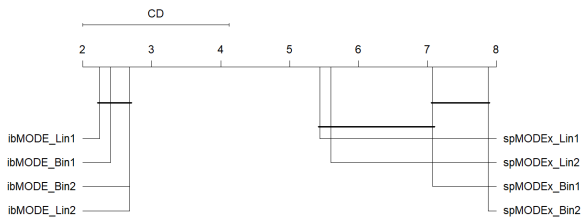
(b)  $f_2(s)$



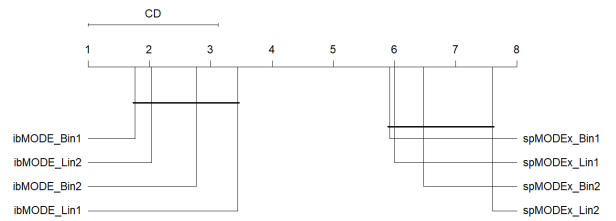
(c)  $f_3(s)$



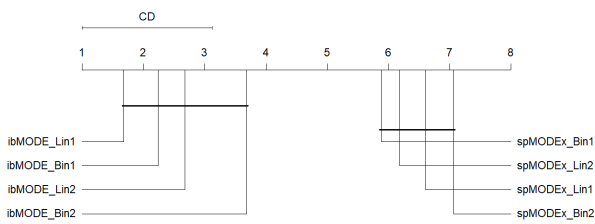
(d)  $f_4(s)$



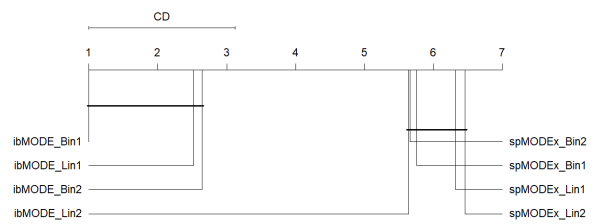
(e)  $f_5(s)$



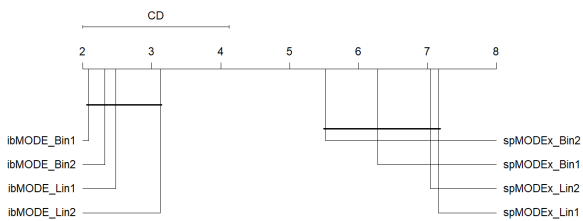
(f)  $f_6(s)$



(g)  $f_7(s)$



(h)  $f_8(s)$

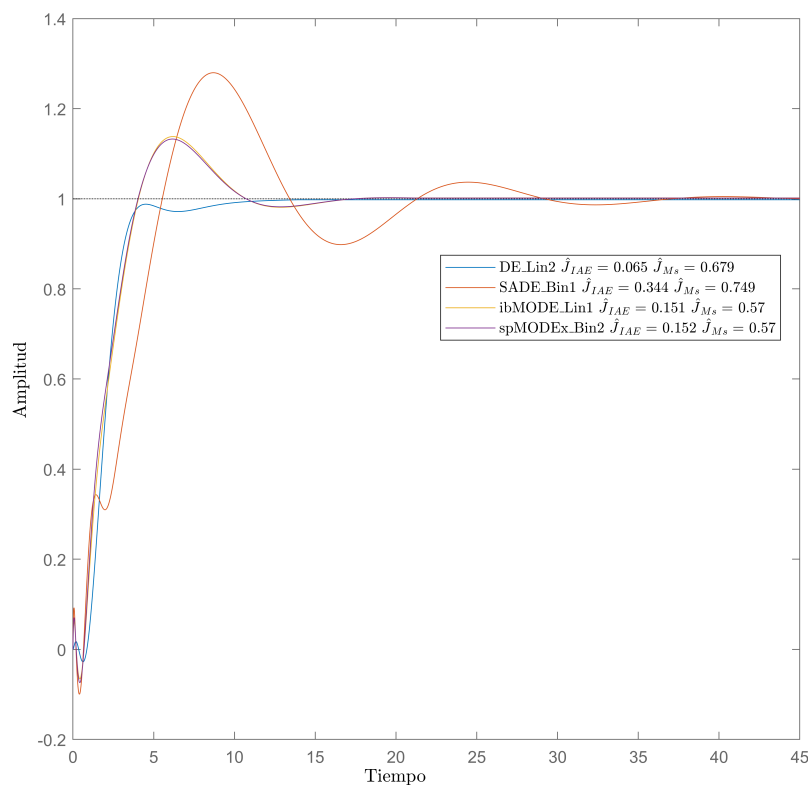


(i)  $f_9(s)$

Figura 4: Diagrama de Diferencias Críticas para enfoque basado en Hipervolumen.

Tabla 5: Parámetros recomendados para ajuste de controladores PID bajo enfoque basado en AOF (izq.) y basado en Hipervolumen (der.)

Proceso	Alg.	Est.	Hiper-parámetros			Alg.	Est.	Hiper-parámetros		
			$N_p$	$F$	$C_r$			$N_p$	$F$	$C_r$
$f_1(s)$	DE	Lin2	21	$50.03 \times 10^{-2}$	$51.75 \times 10^{-2}$	ibMODE	Bin1	39	$66.32 \times 10^{-2}$	$77.71 \times 10^{-2}$
$f_2(s)$	SADE	Bin1	43	-	-	ibMODE	Lin1	48	$89.97 \times 10^{-2}$	$82.74 \times 10^{-2}$
$f_3(s)$	DE	Lin2	21	$52.68 \times 10^{-2}$	$90.98 \times 10^{-2}$	ibMODE	Lin2	74	$73.87 \times 10^{-2}$	$75.67 \times 10^{-2}$
$f_4(s)$	DE	Bin2	22	$53.16 \times 10^{-2}$	$99.44 \times 10^{-2}$	ibMODE	Lin1	38	$56.23 \times 10^{-2}$	$96.32 \times 10^{-2}$
$f_5(s)$	DE	Lin1	21	$79.27 \times 10^{-2}$	$55.10 \times 10^{-2}$	ibMODE	Lin1	91	$80.32 \times 10^{-2}$	$82.28 \times 10^{-2}$
$f_6(s)$	DE	Lin2	21	$56.57 \times 10^{-2}$	$96.43 \times 10^{-2}$	ibMODE	Bin1	99	$98.26 \times 10^{-2}$	$75.23 \times 10^{-2}$
$f_7(s)$	DE	Lin2	74	$57.44 \times 10^{-2}$	$72.98 \times 10^{-2}$	ibMODE	Lin1	64	$63.23 \times 10^{-2}$	$88.40 \times 10^{-2}$
$f_8(s)$	DE	Bin1	21	$82.93 \times 10^{-2}$	$94.33 \times 10^{-2}$	ibMODE	Bin1	95	$97.51 \times 10^{-2}$	$51.12 \times 10^{-2}$
$f_9(s)$	DE	Lin1	21	$70.17 \times 10^{-2}$	$50.09 \times 10^{-2}$	ibMODE	Bin1	94	$90.27 \times 10^{-2}$	$59.38 \times 10^{-2}$

Figura 5: Respuesta del  $f_7(s)$  frente a controladores calculados en base a algoritmos DE

### 5.3. Alcance del estudio y trabajos futuros

Considerando los aspectos previamente mencionados, se presenta una herramienta de gran utilidad al configurar las variantes de algoritmo DE bajo estudio para de ajuste de controladores PID en procesos SISO, sobretodo para los nuevos desarrolladores de control. Los procesos a emplear pueden ser aquellos pertenecientes a las familias bajo estudio; o procesos que pudiesen ser aproximables a una de dichas familias. Es relevante mencionar que; si bien los procesos estudiados generan lazos de control relativamente simples, puede ser aún más interesante identificar lazos más complejos o lazos multivariables; lo cual se encuentra dentro de nuestra línea de trabajos futuros.

### Agradecimientos

Trabajo financiado parcialmente por el *Conselho Nacional de Desenvolvimento Científico e Tecnológico* (CNPq), y la *Fun-*

*dação Araucária* (FAPPR) - Brasil - proyectos 310079/2019-5-PQ2, 4408164/2021-2-Univ y PRONEX-51432/2018-PPP.

### Referencias

- Alhanjouri, M., 2017. Modern Optimization Techniques for PID Parameters of Electrohydraulic Servo Control System. *Int. J. Recent Innov. Trends Comput. Commun.* 5 (March), 71–79.
- Ang, K. H., Chong, G., Li, Y., 2005. PID control system analysis, design, and technology. *IEEE Transactions on Control Systems Technology* 13 (4), 559–576.
- Åström, K. J., Hägglund, T., 1995. *PID controllers: theory, design, and tuning*. Vol. 2. ISA.
- Åström, K. J., Hägglund, T., 2004. Revisiting the Ziegler-Nichols step response method for PID control. *J. Process Control* 14 (6), 635–650.
- Bilal, Pant, M., Zaheer, H., Garcia-Hernandez, L., Abraham, A., 2020. Differential Evolution: A review of more than two decades of research. *Eng. Appl. Artif. Intell.* 90 (October 2019).

- Borase, R., Maghade, D., Sondkar, S., Pawar, S., 06 2021. A review of pid control, tuning methods and applications. *International Journal of Dynamics and Control* 9.
- Brest, J., Žumer, V., Maučec, M. S., 2006. Self-adaptive differential evolution algorithm in constrained real-parameter optimization. 2006 IEEE Congr. Evol. Comput. CEC 2006, 215–222.
- Calvo, B., Santafé, G., 2016. scamp: Statistical comparison of multiple algorithms in multiple problems. *R Journal* 8 (1), 248–256. DOI: 10.32614/rj-2016-017
- Chiha, I., Ghabi, J., Liouane, N., 05 2012. Tuning pid controller with multi-objective differential evolution. 5th International Symposium on Communications Control and Signal Processing, ISCCSP 2012.
- Coello, C. A. C., Lamont, G. B., Veldhuizen, D. A. V., Goldberg, D. E., Koza, J. R., 2007. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Springer.
- Das, S., Suganthan, P. N., 2011. Differential evolution: A survey of the state-of-the-art. *IEEE Trans. Evol. Comput.* 15 (1), 4–31.
- Dashti, M., Shojaei, K., Seyedkashi, S. M., Behnam, M., 2010. Tuning of digital PID controller using particle swarm optimization. *Proc. 29th Chinese Control Conf. CCC'10*, 3383–3389.
- De Landgraaf, W. A., Eiben, A. E., Nannen, V., 2007. Parameter calibration using meta-algorithms. 2007 IEEE Congr. Evol. Comput. CEC 2007, 71–78.
- Demšar, J., 2006. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research* 7, 1–30.
- Eiben, A. E., Michalewicz, Z., Schoenauer, M., Smith, J. E., 2007. Parameter control in evolutionary algorithms. *Stud. Comput. Intell.* 54 (2), 19–46.
- Fister, D., Fister, I., Fister, I., Šafarič, R., 2016. Parameter tuning of PID controller with reactive nature-inspired algorithms. *Rob. Auton. Syst.* 84, 64–75.
- Jaen-Cuellar, A. Y., Romero-Troncoso, R. D. J., Morales-Velazquez, L., Osornio-Rios, R. A., 2013. PID-controller tuning optimization with genetic algorithms in servo systems. *Int. J. Adv. Robot. Syst.* 10.
- Konstantinov, S. V., Baryshnikov, A. A., 2017. Comparative Analysis of Evolutionary Algorithms for the Problem of Parametric Optimization of PID Controllers. *Procedia Comput. Sci.* 103 (October 2016), 100–107.
- Kozák, Š., 2014. State-of-the-art in control engineering. *J. Electr. Syst. Inf. Technol.* 1 (1), 1–9.
- Lakshmi, K. V., Srinivas, P., Harshad, S., 2019. Differential Evolution Based PID Controller For Three Tank Level Process. *International Journal of Engineering and advanced technology (IJEAT)* 8 (4), 1274–1278.
- Messac, A., 1996. Physical programming: Effective optimization for computational design. *AIAA Journal* 34 (1), 149–158. DOI: 10.2514/3.13035
- Miettinen, K., 1998. *Nonlinear Multiobjective Optimization*. Springer Science & Business Media.
- Montero, E., Riff, M. C., Neveu, B., 2014. A beginner's guide to tuning methods. *Appl. Soft Comput.* 17, 39–51.
- Nannen, V., Eiben, A. E., 2007. Relevance estimation and value calibration of evolutionary algorithm parameters. *IJCAI International Joint Conference on Artificial Intelligence*, 975–980.
- Neumüller, C., Wagner, S., Kronberger, G., Affenzeller, M., 2012. Parameter meta-optimization of metaheuristic optimization algorithms. *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)* 6927 LNCS (PART 1), 367–374.
- Price, K. V., Storn, R. M., Lampinen, J. A., 2005. *Differential evolution: a practical approach to global optimization*. Natural computing series. Springer, Berlin ; New York.
- Qin, A. K., Huang, V. L., Suganthan, P. N., 2009. Differential evolution algorithm with strategy adaptation for global numerical optimization. *IEEE Transactions on Evolutionary Computation* 13 (2), 398–417. DOI: 10.1109/TEVC.2008.927706
- Qin, A. K., Suganthan, P. N., 2005. Self-adaptive differential evolution algorithm for numerical optimization. 2005 IEEE Congr. Evol. Comput. IEEE CEC 2005. Proc. 2, 1785–1791.
- Reynoso-Meza, G., Blasco, X., Sanchis, J., Martínez, M., 2014a. Controller tuning using evolutionary multi-objective optimisation: Current trends and applications. *Control Eng. Pract.* 28 (1), 58–73.
- Reynoso-Meza, G., Ferragud, X. B., Saez, J. S., Durá, J. M. H., 2017. Controller Tuning with Evolutionary Multiobjective Optimization: A Holistic Multiobjective Optimization Design Procedure. Vol. 85. Springer.
- Reynoso-Meza, G., Sanchis, J., Blasco, X., García-Nieto, S., 2014b. Physical programming for preference driven evolutionary multi-objective optimization. *Applied Soft Computing Journal* 24, 341–362. DOI: 10.1016/j.asoc.2014.07.009
- Reynoso-Meza, G., Sanchis, J., Blasco, X., Herrero, J. M., 2012. Multiobjective evolutionary algorithms for multivariable PI controller design. *Expert Syst. Appl.* 39 (9), 7895–7907.
- Reynoso-Meza, G., Sanchis, J., Blasco, X., Herrero, J. M., 2014c. A stabilizing PID controller sampling procedure for stochastic optimizers. Vol. 19. IFAC.
- Reynoso-Meza, G., Sanchis, J., Blasco, X., Martínez, M., 2016. Preference driven multi-objective optimization design procedure for industrial controller tuning. *Inf. Sci. (Ny)*. 339, 108–131.
- Rodríguez-Molina, A., Mezura-Montes, E., Villarreal-Cervantes, M. G., Aldape-Pérez, M., 2020. Multi-objective meta-heuristic optimization in intelligent control: A survey on the controller tuning problem. *Appl. Soft Comput. J.* 93, 106342.
- Saad, M. S., Jamaluddin, H., Darus, I. Z., 2012. PID controller tuning using evolutionary algorithms. *WSEAS Trans. Syst. Control* 7 (4), 139–149.
- Sanchis, J., Martnez, M. A., Blasco, X., Reynoso-Meza, G., 2010. Modelling preferences in multi-objective engineering design. *Engineering Applications of Artificial Intelligence* 23 (8), 1255–1264.
- Singh, J., Singh, B., Joshi, N., 2017. Tuning Techniques of PID controller: A review. *Int. J. Emerg. Technol. (Special Issue NCETST 8 (1))*, 481–485.
- Smit, S. K., Eiben, A. E., 2010. Parameter tuning of evolutionary algorithms: Generalist vs. specialist. *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)* 6024 LNCS (PART 1), 542–551.
- Smit, S. K., Eiben, A. E., Amsterdam, V. U., 2009. Comparing parameter tuning methods for evolutionary algorithms. In: 2009 IEEE Congress on Evolutionary Computation. pp. 399–406.
- Storn, R., Price, K., 1997. Differential Evolution - A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. *J. Glob. Optim.* 11 (4), 341–359.
- Tan, N., Kaya, I., Yeroglu, C., Atherton, D. P., 2006. Computation of stabilizing PI and PID controllers using the stability boundary locus. *Energy Convers. Manag.* 47 (18-19), 3045–3058.
- Tanabe, R., Fukunaga, A., 2015. Tuning differential evolution for cheap, medium, and expensive computational budgets. 2015 IEEE Congr. Evol. Comput. CEC 2015 - Proc., 2018–2025.
- Ugolotti, R., Sani, L., Cagnoni, S., 2019. What can we learn from multi-objective meta-optimization of Evolutionary Algorithms in continuous domains? *Mathematics* 7 (3).
- Urrea-Quintero, J.-H., Hernández-Riveros, J.-A., Muñoz-Galeano, N., 2018. Optimum PI/PID Controllers Tuning via an Evolutionary Algorithm. *PID Control for Industrial Processes*.
- Červenka, M., Boudna, H., april 2018. Visual Guide on F and CR Parameters Influence on Differential Evolution Solution Quality. In: *Engineering Mechanics 2018*. pp. 234–238.
- Veček, N., Mernik, M., Filipič, B., Črepinšek, M., 2016. Parameter tuning with Chess Rating System (CRS-Tuning) for meta-heuristic algorithms. *Inf. Sci. (Ny)*. 372, 446–469.
- Wolpert, D. H., Macready, W. G., 1997. No free lunch theorems for optimization. *IEEE Trans. Evol. Comput.* 1 (1), 67–82.
- Ziegler, J. G., Nichols, N. B., 1993. Optimum settings for automatic controllers. *J. Dyn. Syst. Meas. Control. Trans. ASME* 115 (2B), 220–222.