Selección adaptativa de operadores en un algoritmo microevolutivo multiobjetivo con neuronas pulsantes

Carlos Benito-Jareño[0009−0003−9327−9206], Juan Carlos de la Torre[0000−0002−3662−6422], Patricia Ruiz[0000−0001−5891−0646] y Bernabé Dorronsoro[0000−0003−0481−790X]

Escuela Superior de Ingeniería, Universidad de Cádiz, España
{carlos.benito, juan.detorre, patricia.ruiz, bernabe.dorronsoro}@uca.es

Resumen. La selección adaptativa de operadores es fundamental para equilibrar la exploración y explotación en algoritmos metaheurísticos. Muchos enfoques actuales, tales como el del algoritmo μFAME, dependen de mecanismos como ventanas de memoria para estimar el rendimiento de los operadores, los cuales pueden penalizar la reactividad del algoritmo ante cambios en el paisaje de búsqueda. En este trabajo proponemos μAdEx, un algoritmo microevolutivo multiobjetivo que sustituye el controlador de inferencia difusa de μFAME por una red neuronal pulsante para implementar un mecanismo de selección de operadores reactivo y sin retardos. La validación experimental sobre la suite de problemas WFG demuestra que μAdEx consigue superar de forma estadísticamente significativa al algoritmo original en varias métricas de calidad de optimización multiobjetivo, logrando convergencias más rápidas gracias a su capacidad de discriminación inmediata de operadores ineficientes.

Palabras clave: Selección adaptativa de operadores · Algoritmos microevolutivos · Optimización multiobjetivo · Redes neuronales pulsantes · Modelo AdEx

1. Introducción

La optimización multiobjetivo mediante algoritmos evolutivos (EAs) requiere equilibrar continuamente la exploración y la explotación del espacio de búsqueda. Para ello, el paradigma de selección adaptativa de operadores (AOS) ajusta dinámicamente qué operador genético utilizar en cada momento en función de su rendimiento durante la ejecución.

Sin embargo, los métodos AOS tradicionales como el del EA μFAME [12] a menudo sufren problemas de latencia. Esta lentitud de respuesta se debe, por un lado, al uso de ventanas de memoria para promediar la alta varianza de las recompensas y, por otro, a la división rígida de la búsqueda en fases estáticas. Ambas estrategias generan inercia, lo que retrasa la adaptación del algoritmo ante cambios rápidos y hace que se desperdicien evaluaciones en soluciones subóptimas [11], algo especialmente crítico en problemas computacionalmente costosos.

Para solucionar este problema, este trabajo propone μAdEx, un EA que sustituye el controlador difuso original de μFAME por una arquitectura basada en una red neuronal pulsante (SNN) del modelo *Adaptive Exponential Integrate-and-Fire* (AdEx). La dinámica biofísica de estas neuronas modela el rendimiento de los operadores en tiempo real y traduce su éxito o fracaso directamente en corrientes eléctricas, absorbiendo el ruido de las evaluaciones para permitir una adaptación inmediata sin necesidad de ventanas de memoria ni periodos rígidos. Asimismo, el mecanismo integra una política termodinámica *softmax* que regula la entropía del sistema y la probabilidad de aplicación de cada operador mediante un único parámetro de temperatura. El sistema se compara empíricamente con el algoritmo original, usando la suite WFG y midiendo diversas métricas clave que cuantifican la calidad de la optimización multiobjetivo.

El resto de este artículo se organiza de la siguiente manera: la sección 2 revisa los trabajos relacionados. La sección 3 describe la formulación de la arquitectura propuesta. La sección 4 detalla la configuración de los experimentos. La sección 5 analiza los resultados de convergencia y la dinámica interna del controlador. Por último, la sección 6 resume las conclusiones del estudio.

2. Antecedentes y trabajos relacionados

Como ya se ha mencionado, la latencia es un desafío crítico en los métodos AOS, donde la alta varianza en el éxito de los operadores dificulta detectar su rendimiento real. Para mitigar este ruido, la literatura suele recurrir a comparaciones relativas, que introducen dependencias cruzadas y oscurecen la detección de cambios, o al uso de ventanas de memoria para promediar el éxito, lo cual reduce drásticamente la reactividad del algoritmo. Un claro exponente de esta última aproximación es el algoritmo μFAME, el cual emplea ventanas fijas de evaluaciones junto con un sistema de inferencia difuso para recalcular las probabilidades de selección.

A la hora de diseñar un nuevo controlador AOS basado en SNN resulta crucial elegir el modelo neuronal adecuado. Los modelos biofísicamente exactos como el de Hodgkin-Huxley pueden ser demasiado pesados computacionalmente para un bucle de optimización, mientras que los modelos de integración y disparo más simples (como el IF o el LIF) carecen de variables de adaptación, por lo que mantendrían frecuencias de disparo constantes que anularían la exploración [4]. El modelo AdEx, formulado por Brette y Gerstner [1] basándose en Izhikevich [6], ofrece el equilibrio perfecto: incorpora un mecanismo de adaptación de corriente que proporciona de forma natural la "fatiga" necesaria para penalizar a los operadores ineficientes, mientras que su conductancia de fuga hace que el voltaje decaiga suavemente, actuando como un filtro intrínseco contra el ruido de las recompensas. Esto elimina por completo la necesidad de contabilizar el estancamiento de forma artificial.

Finalmente, cabe destacar cómo se enmarca este trabajo dentro del estado del arte. Actualmente, cuando se busca mejorar los sistemas AOS con inteligencia artificial, la tendencia es recurrir a técnicas de aprendizaje profundo [14,7], las cuales también pueden sufrir de retardo debido a sus entrenamientos por

intervalos fijos. Por su parte, cuando se combinan EAs y SNNs, casi siempre se hace para optimizar la propia red neuronal (neuroevolución) [10,2]. Sin embargo, el enfoque inverso apenas ha sido explorado. El uso de la dinámica biofísica e instantánea de una SNN para controlar en tiempo real el comportamiento de un EA representa una vía novedosa y directa para solucionar las deficiencias estructurales de latencia y ruido en los sistemas AOS.

3. Arquitectura propuesta: μAdEx

Para superar la latencia inherente de los enfoques AOS basados en ventanas de memoria, proponemos una arquitectura híbrida que sustituye el controlador difuso de μFAME por una SNN. El objetivo es aprovechar la dinámica matemática intrínseca de las neuronas para gestionar el equilibrio entre exploración y explotación de forma reactiva y en tiempo real. Cada operador genético disponible en el EA está representado por una neurona independiente, cuyo potencial determina su probabilidad de aplicación en cada evaluación.

3.1. El modelo neuronal AdEx

Para simular el comportamiento de estas neuronas empleamos el modelo AdEx, de manera que cada neurona posee un potencial de membrana (V) y una corriente de adaptación (w) que se rigen por el siguiente sistema de ecuaciones diferenciales:

$$C\frac{dV}{dt} = -g_L(V - E_L) + g_L\Delta_T \exp\left(\frac{V - V_T}{\Delta_T}\right) - w + I_{ext} \ , \tag{1}$$

$$\tau_w\frac{dw}{dt} = a(V - E_L) - w \ , \tag{2}$$

donde, por un lado, C es la capacitancia de la membrana, g_L es la conductancia de fuga, E_L representa el potencial de reposo, V_T es el umbral de inicio exponencial, Δ_T es el factor de pendiente, e I_{ext} es la corriente externa inyectada como recompensa o castigo. Por el otro lado, la dinámica de la fatiga iónica está controlada por la constante de tiempo τ_w y el nivel de acoplamiento subumbral a. Cuando el voltaje alcanza el umbral de corte crítico ($V \geq V_{spike}$), la neurona emite un pulso y sus dos variables se reinician ($V \leftarrow V_r$; $w \leftarrow w + b$), simulando el pico de fatiga tras la descarga.

En este acercamiento inicial, la red propuesta utiliza una parametrización consistente con la descripción de neuronas corticales de disparo regular, basada en el modelo AdEx y en posteriores caracterizaciones de sus patrones de disparo [9,4]: $C = 200$ pF; $g_L = 10{,}0$ nS; $E_L = -70{,}0$ mV; $V_T = -50{,}0$ mV; $\Delta_T = 2{,}0$ mV; $\tau_w = 144{,}0$ ms; $a = 2{,}0$ nS; $V_{spike} = -40{,}0$ mV; $V_r = -58{,}0$ mV; $b = 40{,}0$ pA.

3.2. Dinámica de recompensas

La integración de las neuronas con el EA requiere un mapeo preciso entre el rendimiento del operador y la corriente externa (I_{ext}). Con el propósito de

mantener una mayor simplicidad en esta primera prueba de concepto se define un esquema de recompensa binario, de manera que si un operador genera una solución que mejora el hipervolumen del archivo de soluciones no dominadas, la neurona que controla su aplicación recibe una corriente de recompensa que la excita; en caso contrario, recibe una corriente de castigo a modo de inhibición sináptica. Paralelamente, las neuronas de los operadores no utilizados no reciben ninguna corriente, lo cual provoca que tiendan al potencial de reposo (E_L).

Para garantizar la estabilidad numérica y la plausibilidad biológica del modelo, las corrientes se han derivado analíticamente a partir del rango dinámico de la membrana. Por una parte, a la hora de establecer la corriente de recompensa elegimos un valor superior a la reobase (I_{rh}), que es el estímulo continuo mínimo necesario para llegar a desencadenar un disparo desde el reposo [4]. La reobase en el modelo AdEx se aproxima anulando la derivada de la ecuación 1:

$$I_{rh} \approx g_L(V_T - E_L) = 10{,}0 \times (-50{,}0 - (-70{,}0)) = 200{,}0 \text{ pA} \ . \tag{3}$$

En base a esto, establecemos una recompensa de $I_{ext} = +300$ pA, un valor que garantiza una excitación rápida y robusta de la neurona. En cambio, teniendo en cuenta por un lado que ante una corriente inhibitoria constante el voltaje de la neurona tiende a un estado estacionario V_∞, y por otro los límites físicos dictados por el potencial de equilibrio del ion potasio ($E_K \approx -80$ mV) [8], establecemos un límite inferior estable de $-75{,}0$ mV, que da lugar a una corriente de castigo de $I_{ext} = -50$ pA mediante una aproximación como la anterior:

$$I_{ext} \approx g_L(V_\infty - E_L) = 10{,}0 \times (-75{,}0 - (-70{,}0)) = -50{,}0 \text{ pA} \ . \tag{4}$$

3.3. Política de decisión termodinámica (softmax)

El potencial de membrana de cada neurona (V_i) actúa como un registro histórico continuo de la efectividad de su operador asociado i. Dado que los voltajes operan en el dominio de los números negativos, se requiere un mapeo no lineal para transformarlos en una distribución de probabilidad válida para la ruleta de selección. Para ello, utilizamos la función softmax, equivalente a la selección de acción de Boltzmann utilizada clásicamente en aprendizaje por refuerzo [13]:

$$P(i) = \frac{\exp\left(\frac{V_i}{T}\right)}{\sum_{j=1}^{K} \exp\left(\frac{V_j}{T}\right)} \ , \tag{5}$$

donde K es el número total de operadores disponibles y T es un parámetro de temperatura, que modula la entropía del sistema. La principal ventaja de esta arquitectura es que desacopla el comportamiento biológico de la neurona de la política de decisión del algoritmo. Valores altos de temperatura ($T \to 10$) suavizan las diferencias de potencial, haciendo menos bruscos los cambios de probabilidad y fomentando la diversidad. Por el contrario, valores bajos ($T \to 1$) amplifican de forma drástica diferencias milivoltáicas mínimas, forzando un comportamiento voraz que margina rápidamente a los operadores ineficientes, promoviendo una convergencia temprana en el espacio de búsqueda.

3.4. Estructura general del algoritmo

La arquitectura propuesta mantiene intacto el motor de búsqueda basado en micropoblaciones de μFAME, conservando el uso de torneos por contribución al hipervolumen para la selección y de un archivo de soluciones no dominadas como población, a la vez que sustituye las variables de control de ventana y estancamiento y el sistema de inferencia difuso por la SNN descrita.

El pseudocódigo resultante se detalla en el algoritmo 1, donde `Inicializar-RedAdEx` establece los valores iniciales de cada neurona ($V_i \leftarrow E_L$; $w_i \leftarrow 0$), `CalcularProbabilidadesSoftmax` traduce los potenciales de las neuronas a las probabilidades de los operadores según la ecuación 5, y `AplicarEcuacionesAdEx` actualiza los valores de las variables neuronales según las ecuaciones 1 y 2 tras cada paso de tiempo (esto es, cada evaluación), tomando dt como 1.

Algoritmo 1 Pseudocódigo de μAdEx

```
Individuos ← Inicializar(N)                          ▷ Generar N individuos aleatorios
PoblaciónHV.añadir(Individuos)
InicializarRedAdEx()                                 ▷ Inicializar V_i y w_i para cada neurona
mientras no se alcance el criterio de parada hacer
    ProbOp ← CalcularProbabilidadesSoftmax()         ▷ Según T y todo V_i
    para i = 1 hasta MáxPadres hacer
        Padres[i] ← TorneoHV(PoblaciónHV)
    fin para
    Operador ← Ruleta(ProbOp)
    Hijo ← Operador(Padres)
    Evaluar(Hijo)
    si no (PoblaciónHV.añadir(Hijo)) entonces
        I_Op ← EstímuloPenalización                  ▷ Inserción fallida: corriente negativa
    sino
        I_Op ← EstímuloRecompensa                    ▷ Inserción exitosa: corriente positiva
    fin si
    AplicarEcuacionesAdEx(Operador, I_Op)            ▷ Actualizar todo V_i y w_i
fin mientras
devolver PoblaciónHV
```

4. Configuración de los experimentos

Para validar el rendimiento y la capacidad de control del mecanismo AOS propuesto, se ha diseñado una batería de experimentos que comparan el algoritmo μFAME original frente a 10 variantes de μAdEx. Todas las implementaciones y la ejecución de los experimentos se toman como base el *framework* de optimización jMetal [3] en su versión 7.3.

El entorno de evaluación seleccionado es la familia de problemas WFG [5], ampliamente utilizada en la literatura debido a su capacidad para generar frentes de Pareto con diversas topologías y diferentes sesgos en el espacio de búsqueda. La dimensionalidad de los problemas utilizada es la misma que la empleada en

el artículo de presentación de μFAME, que además coincide con los valores por defecto de jMetal: 2 variables de posición, 4 variables de distancia y 2 objetivos.

La calidad de las aproximaciones del frente de Pareto se evalúa mediante tres de los indicadores más conocidos de la literatura, replicando el diseño experimental original de μFAME: hipervolumen (I_{HV}), épsilon aditivo ($I_{\epsilon+}$) y distancia generacional invertida plus (I_{IGD+}): I_{HV} proporciona una medida conjunta de convergencia y diversidad basada en el volumen dominado por el frente aproximado; $I_{\epsilon+}$ evalúa el mínimo factor de desplazamiento aditivo necesario para que una aproximación domine completamente a otra, capturando el peor caso de desviación entre frentes; y I_{IGD+} mide la distancia promedio desde puntos del frente óptimo hasta su aproximación. Se busca maximizar I_{HV} a la vez que se desea minimizar $I_{\epsilon+}$ y I_{IGD+}. Los valores de los tres indicadores se calculan con la implementación de jMetal y en base a los frentes de referencia para cada problema incluidos en su repositorio.

Dado el carácter estocástico de las metaheurísticas, se lanzan 100 ejecuciones independientes de cada variante algorítmica, de las que se guardan los valores de los tres indicadores de calidad en cada unidad porcentual de compleción (del 1 al 100 %) y la probabilidad asignada a cada operador genético tras cada evaluación. Para determinar la existencia de diferencias estadísticamente significativas en los indicadores de calidad de los frentes de Pareto finales alcanzados por la propuesta neuronal y el algoritmo base, se aplica el test no paramétrico de suma de rangos de Wilcoxon con un nivel de significancia de $\alpha = 0{,}05$. Adicionalmente, para estudiar el coste computacional de la nueva propuesta, de cada ejecución individual se recogen tanto el tiempo de ejecución total como el tiempo acumulado dedicado al mecanismo de AOS de cada algoritmo (esto es, la resolución del motor difuso en μFAME, y el cálculo de las ecuaciones de la SNN en μAdEx). Los experimentos se ejecutan en un mini PC Minisforum UN1290 con procesador Intel Core i9-12900HK y, para evitar posibles fluctuaciones debido a concurrencia, todas las ejecuciones de los algoritmos se realizan secuencialmente.

El presupuesto computacional de todos los algoritmos se limita a 25.000 evaluaciones de la función objetivo, valor establecido como estándar en la literatura para el análisis de la familia WFG [12].

Para garantizar una comparativa justa y replicar el entorno experimental original de μFAME, ambas familias de algoritmos comparten la misma parametrización y el mismo conjunto de operadores genéticos: evolución diferencial (ED: $p_c = 1{,}0$; $CR = 1{,}0$; $F = 0{,}5$), cruce binario simulado (SBX: $p_c = 1{,}0$; $\eta_c = 20{,}0$), mutación polinomial (MP: $p_m = 0{,}25$; $\eta_m = 20{,}0$) y mutación uniforme (MU: $p_m = 0{,}25$; $u_r = x_i \pm 0{,}05$). Asimismo, el tamaño máximo del archivo de soluciones no dominadas se fija en 100 individuos, el tamaño del torneo en 5 y la ventana de memoria de μFAME en 14 evaluaciones. Por último, para aislar y analizar de forma independiente el impacto de la política de decisión termodinámica, se instancian 10 variantes de μAdEx modificando exclusivamente la temperatura de la función softmax en el rango discreto $T \in \{1, \ldots, 10\}$, las cuales se identifican como T01 hasta T10 respectivamente.

5. Resultados y discusión

En esta sección se analizan los resultados de evaluar la arquitectura propuesta utilizando la batería de problemas WFG. El estudio se divide en tres partes: una comparativa estadística global, un análisis de la velocidad de convergencia, y un estudio de la dinámica de selección de operadores.

5.1. Comparativa de rendimiento global

La tabla 1 muestra la mediana y el rango intercuartílico (IQR) de I_{HV}, $I_{\epsilon+}$ e I_{IGD+} tras 100 ejecuciones a modo de medidas de tendencia central y dispersión estadística. Los fondos gris oscuro y gris claro indican el mejor y segundo mejor resultado absoluto respectivamente (desempatando por menor IQR). Según el test de Wilcoxon frente al algoritmo base, ▼ indica que μFAME es significativamente peor, △ que es mejor, y ≈ señala un empate. Los valores de I_{HV} usan cinco decimales para apreciar estas diferencias. A modo de resumen, la tabla 2 agrega los resultados del test de Wilcoxon, mientras que la tabla 3 agrupa el recuento de podios logrados por las variantes de μAdEx que superan estadísticamente a μFAME.

Los datos por problema (tab. 1) muestran que μAdEx logra superar a μFAME, especialmente en I_{HV}. El escenario más favorable es WFG1, donde varias variantes vencen al algoritmo base en las tres métricas de calidad simultáneamente. En el extremo opuesto, WFG8 es el único problema donde ninguna variante logra imponerse, mientras que WFG4 concentra el mayor número de derrotas, siendo el entorno donde μFAME muestra mayor resistencia.

Por otro lado, un análisis por temperaturas (tabs. 2 y 3) revela que las temperaturas bajas son muy poco competitivas, llegando el caso límite T01 a ser derrotado en los 27 escenarios evaluados debido posiblemente a una selección de operadores excesivamente voraz que provoca un estancamiento prematuro. En cambio, las temperaturas altas logran minimizar las derrotas, aunque son las temperaturas intermedias las que logran el rendimiento más equilibrado: presentan algunas derrotas, pero mantienen el número de victorias de las temperaturas altas y a su vez acaparan la gran mayoría de los primeros y segundos puestos en el *ranking* absoluto.

En base a estas observaciones, y para acotar el alcance del resto de análisis de resultados, se escoge T05 como la variante más equilibrada en cuanto a victorias estadísticas y presencia en el podio, y se toma T01 como línea base del comportamiento degradado. Asimismo, se eligen dos escenarios contrastados: WFG1, donde el beneficio de nuestra propuesta destaca claramente, y WFG4, que representa el reto más competitivo para la arquitectura.

Respecto al coste computacional, las mediciones revelan que el tiempo medio del mecanismo AOS es de $6{,}08 \pm 0{,}17$ ms por ejecución en μFAME, reduciéndose a $3{,}86 \pm 0{,}27$ ms en la variante T05 y a $3{,}49 \pm 0{,}28$ ms en T01. A pesar de ejecutarse tras cada evaluación individual en lugar de cada 14 evaluaciones como el motor difuso, el controlador neuronal resulta un 37 % más rápido en la toma de decisiones. No obstante, como la duración global por ejecución supera los 10 segundos habitualmente, el tiempo invertido en las decisiones del AOS representa menos del 0,05 % del total y resulta despreciable en la práctica.

Tabla 1: Mediana e IQR para I_{HV}, $I_{\epsilon+}$ e I_{IGD+}

		μFAME	T10	T09	T08	T07	T06	T05	T04	T03	T02	T01
I_{HV}	WFG1	6,32749e−1 7,1e−4	6,33110e−1 7,3e−4 ▼	6,33193e−1 6,6e−4 ▼	6,33122e−1 7,6e−4 ▼	6,33086e−1 7,6e−4 ▼	6,33229e−1 7,3e−4 ▼	6,33343e−1 9,5e−4 ▼	6,33276e−1 8,0e−4 ▼	6,32937e−1 1,5e−3 ≈	4,12729e−1 2,0e−1 △	1,91126e−1 2,0e−1 △
	WFG2	5,64661e−1 1,1e−5	5,64665e−1 1,1e−5 ▼	5,64664e−1 9,8e−6 ▼	5,64666e−1 1,1e−5 ▼	5,64664e−1 1,4e−5 ▼	5,64667e−1 9,8e−6 ▼	5,64668e−1 1,0e−5 ▼	5,64668e−1 1,4e−5 ▼	5,64662e−1 5,0e−5 ≈	5,63945e−1 9,0e−3 △	5,35885e−1 4,9e−2 △
	WFG3	4,94924e−1 8,8e−6	4,94928e−1 6,7e−6 ▼	4,94928e−1 7,1e−6 ▼	4,94929e−1 5,8e−6 ▼	4,94931e−1 7,4e−6 ▼	4,94933e−1 6,7e−6 ▼	4,94933e−1 7,4e−6 ▼	4,94934e−1 5,8e−6 ▼	4,94930e−1 1,2e−5 ▼	4,92076e−1 1,5e−2 △	4,83151e−1 3,3e−2 △
	WFG4	2,19492e−1 1,7e−4	2,19459e−1 1,6e−4 ≈	2,19349e−1 1,6e−4 ≈	2,19349e−1 1,7e−4 ≈	2,19344e−1 2,0e−4 △	2,19339e−1 2,5e−4 △	2,19236e−1 1,7e−4 △	2,19176e−1 4,5e−4 △	2,18511e−1 1,0e−3 △	2,17725e−1 2,2e−3 △	2,17214e−1 3,6e−3 △
	WFG5	1,96529e−1 3,3e−5	1,96536e−1 3,5e−5 ▼	1,96533e−1 2,9e−5 ≈	1,96533e−1 3,5e−5 ≈	1,96530e−1 3,6e−5 ≈	1,96534e−1 3,2e−5 ≈	1,96538e−1 3,1e−5 ▼	1,96537e−1 2,9e−5 ▼	1,96541e−1 2,4e−5 ▼	1,96528e−1 4,9e−5 ≈	1,96496e−1 5,1e−3 △
	WFG6	2,10987e−1 3,1e−5	2,11004e−1 1,7e−5 ▼	2,11003e−1 1,5e−5 ▼	2,11006e−1 1,5e−5 ▼	2,11008e−1 1,3e−5 ▼	2,11008e−1 1,4e−5 ▼	2,11010e−1 1,4e−5 ▼	2,11010e−1 1,1e−5 ▼	2,11001e−1 4,1e−5 ▼	2,05718e−1 2,5e−2 △	1,82882e−1 4,2e−2 △
	WFG7	2,11022e−1 5,2e−6	2,11024e−1 4,5e−6 ▼	2,11024e−1 6,4e−6 ≈	2,11024e−1 5,9e−6 ▼	2,11023e−1 5,2e−6 ▼	2,11025e−1 4,3e−6 ▼	2,11025e−1 4,0e−6 ▼	2,11025e−1 5,6e−6 ▼	2,11024e−1 5,0e−6 ▼	2,10966e−1 5,0e−4 △	2,10635e−1 7,4e−3 △
	WFG8	1,52749e−1 1,5e−3	1,52679e−1 1,6e−3 ≈	1,52842e−1 1,2e−3 ≈	1,52892e−1 1,4e−3 ≈	1,52818e−1 1,3e−3 ≈	1,52801e−1 1,4e−3 ≈	1,52900e−1 1,4e−3 ≈	1,52692e−1 1,5e−3 ≈	1,52503e−1 1,2e−3 ≈	1,52094e−1 1,4e−3 △	1,50871e−1 2,2e−3 △
	WFG9	2,40733e−1 1,4e−3	2,40738e−1 2,6e−3 ▼	2,40728e−1 1,7e−3 ≈	2,40719e−1 1,7e−3 ≈	2,40735e−1 2,3e−3 ≈	2,40737e−1 1,8e−3 ≈	2,40682e−1 1,7e−3 ≈	2,39830e−1 3,5e−3 △	2,39067e−1 4,0e−3 △	2,36635e−1 9,1e−3 △	2,36653e−1 7,9e−3 △
$I_{\epsilon+}$	WFG1	2,01e−2 4,0e−3	1,80e−2 5,1e−3 ▼	1,75e−2 4,6e−3 ▼	1,81e−2 5,1e−3 ▼	1,81e−2 5,0e−3 ▼	1,72e−2 4,8e−3 ▼	1,62e−2 6,8e−3 ▼	1,67e−2 5,5e−3 ▼	1,88e−2 9,4e−3 ≈	1,74e−1 1,5e−1 △	3,82e−1 2,3e−1 △
	WFG2	2,25e−3 9,1e−5	2,25e−3 8,9e−5 ≈	2,25e−3 6,5e−5 ≈	2,25e−3 9,7e−5 ≈	2,25e−3 6,9e−5 ≈	2,25e−3 1,1e−4 ≈	2,25e−3 8,3e−5 ≈	2,25e−3 8,6e−5 ≈	2,26e−3 1,1e−4 ≈	2,69e−3 6,8e−3 △	3,27e−2 1,8e−1 △
	WFG3	5,34e−3 1,8e−4	5,31e−3 1,5e−4 ≈	5,32e−3 1,9e−4 ≈	5,34e−3 1,4e−4 ≈	5,32e−3 1,7e−4 ≈	5,31e−3 1,5e−4 ≈	5,33e−3 2,0e−4 ≈	5,30e−3 1,6e−4 ▼	5,32e−3 1,5e−4 ≈	7,05e−3 7,6e−3 △	1,14e−2 1,7e−2 △
	WFG4	4,65e−3 2,1e−4	4,64e−3 2,3e−4 ≈	4,58e−3 2,3e−4 ▼	4,63e−3 2,0e−4 ≈	4,63e−3 2,0e−4 ≈	4,67e−3 2,5e−4 ≈	4,69e−3 2,5e−4 ≈	4,74e−3 3,5e−4 △	5,07e−3 7,2e−4 △	5,53e−3 1,3e−3 △	5,81e−3 2,3e−3 △
	WFG5	2,82e−2 2,4e−4	2,82e−2 2,4e−4 ≈	2,82e−2 2,3e−4 ≈	2,83e−2 2,6e−4 ≈	2,83e−2 2,8e−4 ≈	2,82e−2 2,5e−4 ≈	2,83e−2 2,9e−4 ≈	2,83e−2 2,3e−4 ≈	2,83e−2 2,5e−4 ≈	2,83e−2 3,5e−4 △	2,86e−2 4,3e−3 △
	WFG6	4,39e−3 1,9e−4	4,32e−3 2,3e−4 ▼	4,30e−3 1,6e−4 ▼	4,21e−3 2,0e−4 ▼	4,28e−3 2,0e−4 ▼	4,26e−3 2,1e−4 ▼	4,26e−3 2,2e−4 ▼	4,27e−3 2,0e−4 ▼	4,35e−3 2,8e−4 ≈	7,66e−3 1,7e−2 △	2,34e−2 3,3e−2 △
	WFG7	4,46e−3 1,6e−4	4,44e−3 1,4e−4 ≈	4,45e−3 1,5e−4 ≈	4,46e−3 1,4e−4 ≈	4,43e−3 1,6e−4 ≈	4,44e−3 1,5e−4 ▼	4,45e−3 1,4e−4 ≈	4,42e−3 1,6e−4 ▼	4,44e−3 1,6e−4 ≈	4,58e−3 5,1e−4 △	5,06e−3 5,1e−3 △
	WFG8	2,44e−1 8,0e−2	2,44e−1 1,2e−1 ≈	2,44e−1 1,2e−1 ≈	2,44e−1 1,2e−1 ≈	2,44e−1 1,2e−1 ≈	2,44e−1 2,0e−3 △	2,44e−1 1,2e−1 ≈	2,44e−1 1,1e−1 ≈	2,44e−1 1,2e−1 ≈	2,44e−1 1,2e−1 ≈	2,44e−1 9,8e−2 △
	WFG9	5,64e−3 8,2e−4	5,70e−3 1,2e−3 ≈	5,64e−3 1,0e−3 ≈	5,65e−3 1,1e−3 ≈	5,62e−3 1,1e−3 ≈	5,62e−3 1,2e−3 ≈	5,75e−3 1,1e−3 △	6,05e−3 1,8e−3 △	6,46e−3 2,1e−3 △	7,89e−3 5,8e−3 △	7,95e−3 5,5e−3 △
I_{IGD+}	WFG1	2,13e−3 1,2e−4	2,07e−3 1,3e−4 ▼	2,05e−3 1,3e−4 ▼	2,07e−3 1,3e−4 ▼	2,05e−3 1,4e−4 ▼	2,03e−3 1,0e−4 ▼	2,03e−3 1,4e−4 ▼	2,02e−3 1,5e−4 ▼	2,09e−3 2,2e−4 ≈	1,61e−1 1,6e−1 △	3,77e−1 2,3e−1 △
	WFG2	8,45e−4 5,7e−5	8,33e−4 6,0e−5 ≈	8,30e−4 6,2e−5 ▼	8,39e−4 5,8e−5 ≈	8,45e−4 6,0e−5 ≈	8,38e−4 6,0e−5 ≈	8,31e−4 6,3e−5 ≈	8,40e−4 6,9e−5 ≈	8,37e−4 7,3e−5 ≈	1,22e−3 7,7e−3 △	2,59e−2 3,7e−2 △
	WFG3	2,53e−3 2,4e−5	2,53e−3 2,2e−5 ≈	2,52e−3 2,6e−5 ≈	2,53e−3 2,0e−5 ≈	2,52e−3 2,5e−5 ≈	2,52e−3 2,3e−5 ▼	2,52e−3 2,2e−5 ▼	2,52e−3 2,4e−5 ≈	2,53e−3 2,3e−5 ≈	4,02e−3 8,8e−3 △	9,34e−3 2,2e−2 △
	WFG4	1,99e−3 1,1e−4	2,00e−3 8,6e−5 ≈	2,03e−3 1,2e−4 △	2,03e−3 1,1e−4 △	2,03e−3 1,1e−4 △	2,06e−3 1,2e−4 △	2,08e−3 1,3e−4 △	2,13e−3 2,7e−4 △	2,50e−3 4,9e−4 △	2,84e−3 1,1e−3 △	3,12e−3 1,9e−3 △
	WFG5	2,70e−2 4,1e−5	2,70e−2 5,2e−5 ≈	2,70e−2 4,6e−5 ≈	2,70e−2 5,3e−5 ≈	2,70e−2 4,9e−5 ≈	2,70e−2 5,4e−5 ≈	2,70e−2 5,4e−5 ≈	2,70e−2 4,6e−5 ≈	2,70e−2 5,4e−5 ≈	2,70e−2 4,5e−5 ≈	2,71e−2 4,3e−3 △
	WFG6	1,91e−3 6,8e−5	1,89e−3 6,4e−5 ▼	1,89e−3 6,5e−5 ▼	1,89e−3 5,6e−5 ▼	1,90e−3 5,3e−5 ▼	1,89e−3 5,5e−5 ▼	1,88e−3 6,0e−5 ▼	1,89e−3 4,4e−5 ▼	1,91e−3 5,5e−5 ≈	5,06e−3 1,8e−2 △	2,27e−2 3,5e−2 △
	WFG7	2,04e−3 7,5e−5	2,02e−3 8,2e−5 ≈	2,03e−3 8,1e−5 ≈	2,03e−3 7,4e−5 ≈	2,02e−3 7,5e−5 ▼	2,03e−3 8,6e−5 ≈	2,04e−3 7,6e−5 ≈	2,03e−3 7,2e−5 ≈	2,02e−3 9,3e−5 ≈	2,08e−3 2,9e−4 △	3,21e−3 4,6e−3 △
	WFG8	4,27e−2 7,9e−3	4,28e−2 8,3e−3 ≈	4,27e−2 8,2e−3 ≈	4,26e−2 8,6e−3 ≈	4,28e−2 8,5e−3 ≈	4,28e−2 3,4e−3 ≈	4,29e−2 8,3e−3 ≈	4,27e−2 8,3e−3 ≈	4,32e−2 8,6e−3 ≈	4,32e−2 8,7e−3 ≈	4,45e−2 9,1e−3 △
	WFG9	2,88e−3 7,2e−4	2,88e−3 1,4e−3 ≈	2,88e−3 9,9e−4 ≈	2,89e−3 9,7e−4 ≈	2,88e−3 1,3e−3 ≈	2,90e−3 1,0e−3 ≈	2,93e−3 1,0e−3 △	3,37e−3 2,0e−3 △	3,81e−3 2,3e−3 △	5,24e−3 5,9e−3 △	5,26e−3 5,1e−3 △

Tabla 2: Resumen total del test de Wilcoxon frente a μFAME

Indicador	**T10**	**T09**	**T08**	**T07**	**T06**	**T05**	**T04**	**T03**	**T02**	**T01**
Mejor (▼)	11	10	9	10	11	11	**12**	4	0	0
Empate (≈)	16	16	17	15	13	12	9	17	4	0
Peor (△)	**0**	1	1	2	3	4	6	6	23	27

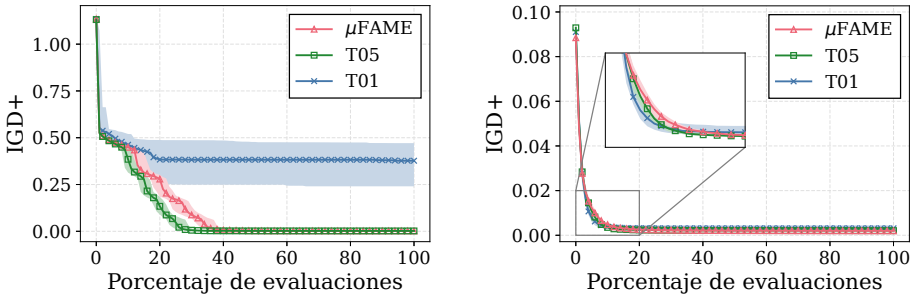
Tabla 3: Recuento de mejores resultados donde se vence a μFAME (▼)

Puesto	**T10**	**T09**	**T08**	**T07**	**T06**	**T05**	**T04**	**T03**	**T02**	**T01**
1º Mejor	1	2	1	1	0	4	**6**	1	0	0
2º Mejor	0	1	0	0	2	**4**	3	0	0	0

En cuanto al tiempo total de optimización, la media global por ejecución se sitúa en $13{,}62 \pm 0{,}31$ s para μFAME, $15{,}00 \pm 0{,}37$ s para T05 y $15{,}63 \pm 0{,}70$ s para T01. Este incremento de aproximadamente un 10 % en el tiempo global de la propuesta no se debe al mecanismo AOS, el cual ha demostrado ser computacionalmente más ligero. La diferencia respecto al algoritmo base puede deberse a una mayor preferencia de μAdEx hacia operadores genéticos más costosos en sus transformaciones sobre las variables de decisión, o a que el descubrimiento de frentes de Pareto más amplios y poblados aumente el coste aritmético derivado de las operaciones de inserción, estimación de densidad y truncamiento del archivo de soluciones no dominadas para preservar la diversidad del frente.

5.2. Análisis de convergencia

Para entender mejor el ritmo de optimización, la figura 1 muestra cómo evoluciona el indicador I_{IGD+} a lo largo de las evaluaciones, donde la línea continua representa la mediana y el área sombreada el IQR.



(a) WFG1 (b) WFG4

Figura 1: Evolución de I_{IGD+} para WFG1 y WFG4

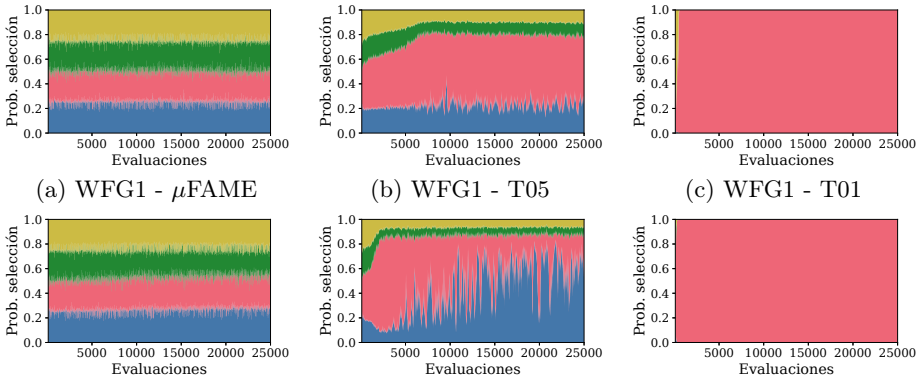
Las gráficas muestran comportamientos muy distintos según el problema. En WFG1, la variante T05 es claramente superior: converge más rápido y consigue el mejor valor final. En cambio, μFAME necesita alrededor de un 10 % más de evaluaciones para llegar a los mismos niveles, mientras que la variante T01 se estanca prematuramente y presenta valores mucho más dispersos.

En WFG4 el escenario es diferente, mostrando menores diferencias. Aquí, T01 empieza bajando muy rápido pero acaba dando el peor resultado final. Por el contrario, μFAME empieza siendo el más lento, pero esa misma inercia le permite afinar mejor al final de la ejecución y lograr el mejor valor de I_{IGD+}. La configuración T05 se mantiene en un punto intermedio, ofreciendo un balance entre rapidez inicial y calidad final.

Esto parece confirmar que un ajuste equilibrado ($T = 5$) aporta la flexibilidad necesaria para resolver problemas complejos sin estancarse, mientras que temperaturas bajas como T01 pueden resultar útiles en determinados escenarios cuando se busca una convergencia inicial muy rápida a costa de la calidad final, y dinámicas más lentas como la de μFAME siguen siendo válidas para el refinado final de las soluciones en topologías concretas.

5.3. Dinámica de selección

Para entender por qué los algoritmos convergen a distinto ritmo, se han analizado las decisiones que toma el controlador AOS internamente. La figura 2 muestra qué probabilidad asigna cada algoritmo a los distintos operadores genéticos a lo largo de las evaluaciones en el caso mediano de las 100 ejecuciones.



(a) WFG1 - μFAME (b) WFG1 - T05 (c) WFG1 - T01

(d) WFG4 - μFAME (e) WFG4 - T05 (f) WFG4 - T01

Figura 2: Evolución de la probabilidad de selección de los operadores para WFG1 y WFG4. Los colores representan cada operador: DE, SBX, MP, y MU.

En las gráficas de μFAME (figs. 2a y 2d) se aprecia que, en WFG1, las probabilidades apenas se desvían del 25 % inicial, funcionando casi como una selección aleatoria en el caso mediano. En WFG4 sí se nota cierta adaptación, pero es tan lenta que frena la velocidad del algoritmo en las primeras fases.

En cuanto a μAdEx, usar una temperatura muy baja ($T = 1$, figs. 2c y 2f) hace que el sistema deje de explorar de forma temprana. Ante los primeros éxitos del operador SBX, la SNN tiende a asignarle prácticamente toda la probabilidad disponible y descarta el resto de operadores. Esto explica el estancamiento prematuro de T01 que se vio en las gráficas de convergencia.

Finalmente, la configuración T05 (figs. 2b y 2e) muestra un comportamiento mucho más equilibrado y reactivo. En WFG1, suele identificar rápido que SBX es el mejor operador y le da prioridad, pero no llega a bloquear a los demás. En WFG4, el controlador neuronal tiende a empezar priorizando SBX, pero cuando el operador DE comienza a dar buenos resultados, la red reacciona al momento y va adaptando las probabilidades, dando lugar a un relevo periódico. Esto parece indicar que la arquitectura puede ajustarse en tiempo real a los cambios del problema sin sufrir los retardos de los controladores AOS tradicionales.

6. Conclusiones y trabajo futuro

Este artículo propone y evalúa μAdEx, un algoritmo evolutivo con una arquitectura diseñada para eliminar la latencia en los mecanismos de selección

adaptativa de operadores. Para resolver este problema, se ha sustituido un controlador tradicional basado en ventanas de memoria por una red neuronal pulsante basada en el modelo AdEx. Este enfoque permite traducir el éxito o fracaso algorítmico directamente con la simulación de inyecciones de corriente eléctrica, logrando que el sistema asigne recompensas y penalizaciones de forma inmediata, sin los retardos habituales.

La validación empírica sobre la suite WFG confirma que esta adaptación en tiempo real puede mejorar el rendimiento general de la búsqueda. Los resultados muestran que un control de temperatura intermedio en la función softmax es capaz de superar estadísticamente al algoritmo base μFAME en indicadores de calidad clave como lo son el hipervolumen, el épsilon aditivo y la distancia generacional invertida plus. El análisis también corrobora un compromiso claro: mientras que el controlador neuronal puede acelerar notablemente la optimización al descartar operadores ineficientes al instante, la convergencia más lenta de μFAME sigue siendo útil para el refinado final en determinados escenarios.

Como trabajo futuro se plantean varias líneas de investigación para ampliar la aplicabilidad de la arquitectura propuesta. A nivel biofísico y termodinámico, se estudiará el desarrollo de reglas heurísticas para ajustar la temperatura según el problema, así como un control adaptativo que transite de forma automática entre fases de exploración y explotación. En esta misma línea, se buscará sustituir las recompensas de estímulo fijo por corrientes continuas proporcionales a la contribución de cada solución al hipervolumen. Asimismo, se analizará la influencia de los parámetros intrínsecos del modelo AdEx mediante un análisis de sensibilidad, evaluando la viabilidad de modificarlos dinámicamente durante la ejecución. A nivel algorítmico, se propone diseñar una red neuronal que regule simultáneamente la elección del operador, sus hiperparámetros y la configuración del torneo de selección. Finalmente, se estudiará la aplicación de este mecanismo en otros motores evolutivos; dado que la red es agnóstica respecto al uso de micropoblaciones y torneos de hipervolumen, el control puede adaptarse a otros paradigmas mediante señales de recompensa alternativas que cuantifiquen el éxito de los operadores según los requisitos de cada método.

Agradecimientos. Esta publicación forma parte del proyecto eMob (PID2022-137858OB-I00), financiado por MICIU/AEI/10.13039/501100011033 y por el Fondo Europeo de Desarrollo Regional (FEDER), «Una manera de hacer Europa» así como de los proyectos GRASS (FEDER-UCA-2024-A2-23), financiado por el Programa Operativo FEDER Andalucía 2021-2027 y la Consejería de Universidad, Investigación e Innovación de la Junta de Andalucía, y gCODE (PIDI_2024_01307), financiado por los Fondos Europeos de Desarrollo Regional (FEDER) y la Consejería de Universidad, Investigación e Innovación de la Junta de Andalucía. C. Benito-Jareño desea agradecer al Ministerio de Ciencia, Innovación y Universidades de España su apoyo mediante la ayuda FPU24/04137.

Declaración de intereses. Los autores declaran no tener conflictos de intereses relevantes para el contenido de este artículo.

Referencias

1. Brette, R., Gerstner, W.: Adaptive exponential integrate-and-fire model as an effective description of neuronal activity. Journal of Neurophysiology **94**(5), 3637–3642 (2005). https://doi.org/10.1152/jn.00686.2005
2. David Schaffer, J.: Evolving spiking neural networks: A novel growth algorithm corrects the teacher. In: 2015 IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA). pp. 1–8 (2015). https://doi.org/10.1109/CISDA.2015.7208630
3. Durillo, J.J., Nebro, A.J.: jMetal: A Java framework for multi-objective optimization. Advances in Engineering Software **42**(10), 760–771 (2011). https://doi.org/10.1016/j.advengsoft.2011.05.014
4. Gerstner, W., Kistler, W.M., Naud, R., Paninski, L.: Neuronal Dynamics: From Single Neurons to Networks and Models of Cognition. Cambridge University Press (2014)
5. Huband, S., Hingston, P., Barone, L., While, L.: A review of multiobjective test problems and a scalable test problem toolkit. IEEE Transactions on Evolutionary Computation **10**(5), 477–506 (2006). https://doi.org/10.1109/TEVC.2005.861417
6. Izhikevich, E.: Simple model of spiking neurons. IEEE Transactions on Neural Networks **14**(6), 1569–1572 (2003). https://doi.org/10.1109/TNN.2003.820440
7. Jeong, H.Y., Song, B.D.: Meta-learning-based adaptive operator selection for traveling salesman problem. Applied Soft Computing **185**, 113930 (2025). https://doi.org/10.1016/j.asoc.2025.113930
8. Kandel, E.R., Schwartz, J.H., Jessell, T.M. (eds.): Principles of Neural Science. Elsevier, New York, third edn. (1991)
9. Naud, R., Marcille, N., Clopath, C., Gerstner, W.: Firing patterns in the adaptive exponential integrate-and-fire model. Biological Cybernetics **99**(4–5), 335–347 (2008). https://doi.org/10.1007/s00422-008-0264-7
10. Parsa, M., Kulkarni, S.R., Coletti, M., Bassett, J., Mitchell, J.P., Schuman, C.D.: Multi-objective hyperparameter optimization for spiking neural network neuroevolution. In: 2021 IEEE Congress on Evolutionary Computation (CEC). pp. 1225–1232 (2021). https://doi.org/10.1109/CEC45853.2021.9504897
11. Pei, J., Mei, Y., Liu, J., Zhang, M., Yao, X.: Adaptive operator selection for metaheuristics: A survey. IEEE Transactions on Artificial Intelligence **6**(8), 1991–2012 (2025). https://doi.org/10.1109/TAI.2025.3545792
12. Santiago, A., Dorronsoro, B., Fraire, H.J., Ruiz, P.: Micro-genetic algorithm with fuzzy selection of operators for multi-objective optimization: μFAME. Swarm and Evolutionary Computation **61**, 100818 (2021). https://doi.org/10.1016/j.swevo.2020.100818
13. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction. The MIT Press, second edn. (2018)
14. Yin, S., Xiang, Z.: Adaptive operator selection with dueling deep q-network for evolutionary multi-objective optimization. Neurocomputing **581**, 127491 (2024). https://doi.org/10.1016/j.neucom.2024.127491