

UNIVERSIDAD DE GRANADA

Teaching–Learning-Based Optimization

Metaheurísticas

Curso Académico 2023-2024

Nombre del Estudiante: Pablo Olivares Martínez

DNI: 24411228V

Email: pablolivares@correo.ugr.es

Grupo: 1

Pablo Olivares Martínez

19 de junio de 2024

ÍNDICE

1 Descripción del problema	3
2 Teaching-Learning Based Optimization	3
2.1 Fase del Profesor	4
2.2 Fase del Aprendiz	5
3 Análisis y propuesta de mejora	6
3.1 Hibridación TLBO-SW	6
3.2 TLBO modificado con elitismo y eliminación de duplicados	7
3.3 TLBO con consciencia del sesgo	7
3.4 Hibridación TLBO-SW elitista con consciencia de sesgo	8
4 Manual de usuario y ejecución	8
5 Experimentos y análisis de resultados	9
5.1 Resultados medios	10
5.2 Resultados en rankings	14
Apéndice	16

1. DESCRIPCIÓN DEL PROBLEMA

En la actualidad, un gran número de problemas surgen en el ámbito de la optimización de funciones. Además, el volumen de datos relacionado con los problemas a resolver es cada vez mayor. Como consecuencia, áreas del conocimiento como el aprendizaje automático o las metaheurísticas han avanzado notablemente con la intención de resolver estos problemas. Más formalmente, definimos un problema de **optimización en varias variables reales** como la búsqueda de un vector \mathbf{x}^* de dimensión D en un dominio Ω de forma que

$$\mathbf{x}^* = \arg \min_{\mathbf{x} \in \Omega} F(\mathbf{x}),$$

siendo $F : \Omega \rightarrow \mathbb{R}$ la función a optimizar y $\Omega \subseteq \mathbb{R}^D$. Generalmente, se le conoce a F como **función objetivo**. El propósito será optimizar dicha función a partir de un conjunto de datos $\mathcal{D} = (\mathbf{x}_i, y_i)_{i=1}^M$, donde M es el número de muestras, \mathbf{x}_i son las instancias de las muestras del dominio e y_i su imagen $F(\mathbf{x}_i)$.

En particular, nosotros trabajaremos sobre el conocido *benchmark* **CEC' 2017**, compuesto por 30 funciones *test* a minimizar definidas sobre el dominio $\Omega = [-100, 100]^D$. Cada una de estas funciones se evaluará para las dimensiones $D \in \{10, 30, 50\}$ un total de 10 veces con distintas semillas.

2. TEACHING-LEARNING BASED OPTIMIZATION

Existen diversas formas de afrontar un problema de este tipo. Una de las más populares es el uso de **metaheurísticas**, procesos de búsqueda en espacios de soluciones cuya exploración para encontrar el óptimo global es inabarcable en términos computacionales. Estos procesos muchas veces se inspiran en la naturaleza o en las relaciones sociales entre individuos para explorar dicho espacio. Nosotros vamos a centrarnos en el estudio de una metaheurística en particular, conocida como **Teaching-Learning Based Optimization** (TLBO) propuesta por Rao et al. en 2011 [6].

La metaheurística TLBO es un algoritmo basado en poblaciones y, en concreto, en el comportamiento social humano, cuya heurística está inspirada en el método de aprendizaje humano dentro del sistema educativo. Aquí, un **profesor** interactúa con sus **alumnos** para transmitirles sus conocimientos y estos, interactúan entre ellos para afianzar dichos conocimientos.

Supongamos que tenemos dos profesores T_1 y T_2 que imparten el mismo curso a dos grupos diferentes G_1 y G_2 respectivamente, ambos con la misma capacidad de aprendizaje siguiendo una distribución normal. Si evaluamos ambos grupos en la materia, podemos suponer que el grupo G_1 es mejor que G_2 si la media de los resultados es mayor, es decir, si $f(\bar{G}_1) > f(\bar{G}_2)$, donde $f : \Omega \rightarrow \mathbb{R}$ es la función de evaluación o *fitness* y Ω representa el espacio de todas las configuraciones posibles de una clase. Como ambos grupos tenían la misma capacidad de aprendizaje, entonces podemos deducir que el profesor T_1 es mejor en la enseñanza de dicho curso que T_2 .

Basado en este pretexto, TLBO sigue el siguiente esquema. Primero, consideramos una población inicial G_1 , que llamaremos **clase**, y tomaremos como profesor T_1 al individuo de la

población con el mejor valor de *fitness*. El objetivo del profesor será «transmitir su conocimiento» de forma que el resultado global de la clase mejore. Una vez que el profesor haya logrado su objetivo, los alumnos interactuarán entre ellos para compartir sus conocimientos y, en esta nueva población G_2 , se escogerá un nuevo profesor T_2 que sea capaz de mejorar las habilidades de esta nueva población. Esta idea se modela en dos fases: la «**Fase del Profesor**» y la «**Fase del Aprendiz**».

A continuación se muestra el pseudocódigo del bucle principal de ejecución:

Algorithm 1 Bucle Principal de TLBO

```

1:  $t \leftarrow 0$ , evals  $\leftarrow 0$ 
2: Inicializar población de tamaño  $N$ ,  $G_0$ , con valores en  $\mathcal{U}([0, 1])^D$ 
3: Evaluamos toda la población  $G_0$  ▷ Se almacena en un vector
4: evals  $\leftarrow$  evals +  $N$ 
5: while evals < MAX_EVALS do
6:   FASEPROFESOR( $G_t$ )
7:   FASEAPRENDIZ( $G_t$ )
8:    $t \leftarrow t + 1$ 
9: end while

```

2.1. FASE DEL PROFESOR

En esta primera fase, el profesor T_t intenta que todos los alumnos, cuya clase en es representada por G_t , alcancen un rendimiento óptimo en la iteración t . Denotemos por \bar{G}_t el valor medio de las características de todos los alumnos en dicha iteración. La dinámica de mejora de la clase se modela a través del ajuste de la media de acuerdo a la influencia del profesor:

$$\Delta \bar{G}_t = \eta_t (T_t - T_F \bar{G}_t),$$

donde $\eta_t \in \mathcal{U}([0, 1])$ es un valor aleatorio en una distribución uniforme en $[0, 1]$ y T_F es el **factor de enseñanza**, que heurísticamente determina la influencia del profesor sobre la media de la clase. El valor de T_F puede ser 1 o 2, obtenido de una distribución uniforme discreta. Finalmente, actualizamos las características de cada individuo $X_{t,i}$ en la clase de tamaño N , para cada $i \in \{1, \dots, N\}$, de la siguiente manera:

$$X'_{t,i} = X_{t,i} + \Delta \bar{G}_t.$$

La actualización intenta acercar a cada estudiante a la nueva media ajustada, mostrando así el efecto acumulativo de la enseñanza sobre la clase.

Finalmente, se aplica un método de selección de candidatos conocido como **heurística de Deb** para problemas definidos en intervalos restringidos. Dados dos candidatos, se mantiene en la población aquel que:

1. Si ambos candidatos cumplen las restricciones, se escoge aquel con mejor valor *fitness*.
2. Si un candidato cumple las restricciones pero otro no, se escoge el que las cumple.

3. Si ambos candidatos violan las restricciones, se escoge el que las viole en menor medida.

Algorithm 2 Fase del Profesor en TLBO

```

1:  $\bar{G}_t \leftarrow$  Calcular vector medio de  $G_t$ 
2: Inicializar profesor  $T_t$  con el peor valor de fitness
3: for  $X \in G_t$  do                                     ▷ Identificación del profesor
4:   if  $f(X) < f(T_t)$  then
5:      $T_t \leftarrow X$ 
6:   end if
7: end for
8: for  $i = 1$  to  $N$  do                                     ▷ Proceso de enseñanza
9:    $T_F \leftarrow$  Aleatorio entre 1 y 2
10:   $\eta_t \leftarrow$  Valor de  $\mathcal{U}([0, 1])$ 
11:   $X'_{t,i} \leftarrow X_{t,i} + \eta_t \cdot (T_t - T_F \bar{G}_t)$ 
12:  Evaluar  $f(X'_{t,i})$ 
13:  Aplicar heurística de Deb para  $X_{t,i}$ ,  $X'_{t,i}$ 
14:  Incrementar evals; if  $evals \geq MAX\_EVALS$  then break
15: end for

```

2.2. FASE DEL APRENDIZ

La «Fase del Aprendiz» se enfoca en la interacción entre los alumnos en cada iteración, más allá de la influencia directa del profesor. En esta fase, cada alumno, representado por un vector de características $X_{t,i}$ en la iteración t , aprende interaccionando con sus pares:

Algorithm 3 Fase del Aprendiz en TLBO

```

1: for  $i = 1$  to  $N$  do
2:   Seleccionar aleatoriamente otro índice  $j$  con  $j \neq i$ 
3:   Inicializar  $\eta_{t,i} \in \mathcal{U}([0, 1])$ 
4:   if  $f(X_{t,j}) < f(X_{t,i})$  then
5:      $X'_{t,i} \leftarrow X_{t,i} + \eta_{t,i} \cdot (X_{t,i} - X_{t,j})$ 
6:   else
7:      $X'_{t,i} \leftarrow X_{t,i} + \eta_{t,i} \cdot (X_{t,j} - X_{t,i})$ 
8:   end if
9:   Calculamos  $f(X'_{t,i})$ 
10:   $evals \leftarrow evals + 1$ 
11:  Aplicamos la heurística de Deb para  $X_{t,i}$  y  $X'_{t,i}$ 
12:  Incrementar evals; if  $evals \geq MAX\_EVALS$  then break
13: end for
14:  $G_{t+1} \leftarrow G_t$ 

```

La función $f(X_{t,i})$ evalúa la aptitud o *fitness* del alumno basado en sus características en la iteración t dentro de la población. En ambas fases se ha almacenado dicho valor para evitar repetir llamadas a la función objetivo.

3. ANÁLISIS Y PROPUESTA DE MEJORA

Esta estrategia de aprendizaje facilita la **exploración** del espacio de soluciones mediante la interacción entre soluciones, permitiendo que los alumnos no solo adopten conocimientos del mejor entre ellos, sino que también exploren nuevas posibilidades al ajustar sus características basándose en las comparaciones con otros alumnos. Por otro lado, la actualización de la posición sigue un esquema heurístico de **explotación** donde los alumnos con peores desempeños tienden a moverse hacia el profesor, promoviendo así la mejora continua de la clase en su conjunto. Sin embargo, este algoritmo carece de mecanismos que permitan a una solución individual explotar sus capacidades por sí solo. Además, TLBO muestra un claro sesgo hacia las soluciones centradas en el origen, lo que puede resultar una desventaja para problemas cuyo óptimo global no se encuentre en el origen [3].

Para solventar los diversos problemas que presenta este modelo han surgido varias propuestas. Primero, Rao et al. [5] propusieron un leve cambio respecto al modelo original para adaptarlo al problema con restricciones y seleccionar la mejor solución de ambas en cada iteración. Posteriormente, trabajos como el de Črepinšek et al. [1] mostraron una revisión del modelo original unificando los dos bucles de ambas fases en uno, eliminando duplicados y aplicando otro tipo de elitismo, mostrando mejoras en eficiencia y resultados. Nuevas versiones del modelo han ido apareciendo desde entonces. **Improved TLBO** (I-TLBO) fue propuesto por Rao y Patel en 2012 [4] introduciendo varios profesores de forma simultánea, un factor de aprendizaje adaptativo y aprendizaje individual. Más tarde, trabajos como el de Kommadath y Kotecha [2] con **Teaching Learning Based Optimization with Focused Learning** (TLBO-FL) han mostrado buenos resultados en competiciones como CEC' 2017, mostrando mejores resultados que el modelo base en funciones de composición, el cual presentaba convergencia prematura de la población. Aquí, TLBO-FL lleva a cabo la «Fase del Aprendiz» sólo si el alumno no mejoraba tras la acción del maestro, donde esta vez dicho alumno aprendía de otros dos compañeros al mismo tiempo.

Dada su popularidad y rendimiento, hoy en día existen un gran número de algoritmos inspirados en TLBO, desde aquellos empleando infinidad de estrategias de aprendizaje diferentes hasta hibridaciones de todo tipo [7]. En lo que sigue, presentaremos una propuesta de mejora mediante una hibridación con un algoritmo de búsqueda local y proponemos una serie de mejoras con el fin de evitar la convergencia prematura, el sesgo en el origen y una mejor explotación. Para ello, realizaremos implementaciones progresivas para estudiar cómo afectan los cambios al modelo original.

3.1. HIBRIDACIÓN TLBO-SW

En el análisis anterior comentábamos que uno de los posibles puntos débiles que presentaba este algoritmo es su falta de exploración individual. Por ello proponemos una hibrida-

ción de TLBO con el conocido algoritmo de búsqueda local **Solis-Wets**, a la que llamaremos **Teaching-Learning Based Optimization with Solis-Wets** (TLBO-SW).

El algoritmo de Solis-Wets es una técnica de optimización estocástica que actualiza un vector de soluciones $\mathbf{x} \in \mathbb{R}^D$ utilizando un vector de sesgo $\mathbf{b} \in \mathbb{R}^D$ y un parámetro de perturbación $\delta \in \mathbb{R}$. En cada iteración, un vector de perturbaciones $\mathbf{d} \in \mathbb{R}^D$, muestreado de una distribución uniforme $\mathcal{U}([-\delta, \delta]^D)$, se utiliza para generar una nueva solución $\mathbf{x}' = \mathbf{x} + \mathbf{d} + \mathbf{b}$. Si el valor *fitness* $f(\mathbf{x}')$ es menor que $f(\mathbf{x})$, entonces \mathbf{x} se actualiza a \mathbf{x}' y \mathbf{b} se incrementa según $\mathbf{b} \leftarrow 0,2\mathbf{b} + 0,4(\mathbf{d} + \mathbf{b})$; en caso contrario, se prueba una actualización en la dirección contraria $\mathbf{x}'' = \mathbf{x} - (\mathbf{d} + \mathbf{b})$. Si $f(\mathbf{x}'')$ mejora sobre $f(\mathbf{x})$, \mathbf{x} se actualiza a \mathbf{x}'' y \mathbf{b} se decrementa. Los parámetros δ y \mathbf{b} se ajustan dinámicamente en respuesta al historial de éxito o fracaso de las iteraciones. Nuestra implementación TLBO-SW aprovechará la explotación de Solis-Wets cada vez que un alumno haya finalizado su interacción con sus compañeros. De esta forma, podemos interpretar esta fase de optimización como una «sesión de estudio intensa», donde el alumno, una vez resueltas sus dudas, decide estudiar en profundidad la materia. Dado que esta explotación se aplica a todos los alumnos por igual, se realizará una **explotación de baja intensidad** para poder aprovechar más generaciones, haciendo un máximo de 50 evaluaciones por ejecución de Solis-Wets. La implementación es idéntica a la original, con la excepción de la llamada realizada a Solis-Wets con el i -ésimo individuo al final de cada «Fase del Aprendiz».

3.2. TLBO MODIFICADO CON ELITISMO Y ELIMINACIÓN DE DUPLICADOS

Para hacer la implementación algo más eficiente, unificaremos los dos bloques en un bucle. Reduciremos la convergencia prematura eliminando duplicados en cada iteración. Además, se realizará la selección elitista descrita en [5]: se divide la población en dos mitades y se comparan por pares sin repetición. Si el individuo de la segunda mitad es mejor que el de la primera, sustituye al de la primera. De esta forma, la mejor solución siempre se mantiene y se aumenta la calidad media de las soluciones. Dado que este mecanismo añade redundancia, se realiza una mutación de duplicados tras esta fase a un porcentaje de las características (en este caso, 0,1) aleatoriamente de una distribución normal $\mathcal{N}(0, 1)$. Finalmente, despreciaremos la heurística de Deb en favor de un simple truncamiento en la frontera. Posteriormente, el individuo modificado comparará su *fitness* con el del original y sobrevivirá el mejor de ambos. Esta es la implementación más común de este algoritmo actualmente. Notaremos esta modificación por TLBO-E.

3.3. TLBO CON CONSCIENCIA DEL SESGO

Anteriormente, destacábamos como propiedad de TLBO su sesgo hacia el origen. Para solucionarlo, aplicaremos la propuesta de [3] no para eliminarlo, sino para emplearlo a nuestro favor. La modificación al TLBO propone un cambio en el marco de referencia durante la «Fase del Profesor» para mejorar la capacidad del algoritmo de explorar el espacio de búsqueda de manera más efectiva y reducir el sesgo hacia el origen. Notaremos esta modificación por TLBO-B.

Modificaremos el marco de referencia para centrar la búsqueda alrededor de la mejor solución encontrada hasta el momento, T_t , en lugar de centrarse alrededor del origen, que puede

no ser relevante para el problema en cuestión. Las siguientes ecuaciones muestran cómo se alteran las variables en relación con X_{ref} , que es una medida ajustada basada en T_t :

$$\begin{aligned} X'_{nuevo,i} &= X_{nuevo,i} - X_{nuevo}, \\ X'_i &= X_i - X_{ref}, \\ \Delta \bar{G}_t &= \bar{G}_t - X_{ref}, \\ T'_t &= T_t - X_{ref}. \end{aligned}$$

Este reajuste permite que el algoritmo se adapte al mejor desempeño observado, lo que puede ayudar a evitar mínimos locales y explorar áreas más prometedoras del espacio de búsqueda.

Ahora, la «Fase del Profesor» se modifica para utilizar el nuevo marco de referencia, lo cual afecta cómo los individuos ajustan sus posiciones en respuesta al profesor, que es la mejor solución de la generación actual:

$$\begin{aligned} X'_{dif_i} &= \eta_i \cdot (T_t - T_F \bar{G}_t + (T_f - 1) X_{ref}), \\ X_{nuevo,i} &= X_i + X'_{dif_i}, \end{aligned}$$

donde T_F y η_i son como los definimos originalmente.

Finalmente, X_{ref} se obtiene a partir de un operador de mutación polinomial de manera que refleje la dirección de la mejor solución actual T_t , pero ajustada para mantener la exploración activa y evitar el sesgo de convergencia:

$$X_{ref} = \begin{cases} T_t + (r_1^{1/(n_m+1)} - 1)(T_t - X^-), & \text{si } (T_t - X_i < 0) \text{ o } (T_t - X_i = 0 \text{ y } r_2 < 0,5), \\ T_t + (1 - r_1^{1/(n_m+1)})(X^+ - T_t), & \text{de lo contrario.} \end{cases}$$

Aquí, X^+ y X^- denotan los vectores de límites superior e inferior respectivamente. Esta fórmula asegura que X_{ref} se ajuste cerca de T_t pero con suficiente flexibilidad para adaptarse a las características del espacio de búsqueda, utilizando r_1 y r_2 para ajustar la posición basada en el rendimiento actual y la topología del problema.

3.4. HIBRIDACIÓN TLBO-SW ELITISTA CON CONSCIENCIA DE SESGO

Finalmente, aplicaremos la hibridación realizada en TLBO-SW de forma elitista a un porcentaje de la población en conjunto con las anteriores modificaciones de las secciones 3.2 y 3.3. A diferencia de TLBO-SW, esta versión tan sólo explota las características del 10% mejor de la población. Además, con el fin de evitar la convergencia prematura, la optimización se realiza cada 10 generaciones. A cambio, se ha aumentado la intensidad del algoritmo de búsqueda local a 250 evaluaciones por individuo.

4. MANUAL DE USUARIO Y EJECUCIÓN

Los algoritmos han sido implementados sobre el código base proporcionado por Daniel Molina en el repositorio de GitHub `cec2017real`. Todos los algoritmos están implementados en archivos individuales:

- **TLBO**: Implementado en el archivo `testTLBO.cc`. Lo abreviaremos como `tlbo`.
- **TLBO-SW**: Implementado en el archivo `testTLBOSW.cc`. Lo abreviaremos como `tlbosw`.
- **TLBO-E**: Implementado en archivo `testTLBOImproved.cc`. Lo abreviaremos como `tlboe`.
- **TLBO-B**: Implementado en archivo `testTLBObias.cc`. Lo abreviaremos como `tlbob`.
- **TLBO-EBSW**: Implementado en el archivo `testTLBOEBSW.cc`. Lo abreviaremos como `tlboebsw`.

Para compilar los binarios, ejecuta `cmake .` desde el directorio `/code`. A continuación, ejecuta `make <nombre_algoritmo>`, donde `<nombre_algoritmo>` es la abreviación de dicho nombre.

Para ejecutarlo, basta con llamar al binario con argumento el número de dimensiones:

```
./test<nombre_algoritmo> <dimensiones>
```

Aquí, `<nombre_algoritmo>` debe ser sustituido por el nombre abreviado del algoritmo, que se utilizará para crear el archivo de salida correspondiente. El binario ejecutará 10 veces el algoritmo para cada una de las 30 funciones. Para la ejecución $t \in \{0, \dots, 9\}$, se empleará la semilla $43 + t$. El binario devolverá la media de las 10 ejecuciones para cada función en su respectivo documento de texto. Por ejemplo, para obtener la media de las 10 ejecuciones de TLBO-B de dimensión 50 para las 30 funciones haríamos:

```
./testtlbob 50
```

Otra forma de ejecución alternativa es llamando al script `run.sh`, que ejecutará los binarios de todas las implementaciones para dimensiones 10, 30 y 50.

Después de completar los experimentos, puedes generar un archivo Excel con los resultados obtenidos. Este archivo es necesario para presentar los resultados en la plataforma Tacolab. Ejecuta el siguiente comando en la terminal para generar el archivo:

```
$ python extract.py results_<nombre_algoritmo>
```

Este script procesa los resultados almacenados y genera un archivo de resultados de ejecución `results_cec2017_<nombre_algoritmo>.xlsx`. Este archivo está listo para ser enviado a <https://tacolab.org/>, donde se puede comparar tu rendimiento con el de otros algoritmos.

5. EXPERIMENTOS Y ANÁLISIS DE RESULTADOS

A continuación se van a comparar los resultados obtenidos en el *benchmark* CEC'2017 en problemas de optimización no lineal restringidos de variable real. Para ello, estableceremos la población de cada algoritmo propuesto a 50 individuos. Realizaremos dos comparaciones.

En la primera, compararemos TLBO y las variantes implementadas en media. Además, añadiremos al estudio la variante de TLBO-FL, que participó en CEC'2017. En la segunda comparativa, hemos realizado una comparación en *rankings* con los siguientes tres algoritmos base:

- **Differential Evolution (DE):** Algoritmo de optimización basado en la evolución diferencial, que utiliza diferencias entre vectores de la población para crear nuevas soluciones. Los nuevos candidatos se generan sumando la diferencia ponderada entre dos individuos a un tercero, y se seleccionan las mejores soluciones para la siguiente generación en un proceso iterativo.
- **Particle Swarm Optimization (PSO):** Inspirado en el comportamiento social de enjambres, como bandadas de aves, donde las partículas (individuos) se mueven en el espacio de búsqueda siguiendo a las mejor posicionadas. Cada partícula ajusta su posición basándose en su experiencia personal y la experiencia de sus vecinos, buscando encontrar el óptimo global.
- **Gaining-sharing knowledge based algorithm (GSKA):** Inspirado en el comportamiento humano al compartir conocimiento, utiliza dos criterios para optimizar cada dimensión. En las fases iniciales, se aplica el criterio de *junior gaining-sharing*, donde la variable se actualiza considerando las variables de individuos inmediatamente mejores y peores. En fases posteriores, se aplica el criterio de *senior gaining-sharing*, considerando a los mejores y peores individuos de la población.

Los experimentos se han llevado a cabo en un procesador Intel® Core™ i5-8300H de 4 núcleos a 2.30 GHz.

5.1. RESULTADOS MEDIOS

En la Tabla 5.1 podemos ver los resultados para dimensión 10. Se puede observar que TLBO-FL es el mejor algoritmo frente a los implementados. A pesar de ello, aquellos que incluyen una explotación individual con Solis-Wets son capaces de mejorar a TLBO-FL en algunas funciones. Una de las principales ventajas de TLBO-FL es la selección inteligente de individuos a mejorar, pues sólo mejora aquellos que no han aprendido durante la fase del profesor. Esto da lugar a un mejor aprovechamiento de las llamadas a la función objetivo. Además, al recibir influencia de dos compañeros al mismo tiempo, éste no se desplaza directamente hacia uno de ellos, permitiéndole explorar el espacio que hay entre ambos compañeros. Concluimos que el éxito de TLBO-FL puede deberse principalmente a una exploración mayor y una explotación más eficiente.

Por otro lado, TLBO muestra haber sido el peor de todos, sugiriendo que las mejoras han tenido efecto en mayor o menor medida. A diferencia de lo esperado, TLBO-E ha mostrado peor rendimiento que TLBO y sin embargo, su combinación con TLBO-EBSW ha sido efectiva. En TLBO-E, al aplicarse la selección elitista y no añadir suficientes mutaciones, el algoritmo tiende a una convergencia prematura. Este factor fue resuelto en TLBO-EBSW, donde se aumentó dicho porcentaje.

Otra observación interesante es la mejora general de TLBO-B frente a TLBO y TLBO-E, indicando que probablemente la solución óptima no se encuentre en el origen en la mayoría de funciones.

	TLBO	TLBO-SW	TLBO-E	TLBO-B	TLBO-EBSW	TLBO-FL
F01	2.256e+09	2.804e+03	1.910e+09	1.999e+07	7.612e+02	2.023e+03
F02	1.629e+07	4.618e+07	2.365e+08	8.363e+04	4.174e+09	1.961e-02
F03	4.286e+03	2.550e-01	6.083e+03	1.791e+02	2.000e-10	1.111e-04
F04	1.057e+02	1.823e+00	1.378e+02	7.699e+00	1.040e+00	3.032e+00
F05	4.461e+01	2.708e+01	5.784e+01	2.733e+01	3.442e+01	8.754e+00
F06	2.453e+01	2.838e+01	3.493e+01	1.637e+01	3.834e+01	8.390e-08
F07	4.930e+01	3.761e+01	1.322e+02	3.736e+01	2.048e+02	2.764e+01
F08	3.039e+01	1.492e+01	5.347e+01	1.873e+01	2.836e+01	1.228e+01
F09	1.300e+02	1.246e+02	4.879e+02	1.028e+02	5.879e+02	8.908e-03
F10	1.194e+03	7.430e+02	1.378e+03	1.020e+03	9.582e+02	9.546e+02
F11	9.080e+01	3.598e+01	1.822e+02	6.824e+01	4.264e+01	4.117e+00
F12	3.076e+06	4.504e+03	1.370e+07	6.624e+04	1.502e+05	6.556e+04
F13	2.955e+03	1.235e+03	1.396e+04	3.382e+03	3.995e+03	2.448e+03
F14	1.677e+02	4.496e+01	7.835e+01	6.965e+01	3.890e+02	6.728e+01
F15	1.415e+03	6.467e+01	8.387e+02	9.503e+02	4.939e+03	1.259e+02
F16	2.410e+02	1.562e+02	1.577e+02	1.254e+02	1.404e+02	8.906e+00
F17	7.222e+01	5.521e+01	1.093e+02	7.803e+01	1.319e+02	3.830e+01
F18	1.854e+03	1.173e+02	2.671e+04	6.593e+02	3.083e+03	6.151e+03
F19	4.518e+03	3.232e+01	9.448e+02	1.834e+03	6.651e+03	6.059e+01
F20	8.751e+01	8.216e+01	1.706e+02	7.886e+01	1.322e+02	1.461e+01
F21	1.969e+02	1.432e+02	1.944e+02	1.993e+02	1.290e+02	1.424e+02
F22	2.240e+02	1.067e+02	3.890e+02	1.187e+02	1.087e+02	9.331e+01
F23	4.153e+02	3.734e+02	3.441e+02	3.356e+02	3.351e+02	3.069e+02
F24	2.863e+02	1.000e+02	3.658e+02	3.641e+02	1.612e+02	3.104e+02
F25	4.731e+02	4.277e+02	5.583e+02	4.376e+02	3.719e+02	4.255e+02
F26	6.443e+02	4.107e+02	7.468e+02	3.878e+02	3.683e+02	3.009e+02
F27	4.867e+02	4.255e+02	4.056e+02	4.259e+02	4.044e+02	3.925e+02
F28	6.087e+02	5.364e+02	6.423e+02	6.536e+02	3.920e+02	4.470e+02
F29	3.488e+02	3.169e+02	3.845e+02	3.403e+02	3.637e+02	2.745e+02
F30	3.725e+06	2.002e+04	1.578e+06	2.191e+06	5.702e+05	2.790e+05

Cuadro 5.1: Resultados medios tras el 100% de las evaluaciones de la función objetivo para dimensión 10.

La Tabla 5.2 muestra resultados similares a los de la Tabla 5.1. Sin embargo, esta vez TLBO-FL toma más ventaja sobre nuestras implementaciones. Aquí empieza a notarse el manejo eficiente de las llamadas a la función objetivo, pues más variables implican una mayor difi-

cultad para encontrar el óptimo. Esto también se ve reflejado en TLBO-E frente a TLBO, pues ahora mejora varias de las marcas frente a TLBO respecto a la dimensión 10.

	TLBO	TLBO-SW	TLBO-E	TLBO-B	TLBO-EBSW	TLBO-FL
F01	4.734e+10	7.051e+03	6.168e+10	6.641e+09	3.190e+02	3.508e+03
F02	1.749e+44	1.133e+43	2.625e+42	1.251e+32	1.149e+39	8.520e+16
F03	6.105e+04	3.241e+04	7.152e+04	1.973e+04	9.266e+04	2.995e+03
F04	9.336e+03	1.016e+02	1.373e+04	1.163e+03	7.993e+01	9.007e+01
F05	3.392e+02	2.006e+02	3.890e+02	2.588e+02	2.809e+02	3.953e+01
F06	7.279e+01	5.185e+01	8.016e+01	5.524e+01	6.626e+01	4.872e-01
F07	5.401e+02	4.409e+02	1.129e+03	4.247e+02	1.253e+03	1.393e+02
F08	2.911e+02	1.406e+02	3.598e+02	1.903e+02	2.543e+02	3.665e+01
F09	6.125e+03	4.184e+03	1.063e+04	4.535e+03	7.482e+03	3.446e+01
F10	6.782e+03	3.613e+03	7.423e+03	5.614e+03	4.309e+03	6.687e+03
F11	3.314e+03	1.591e+02	5.228e+03	4.775e+02	1.673e+02	8.161e+01
F12	6.980e+09	1.876e+06	7.984e+09	3.148e+08	7.648e+05	5.746e+04
F13	3.846e+09	9.451e+04	1.830e+09	9.239e+06	7.286e+04	2.025e+04
F14	1.497e+05	1.394e+03	1.559e+05	8.132e+03	2.674e+04	7.101e+03
F15	1.478e+07	2.559e+04	9.966e+07	9.104e+04	2.806e+04	2.162e+04
F16	2.747e+03	1.129e+03	2.538e+03	1.755e+03	1.542e+03	4.919e+02
F17	1.164e+03	4.421e+02	1.052e+03	5.735e+02	7.737e+02	1.409e+02
F18	1.867e+06	5.763e+04	3.956e+06	1.909e+05	1.285e+05	3.666e+05
F19	2.675e+07	8.268e+04	1.255e+08	6.364e+05	4.967e+05	1.070e+04
F20	5.997e+02	4.044e+02	7.617e+02	5.197e+02	6.821e+02	2.210e+02
F21	5.541e+02	3.932e+02	5.724e+02	4.533e+02	3.759e+02	2.336e+02
F22	4.476e+03	8.775e+02	6.955e+03	3.372e+03	4.836e+03	1.012e+02
F23	1.304e+03	9.776e+02	7.862e+02	8.299e+02	6.719e+02	3.960e+02
F24	1.453e+03	9.958e+02	8.187e+02	9.790e+02	6.447e+02	4.692e+02
F25	1.975e+03	4.166e+02	3.532e+03	6.018e+02	4.013e+02	4.016e+02
F26	6.824e+03	2.849e+03	6.699e+03	5.486e+03	3.926e+03	1.419e+03
F27	1.776e+03	1.076e+03	8.646e+02	8.604e+02	7.107e+02	5.322e+02
F28	3.225e+03	3.965e+02	4.454e+03	8.772e+02	3.240e+02	4.300e+02
F29	2.860e+03	1.514e+03	2.609e+03	2.025e+03	1.769e+03	6.153e+02
F30	3.229e+08	3.941e+05	3.110e+08	7.620e+06	5.686e+05	2.571e+04

Cuadro 5.2: Resultados medios tras el 100 % de las evaluaciones de la función objetivo para dimensión 30.

Finalmente, tenemos los resultados de la Tabla 5.3. Es interesante ver que la hibridación TLBO-SW consigue ser el mejor en una función más que en la dimensión 30. Las hibridaciones con Solis-Wets se han mostrado competitivas en las tres dimensiones estudiadas, mientras que el resto de mejoras no han sido capaces de destacar en ninguna función. Esto mues-

tra la importancia de la hibridación en algoritmos meméticos, pues han sido capaces de llevar un algoritmo incapaz de competir contra TLBO-FL a ganarle en varias ocasiones. En cuanto al resto de resultados, presentan los mismos patrones que los observados en las anteriores comparaciones.

	TLBO	TLBO-SW	TLBO-E	TLBO-B	TLBO-EBSW	TLBO-FL
F01	9.420e+10	1.938e+04	1.355e+11	1.507e+10	7.338e+02	6.062e+05
F02	4.313e+78	6.030e+79	8.175e+77	3.678e+65	6.152e+71	1.568e+39
F03	1.696e+05	2.341e+04	2.054e+05	9.478e+04	1.093e+05	2.569e+04
F04	2.925e+04	1.814e+02	3.668e+04	5.264e+03	1.548e+02	1.899e+02
F05	5.991e+02	3.133e+02	7.601e+02	4.638e+02	6.566e+02	9.670e+01
F06	8.740e+01	6.074e+01	9.982e+01	7.431e+01	7.617e+01	4.514e+00
F07	1.159e+03	9.012e+02	2.544e+03	9.737e+02	2.589e+03	1.745e+02
F08	6.235e+02	3.226e+02	7.919e+02	4.725e+02	5.489e+02	9.334e+01
F09	2.660e+04	1.235e+04	3.850e+04	1.847e+04	2.071e+04	1.298e+03
F10	1.233e+04	6.102e+03	1.356e+04	1.157e+04	7.281e+03	1.265e+04
F11	1.504e+04	2.113e+02	1.730e+04	2.314e+03	2.316e+02	1.687e+02
F12	5.953e+10	9.026e+06	4.804e+10	8.985e+09	4.496e+06	9.155e+05
F13	2.329e+10	1.098e+05	1.511e+10	5.371e+08	8.431e+04	8.010e+03
F14	2.419e+07	7.355e+03	5.077e+06	5.775e+05	2.241e+04	8.626e+04
F15	2.604e+09	3.015e+04	4.249e+09	8.787e+06	4.684e+04	6.883e+03
F16	5.074e+03	1.684e+03	4.964e+03	3.790e+03	2.500e+03	8.471e+02
F17	2.892e+03	1.420e+03	4.796e+03	1.979e+03	2.678e+03	8.532e+02
F18	3.190e+07	1.525e+05	4.389e+07	3.177e+06	3.885e+05	1.146e+06
F19	1.170e+09	7.846e+04	1.854e+09	2.513e+06	2.254e+05	1.447e+04
F20	1.503e+03	1.094e+03	1.937e+03	1.140e+03	1.573e+03	1.039e+03
F21	1.000e+03	6.441e+02	1.014e+03	7.658e+02	7.269e+02	2.814e+02
F22	1.280e+04	7.206e+03	1.396e+04	1.182e+04	8.007e+03	6.557e+03
F23	2.205e+03	1.715e+03	1.555e+03	1.526e+03	1.240e+03	5.658e+02
F24	2.541e+03	1.686e+03	1.522e+03	1.736e+03	1.150e+03	6.746e+02
F25	1.090e+04	5.959e+02	1.740e+04	2.402e+03	5.535e+02	6.159e+02
F26	1.328e+04	8.630e+03	1.560e+04	1.028e+04	8.463e+03	2.931e+03
F27	4.208e+03	2.599e+03	1.636e+03	1.905e+03	1.463e+03	8.677e+02
F28	8.239e+03	5.488e+02	9.887e+03	2.807e+03	4.956e+02	6.108e+02
F29	2.329e+04	3.264e+03	1.536e+04	4.837e+03	4.640e+03	1.017e+03
F30	3.185e+09	1.448e+07	4.191e+09	2.413e+08	2.242e+07	1.155e+06

Cuadro 5.3: Resultados medios tras el 100% de las evaluaciones de la función objetivo para dimensión 50.

5.2. RESULTADOS EN RANKINGS

La comparativa de *rankings* medios individuales muestran una notable diferencia entre las metaheurísticas base y los algoritmos propuestos, tal y como desvelan las tablas del Apéndice 5.2. Dichas tablas reflejan claramente la convergencia prematura de nuestro algoritmo, mostrando cómo generalmente es de los mejores modelos en las primeras iteraciones mientras que va perdiendo eficacia hacia las etapas finales de ejecución. Este patrón se repite en todas las versiones implementadas, lo que puede indicar que la falta de diversidad podría no ser el principal causante de la convergencia prematura. A raíz de los hechos, podría estudiarse si realmente la causa de tal problema es debido a la población en sí o a la acción del profesor. Es posible que un profesor caiga en un mínimo local que impida al resto hacer una correcta exploración del entorno.

Por otro lado, las implementaciones que muestran un mejor rendimiento relativo durante toda la ejecución son aquellas que incluyen mecanismos para aprovechar el sesgo hacia al origen de esta metaheurística: TLBO-B y TLBO-EBSW. Este hecho muestra que nuestra modificación respecto al sesgo en el origen ha sido eficaz. En particular, TLBO-EBSW es el que mejor rendimiento muestra de todas las implementaciones realizadas, sugiriendo que una explotación elitista y un reinicio adecuado de la población, si bien por si solos no han mostrado un gran desempeño, han logrado potenciar notablemente el rendimiento del modelo. Está claro que tanto TLBO como las modificaciones realizadas es peor que los tres algoritmos base, a excepción de las hibridaciones con Solis-Wets que en contadas ocasiones logra superar a PSO. Sin embargo, TLBO-FL resultados competentes frente a varios de ellos, tal y como muestra la Tabla 5.4.

	1	2	3	5	10	20	30	40	50	60	70	80	90	100
DE	3.833333	3.766667	3.733333	3.400000	2.50	2.300000	2.233333	2.166667	2.166667	2.133333	2.116667	2.083333	2.083333	2.10
GSKA	1.583333	1.616667	1.616667	1.683333	1.75	1.850000	1.883333	1.950000	1.950000	2.016667	2.033333	2.033333	2.066667	2.05
PSO	2.983333	2.950000	2.950000	3.150000	3.55	3.616667	3.583333	3.616667	3.583333	3.516667	3.483333	3.483333	3.450000	3.45
TLBO-FL	1.600000	1.666667	1.700000	1.766667	2.20	2.233333	2.300000	2.266667	2.300000	2.333333	2.366667	2.400000	2.400000	2.40

Cuadro 5.4: Comparativa de *rankings* medio para dimensión 10 entre DE, GSKA, PSO y TLBO-FL.

Estos resultados se ven mejorados para la dimensión 30, tal y como se ve en la Tabla 5.5. Aquí, durante la primera mitad de la ejecución TLBO-FL logra ser el mejor modelo de manera consistente, hasta que es sobrepasado por GSKA. Esto evidencia en cierta forma el problema de convergencia prematura que sufre este algoritmo y sus variantes, pues generalmente no logran mantener el ritmo en los tramos finales de ejecución.

	1	2	3	5	10	20	30	40	50	60	70	80	90	100
DE	3.733333	3.733333	3.666667	3.000000	2.800000	2.700000	2.633333	2.500000	2.433333	2.400000	2.433333	2.366667	2.300000	2.266667
GSKA	1.716667	1.783333	1.783333	1.816667	1.750000	1.683333	1.616667	1.683333	1.716667	1.716667	1.750000	1.816667	1.850000	1.916667
PSO	3.050000	3.083333	3.150000	3.683333	3.816667	3.850000	3.883333	3.850000	3.850000	3.850000	3.816667	3.816667	3.816667	3.783333
TLBO-FL	1.500000	1.400000	1.400000	1.500000	1.633333	1.766667	1.866667	1.966667	2.000000	2.033333	2.000000	2.000000	2.033333	2.033333

Cuadro 5.5: Comparativa de *rankings* medio para dimensión 30 entre DE, GSKA, PSO y TLBO-FL.

Finalmente, las tornas cambian si nos fijamos en la tabla Tabla 5.6. Aquí, TLBO-FL es claramente la mejor metaheurística. Esto nos puede mostrar dos claves interesantes en este tipo

de problemas: a menor dimensión, hay más probabilidad de convergencia prematura; sin embargo, cuando esta aumenta, un eficiente uso de la función objetivo puede ser crucial para realizar más generaciones y modificar sólo las soluciones necesarias.

	1	2	3	5	10	20	30	40	50	60	70	80	90	100
DE	3.766667	3.700000	3.500000	3.100000	2.833333	2.666667	2.766667	2.766667	2.700000	2.633333	2.600000	2.566667	2.433333	2.433333
GSKA	1.783333	1.850000	1.916667	1.883333	1.850000	1.850000	1.783333	1.816667	1.816667	1.850000	1.883333	1.883333	2.016667	2.016667
PSO	3.116667	3.083333	3.183333	3.616667	3.816667	3.850000	3.783333	3.750000	3.750000	3.750000	3.750000	3.750000	3.750000	3.716667
TLBO-FL	1.333333	1.366667	1.400000	1.400000	1.500000	1.633333	1.666667	1.666667	1.733333	1.766667	1.766667	1.800000	1.800000	1.833333

Cuadro 5.6: Comparativa de *rankings* medio para dimensión 50 entre DE, GSKA, PSO y TLBO-FL.

Este análisis realizado sobre TLBO-FL y los algoritmos base muestran un camino a seguir en futuras mejoras para TLBO. Las mejoras en exploración realizadas han mostrado no ser lo suficientemente buenas, por lo que podrían explorarse nuevas vías más allá de la interacción con un alumno y mutaciones aleatorias. Por otra parte, sería interesante optimizar las llamadas a la función objetivo, de forma que en dimensiones altas se garantice un mayor número de generaciones.

REFERENCIAS

- [1] ČREPINŠEK, M., LIU, S.-H., AND MERNIK, L. A note on teaching-learning-based optimization algorithm. *Information Sciences* 212 (2012), 79–93.
- [2] KOMMADATH, R., AND KOTTECHA, P. Teaching learning based optimization with focused learning and its performance on cec2017 functions. In *2017 IEEE Congress on Evolutionary Computation (CEC)* (2017), pp. 2397–2403.
- [3] PICKARD, J. K., CARRETERO, J. A., AND BHAVSAR, V. C. On the convergence and origin bias of the teaching-learning-based-optimization algorithm. *Applied Soft Computing* 46 (2016), 115–127.
- [4] RAO, R. V., AND PATEL, V. An improved teaching-learning-based optimization algorithm for solving unconstrained optimization problems. *Scientia Iranica* 20, 3 (2013), 710–720.
- [5] RAO, R. V., SAVSANI, V. J., AND VAKHARIA, D. Teaching-learning-based optimization: an optimization method for continuous non-linear large scale problems. *Information sciences* 183, 1 (2012), 1–15.
- [6] RAO, R. V., SAVSANI, V. J., AND VAKHARIA, D. P. Teaching-learning-based optimization: a novel method for constrained mechanical design optimization problems. *Computer-aided design* 43, 3 (2011), 303–315.
- [7] ZOU, F., CHEN, D., AND XU, Q. A survey of teaching-learning-based optimization. *Neurocomputing* 335 (2019), 366–383.

APÉNDICE

	1	2	3	5	10	20	30	40	50	60	70	80	90	100
DE	3.866667	3.800000	3.80	3.400000	2.366667	2.166667	2.066667	2.000000	2.000000	2.000000	1.983333	1.950000	1.950000	1.966667
GSKA	1.816667	1.650000	1.45	1.483333	1.516667	1.583333	1.550000	1.616667	1.650000	1.716667	1.733333	1.733333	1.766667	1.750000
PSO	2.983333	2.983333	2.95	3.083333	3.350000	3.250000	3.150000	3.116667	3.016667	2.883333	2.850000	2.716667	2.650000	2.650000
TLBO	1.333333	1.566667	1.80	2.033333	2.766667	3.000000	3.233333	3.266667	3.333333	3.400000	3.433333	3.600000	3.633333	3.633333

Cuadro .7: Comparativa de *rankings* medio para dimensión 10 entre DE, GSKA, PSO y TLBO.

	1	2	3	5	10	20	30	40	50	60	70	80	90	100
DE	3.766667	3.766667	3.666667	2.366667	2.133333	2.000000	1.966667	1.900000	1.866667	1.866667	1.866667	1.800000	1.766667	1.733333
GSKA	1.283333	1.183333	1.183333	1.216667	1.183333	1.216667	1.216667	1.283333	1.316667	1.283333	1.316667	1.383333	1.416667	1.483333
PSO	2.850000	2.750000	2.716667	3.150000	3.050000	3.050000	3.083333	3.050000	3.050000	3.016667	2.983333	2.950000	2.950000	2.916667
TLBO	2.100000	2.300000	2.433333	3.266667	3.633333	3.733333	3.733333	3.766667	3.766667	3.833333	3.833333	3.866667	3.866667	3.866667

Cuadro .8: Comparativa de *rankings* medio para dimensión 30 entre DE, GSKA, PSO y TLBO.

	1	2	3	5	10	20	30	40	50	60	70	80	90	100
DE	3.800000	3.766667	3.433333	2.433333	2.066667	1.933333	2.033333	2.033333	2.033333	1.966667	1.933333	1.933333	1.800000	1.800000
GSKA	1.183333	1.250000	1.316667	1.283333	1.316667	1.316667	1.283333	1.316667	1.316667	1.350000	1.383333	1.383333	1.516667	1.550000
PSO	2.683333	2.583333	2.616667	2.950000	3.050000	3.016667	2.950000	2.916667	2.883333	2.850000	2.850000	2.850000	2.816667	2.783333
TLBO	2.333333	2.400000	2.633333	3.333333	3.566667	3.733333	3.733333	3.733333	3.766667	3.833333	3.833333	3.833333	3.866667	3.866667

Cuadro .9: Comparativa de *rankings* medio para dimensión 50 entre DE, GSKA, PSO y TLBO.

	1	2	3	5	10	20	30	40	50	60	70	80	90	100
DE	3.700000	3.666667	3.666667	2.966667	2.300000	2.233333	2.200000	2.133333	2.133333	2.133333	2.116667	2.083333	2.083333	2.100000
GSKA	1.083333	1.150000	1.150000	1.316667	1.483333	1.616667	1.616667	1.650000	1.683333	1.750000	1.766667	1.800000	1.833333	1.816667
PSO	2.216667	2.283333	2.383333	2.783333	3.283333	3.383333	3.416667	3.350000	3.316667	3.250000	3.250000	3.250000	3.216667	3.216667
TLBO-SW	3.000000	2.900000	2.800000	2.933333	2.933333	2.766667	2.766667	2.866667	2.866667	2.866667	2.866667	2.866667	2.866667	2.866667

Cuadro .10: Comparativa de *rankings* medio para dimensión 10 entre DE, GSKA, PSO y TLBO-SW.

	1	2	3	5	10	20	30	40	50	60	70	80	90	100
DE	3.600000	3.666667	3.633333	2.500000	2.366667	2.233333	2.166667	2.166667	2.100000	2.066667	2.066667	2.000000	1.966667	1.933333
GSKA	1.216667	1.183333	1.183333	1.216667	1.250000	1.283333	1.250000	1.350000	1.416667	1.416667	1.450000	1.516667	1.550000	1.650000
PSO	2.216667	2.383333	2.550000	3.250000	3.616667	3.683333	3.716667	3.650000	3.616667	3.616667	3.583333	3.583333	3.583333	3.550000
TLBO-SW	2.966667	2.766667	2.633333	3.033333	2.766667	2.800000	2.866667	2.833333	2.866667	2.900000	2.900000	2.900000	2.900000	2.866667

Cuadro .11: Comparativa de *rankings* medio para dimensión 30 entre DE, GSKA, PSO y TLBO-SW.

	1	2	3	5	10	20	30	40	50	60	70	80	90	100
DE	3.700000	3.766667	3.533333	2.800000	2.600000	2.466667	2.566667	2.500000	2.466667	2.433333	2.400000	2.300000	2.133333	2.133333
GSKA	1.183333	1.250000	1.316667	1.316667	1.416667	1.483333	1.483333	1.516667	1.516667	1.550000	1.583333	1.583333	1.716667	1.750000
PSO	2.383333	2.550000	2.750000	3.316667	3.650000	3.716667	3.650000	3.616667	3.616667	3.583333	3.550000	3.550000	3.550000	3.516667
TLBO-SW	2.733333	2.433333	2.400000	2.566667	2.333333	2.333333	2.300000	2.366667	2.400000	2.433333	2.466667	2.566667	2.600000	2.600000

Cuadro .12: Comparativa de *rankings* medio para dimensión 50 entre DE, GSKA, PSO y TLBO-SW.

	1	2	3	5	10	20	30	40	50	60	70	80	90	100
DE	3.766667	3.666667	3.666667	3.00	2.266667	2.033333	2.000000	1.933333	1.933333	1.933333	1.916667	1.883333	1.883333	1.900000
GSKA	1.216667	1.183333	1.216667	1.25	1.350000	1.450000	1.483333	1.550000	1.583333	1.650000	1.666667	1.700000	1.733333	1.716667
PSO	2.783333	2.616667	2.583333	2.75	2.983333	2.950000	2.916667	2.850000	2.816667	2.716667	2.750000	2.716667	2.683333	2.683333
TLBO-E	2.233333	2.533333	2.533333	3.00	3.400000	3.566667	3.600000	3.666667	3.666667	3.700000	3.666667	3.700000	3.700000	3.700000

Cuadro .13: Comparativa de *rankings* medio para dimensión 10 entre DE, GSKA, PSO y TLBO-E.

	1	2	3	5	10	20	30	40	50	60	70	80	90	100
DE	3.600000	3.466667	3.300000	2.266667	2.033333	1.966667	1.933333	1.900000	1.866667	1.866667	1.866667	1.800000	1.766667	1.733333
GSKA	1.183333	1.150000	1.116667	1.150000	1.116667	1.150000	1.150000	1.216667	1.250000	1.250000	1.283333	1.350000	1.383333	1.450000
PSO	2.383333	2.383333	2.450000	2.950000	3.016667	3.050000	3.083333	3.050000	2.950000	2.916667	2.883333	2.883333	2.883333	2.850000
TLBO-E	2.833333	3.000000	3.133333	3.633333	3.833333	3.833333	3.833333	3.833333	3.933333	3.966667	3.966667	3.966667	3.966667	3.966667

Cuadro .14: Comparativa de *rankings* medio para dimensión 30 entre DE, GSKA, PSO y TLBO-E.

	1	2	3	5	10	20	30	40	50	60	70	80	90	100
DE	3.50	3.400000	3.033333	2.300000	1.966667	1.900000	2.000000	2.00	2.000000	1.966667	1.933333	1.933333	1.800000	1.800000
GSKA	1.15	1.183333	1.283333	1.183333	1.183333	1.216667	1.183333	1.25	1.216667	1.250000	1.283333	1.283333	1.416667	1.450000
PSO	2.45	2.416667	2.483333	2.816667	2.983333	2.950000	2.883333	2.85	2.850000	2.816667	2.816667	2.816667	2.816667	2.783333
TLBO-E	2.90	3.000000	3.200000	3.700000	3.866667	3.933333	3.933333	3.90	3.933333	3.966667	3.966667	3.966667	3.966667	3.966667

Cuadro .15: Comparativa de *rankings* medio para dimensión 50 entre DE, GSKA, PSO y TLBO-E.

	1	2	3	5	10	20	30	40	50	60	70	80	90	100
DE	3.866667	3.800000	3.766667	3.366667	2.466667	2.166667	2.133333	2.066667	2.066667	2.066667	2.050000	2.016667	2.016667	2.033333
GSKA	1.950000	1.950000	1.750000	1.583333	1.616667	1.616667	1.583333	1.616667	1.650000	1.716667	1.733333	1.766667	1.800000	1.783333
PSO	3.016667	3.016667	2.983333	3.183333	3.550000	3.550000	3.450000	3.316667	3.283333	3.150000	3.150000	3.116667	3.083333	3.083333
TLBO-B	1.166667	1.233333	1.500000	1.866667	2.366667	2.666667	2.833333	3.000000	3.000000	3.066667	3.066667	3.100000	3.100000	3.100000

Cuadro .16: Comparativa de *rankings* medio para dimensión 10 entre DE, GSKA, PSO y TLBO-B.

	1	2	3	5	10	20	30	40	50	60	70	80	90	100
DE	3.766667	3.766667	3.700000	3.000000	2.366667	2.200000	2.033333	1.966667	1.933333	1.900000	1.90	1.833333	1.800000	1.733333
GSKA	1.550000	1.283333	1.283333	1.316667	1.283333	1.216667	1.216667	1.283333	1.316667	1.316667	1.35	1.416667	1.450000	1.516667
PSO	3.050000	3.050000	3.083333	3.550000	3.683333	3.683333	3.683333	3.583333	3.483333	3.483333	3.35	3.283333	3.283333	3.250000
TLBO-B	1.633333	1.900000	1.933333	2.133333	2.666667	2.900000	3.066667	3.166667	3.266667	3.300000	3.40	3.466667	3.466667	3.500000

Cuadro .17: Comparativa de *rankings* medio para dimensión 30 entre DE, GSKA, PSO y TLBO-B.

	1	2	3	5	10	20	30	40	50	60	70	80	90	100
DE	3.800000	3.766667	3.600000	3.066667	2.500000	2.10	2.133333	2.100000	2.066667	2.00	1.933333	1.933333	1.800000	1.80
GSKA	1.283333	1.283333	1.350000	1.316667	1.316667	1.35	1.316667	1.316667	1.316667	1.35	1.383333	1.383333	1.516667	1.55
PSO	3.083333	3.016667	3.083333	3.516667	3.716667	3.65	3.516667	3.450000	3.383333	3.35	3.283333	3.250000	3.216667	3.15
TLBO-B	1.833333	1.933333	1.966667	2.100000	2.466667	2.90	3.033333	3.133333	3.233333	3.30	3.400000	3.433333	3.466667	3.50

Cuadro .18: Comparativa de *rankings* medio para dimensión 50 entre DE, GSKA, PSO y TLBO-B.

	1	2	3	5	10	20	30	40	50	60	70	80	90	100
DE	3.800000	3.700000	3.666667	3.200000	2.400000	2.166667	2.066667	2.000000	2.000000	2.000000	2.016667	1.983333	1.983333	2.000000
GSKA	1.516667	1.516667	1.450000	1.416667	1.516667	1.583333	1.616667	1.683333	1.716667	1.783333	1.766667	1.800000	1.833333	1.816667
PSO	2.850000	2.716667	2.716667	2.816667	3.216667	3.216667	3.050000	3.016667	2.983333	2.916667	3.083333	3.083333	3.083333	3.050000
TLBO-EBSW	1.833333	2.066667	2.166667	2.566667	2.866667	3.033333	3.266667	3.300000	3.300000	3.300000	3.133333	3.133333	3.100000	3.133333

Cuadro .19: Comparativa de *rankings* medio para dimensión 10 entre DE, GSKA, PSO y TLBO-EBSW.

	1	2	3	5	10	20	30	40	50	60	70	80	90	100
DE	3.733333	3.700000	3.533333	2.666667	2.40	2.233333	2.166667	2.066667	2.033333	2.00	2.000000	1.900000	1.866667	1.833333
GSKA	1.450000	1.316667	1.316667	1.283333	1.25	1.283333	1.250000	1.316667	1.350000	1.35	1.383333	1.450000	1.483333	1.550000
PSO	2.816667	2.816667	2.783333	3.350000	3.35	3.383333	3.516667	3.483333	3.483333	3.45	3.416667	3.416667	3.416667	3.383333
TLBO-EBSW	2.000000	2.166667	2.366667	2.700000	3.00	3.100000	3.066667	3.133333	3.133333	3.20	3.200000	3.233333	3.233333	3.233333

Cuadro .20: Comparativa de *rankings* medio para dimensión 30 entre DE, GSKA, PSO y TLBO-EBSW.

	1	2	3	5	10	20	30	40	50	60	70	80	90	100
DE	3.600000	3.566667	3.366667	2.766667	2.533333	2.300000	2.366667	2.333333	2.333333	2.300000	2.233333	2.200000	2.033333	2.033333
GSKA	1.383333	1.416667	1.483333	1.450000	1.450000	1.516667	1.416667	1.450000	1.450000	1.483333	1.483333	1.483333	1.583333	1.616667
PSO	2.783333	2.750000	2.883333	3.283333	3.450000	3.516667	3.450000	3.383333	3.383333	3.350000	3.350000	3.350000	3.350000	3.316667
TLBO-EBSW	2.233333	2.266667	2.266667	2.500000	2.566667	2.666667	2.766667	2.833333	2.833333	2.866667	2.933333	2.966667	3.033333	3.033333

Cuadro .21: Comparativa de *rankings* medio para dimensión 50 entre DE, GSKA, PSO y TLBO-EBSW.