

Análisis entre ganadores de las competiciones de optimización real del IEEE CEC: ¿Una mejora continua?

Daniel Molina¹, Francisco Moreno García², Francisco Herrera²

¹ Dept. of Computer Science
University of Cádiz, Spain
daniel.molina@uca.es

² Dept. of Computer Science and A.I.
University of Granada, Spain
fram@correo.uca.es, herrera@decsai.ugr.es

Abstract

Durante años se han organizado competiciones de optimización real en el congreso IEEE de Computación Evolutiva. En dichas competiciones los organizadores definen un *benchmark* y unas condiciones de experimentación que deberán seguir las propuestas enviadas, y se comparan los resultados obtenidos por cada una de ellas. Es una forma excelente de indicar qué algoritmos (y planteamientos) parecen mejorar al resto, pudiendo servir de guía para mejorar este campo. Desafortunadamente, al proponerse cada año un nuevo *benchmark*, los resultados entre competiciones no son comparables entre sí, por lo que hay garantía de que los ganadores de unos años sean mejores que los de años anteriores. En este trabajo presentamos una comparación entre ganadores de distintas competiciones usando los diferentes *benchmarks*, para analizar sus resultados y establecer qué algoritmos parecen los más competitivos en ellos, y averiguar si efectivamente se está produciendo un avance en las propuestas enviadas cada año.

1 Introducción

La optimización de parámetros reales es de gran interés para la comunidad científica, debido a la necesidad de optimizar que presentan un gran número de aplicaciones reales en campos como la ingeniería. En dichos problemas, en los que se puede realizar una búsqueda exhaustiva, se suelen aplicar algoritmos evolutivos [4], porque son algoritmos que, sin requerir información del problema, son capaces de obtener buenos resultados en un tiempo razonable.

Debido al interés, se proponen todos los años un gran número de propuestas, desde los originales Algoritmos Genéticos (*Genetic Algorithms*), GAs, como Estrategias de Evolución, o *Estimation of Distribution Algorithms*, EDAs, a otros más modernos como Evolución Diferencial (*Differential Evolution*), DE, muchos algoritmos inspirados en la naturales como las colonias de hormigas artificiales (*Artificial Ant Colony*), ACO, Sistemas de partículas (*Particle Swarm Optimization*), PSO, ... sin olvidar algoritmos híbridos como los algoritmos co-evolutivos o los algoritmos meméticos.

Para ver la evolución del campo y comparar las capacidades de optimización de los distintos algoritmos propuestos, en congresos internacionales como el *IEEE Congress on Evolutionary Computation*, en adelante IEEE CEC, se organizan sesiones especiales de optimización de parámetros reales. En éstos, se suele asociar una competición entre las propuestas usando un determinado *benchmark*. Los organizadores, tras comparar los resultados de las distintas propuestas entre sí, las ordenan y destacan los considerados *ganadoras*. Este enfoque es muy útil ya que estas conclusiones pueden servir para conocer los algoritmos (y las técnicas) que parecen obtener los mejores resultados, sirviendo de guía para obtener mejores algoritmos. Y decimos pueden porque al ser sobre un *benchmark* concreto no es seguro generalizar los resultados[16].

Sin embargo, al comparar únicamente en cada competición del IEEE CEC los algoritmos propuestos ese mismo año no se sabe su comportamiento respecto a los propuestos en años anteriores. A veces los algoritmos de un año sirven para proponer otro mejorado en la siguiente competición, pero en otras ocasiones no, y en ese caso un algoritmo ganador (y por tanto, competitivo) al no usarse de referencia

nunca más, no se sabe si podría ser peor que los nuevos, o llegar a mejorarlos (al usar *benchmarks* diferentes los resultados no son comparables).

Esto arroja una importante duda: *¿A través de los años, los ganadores de una competición obtienen mejores resultados que los de competiciones anteriores?* Es una pregunta crucial si queremos ver las competiciones como una muestra de la evolución del campo. Los investigadores suelen suponer que los últimos algoritmos publicados son mejores que los anteriores, aunque a veces las comparaciones con algoritmos anteriores no siempre usan buenos algoritmos de referencia [9]. En los congresos, que pueden reflejar una visión más actualizada de las tendencias del área, esta duda cobra mayor relevancia si cabe, ya que los algoritmos ganadores son bastante competitivos, y es importante saber si son superados o no por algoritmos posteriores.

En este trabajo nos proponemos discutir esta pregunta abierta, y observar si los ganadores de las competiciones más recientes mejoran o no a ganadores de competiciones anteriores. Para ello, nos centramos en los ganadores de las competiciones del CEC'2013, CEC'2014 y CEC'2015 (en el CEC'2016 no se propuso ningún *benchmark* nuevo, por lo que en ese caso los resultados si eran comparables con los de años anteriores). Para ello, evaluaremos los ganadores de cada año (junto con otros algoritmos de referencia) para cada uno de los tres *benchmarks*, y los compararemos todos entre sí. De esta manera, podemos observar si efectivamente se produce una evolución positiva en el rendimiento de los algoritmos propuestos durante estos años.

Este trabajo se estructura de la siguiente manera: En la Sección 2, describimos brevemente los algoritmos ganadores comparados. En la Sección 3, detallamos el modelo experimental seguido. En la Sección 4, analizamos los resultados obtenidos por los distintos algoritmos. Finalmente, en la Sección 5 destacamos las conclusiones y lecciones aprendidas más relevantes obtenidas.

2 Ganadores de las competiciones

En esta sección describimos muy brevemente los algoritmos considerados en este trabajo. Para comparar los algoritmos más prometedores de los últimos años, hemos considerado los tres ganadores de las competiciones de optimización del IEEE CEC, del CEC'2013, CEC'2014 el CEC'2015, junto con dos algoritmos de referencia ganadores de competiciones anteriores (CEC'2005 y CEC'2011). No se ha valorado las propuestas de la competición del CEC'2016 porque, al no proponerse un nuevo *benchmark* ese año, los resultados obtenidos si eran comparables con los de años anteriores. Además, el ganador de ese año, el UMOEAI [6] sí mejoró al ganador anterior, SPS-L-SHADE-EIG [10].

Para el CEC'2013 los ganadores fueron, en ese orden, ICMAES-ILS [17], NBIPOP-CMA [18], y DRMA [12]. Para el CEC'2014 el ganador claro fue L-SHADE [24], seguido de GaAPPADE [19], y MVMO-SH [7], al que denominaremos MVMO14 para evitar confundirlo con otras versiones del MVMO-SH de años posteriores). Del CEC'2015, los ganadores escogidos son SPS-L-SHADE-EIG [10] y LSHADE-ND [22]. DEsPA [3] no se consideró porque aunque globalmente se identificó como bueno, únicamente lo fue en dimensión 50. MVO [21], empatado como el tercer ganador, no pudo incluirse por problemas de rendimiento. Sin embargo, dado que era superado por los dos anteriores, su presencia no es necesaria para analizar la veracidad de la pregunta planteado. Además, se han incluido, como algoritmos de referencia, al ganador del CEC'2005, IPOP-CMA-ES [2], y al de la competición del CEC'2011 de optimización de problemas reales, GA-MPC [5].

En total, comparamos ocho algoritmos competitivos, seis de los cuales de las competiciones del CEC'2013, CEC'2014 y CEC'2015. A continuación describimos brevemente estos algoritmos, si desea conocer más detalle puede consultar la referencia indicada.

IPOP-CMA-ES [2] aplica el conocido algoritmo CMA-ES [11] en un enfoque multiarranque, incrementando el tamaño de la población tras cada reinicio. GAMPC [5] es un GA con un cruce de múltiples padres combinado con un archivo de soluciones para mayor diversidad. ICMAES-ILS [17] combina IPOP-CMA-ES con una búsqueda local iterativa, aplica ambos un cierto número de evaluaciones, y el que mejor comportamiento ofrezca es el que aplica el resto de evaluaciones. NBIPOP-CMA [18], usa el CMA-ES usando dos poblaciones, y para actualizar la matriz de covarianza usa también las soluciones

peores. DRMA [12] es un MA que combina GA usando CMA-ES como su búsqueda local [20] y divide el espacio de búsqueda en hipercubos con un tamaño que se reduce durante la búsqueda. GaAPPADe [19] usa tres subpoblaciones, cada una evoluciona por un algoritmo distinto (GA, DE y CMA-ES) un tiempo, y luego se aplica a las tres el algoritmo que esté dando mejores resultados (es un enfoque parecido al ICMAES-ILS). MVMO14 [7] es un MVMO con un método adicional de mejora y que cataloga a las soluciones en buenas y malas, que son mejoradas de forma diferente. L-SHADE [24], mejora al conocido SHADE [23], que adapta los parámetros F y CR, reduciendo el tamaño de la población durante la búsqueda. L-SHADE-ND [22] combina un L-SHADE (con pequeños cambios) con un método de optimización neuro-dinámico. Finalmente, SPS-L-SHADE-EIG [10] modifica L-SHADE reemplazando el algoritmo de cruce original y selecciona de forma diferente los padres cuando detecta convergencia prematura.

3 Condiciones de Experimentación

A continuación detallamos el proceso de experimentación seguido.

3.1 Implementación

Para poder comparar los algoritmos para los distintos benchmarks, necesitábamos una implementación de los algoritmos seleccionados. La implementación, en Java, están libremente disponible ¹. Para validar la implementación se compararon los resultados obtenidos por nuestra implementación con los resultados obtenidos por sus autores, y no se detectaron diferencias significativas entre ambos. En el caso de DRMA y SPS-L-SHADE-EIG, el código original fue utilizado para obtener los resultados.

Los parámetros usados son los indicados por sus autores. En CEC'2015, como se permitía, SPS-L-SHADE-EIG ha optimizado sus parámetros para dichas funciones. Esta optimización previa puede hacer que SPS-L-SHADE-EIG tenga peores resultados de lo debido en otros benchmarks.

3.2 Benchmark

Los *benchmarks* del IEEE CEC son bastante similares entre sí (a excepción del CEC'2011). Todos usan múltiples funciones, entre 25 y 30, de distinta complejidad. Las funciones están desplazadas para evitar que el óptimo esté en el centro, y así beneficiar a ciertos algoritmos. Se propone para dimensión 10, 30 y 50 (algunas incluyen 100, pero hemos preferido no usarlo para una comparación más homogénea). Y el criterio de parada es el máximo número de evaluaciones, que depende de la dimensión (normalmente es $10000 \cdot Dim$, donde Dim es la dimensión). Para más información se pueden consultar los siguientes trabajos: CEC'2013 [15], CEC'2014 [14], y CEC'2015 [13].

3.3 Metodología de comparación

Para cada benchmark se han seguido sus condiciones experimentales, obteniendo para cada propuesta, función y dimensión la media de las ejecuciones realizadas. Luego, al igual que los organizadores de las competencias hemos ordenamos, para cada función y dimension, los algoritmos por su media y calculado luego el ranking medio para cada dimensión.

En cada comparativa mostramos dos tablas, con los valores ranking medios, y las posiciones de los algoritmos. En dichas tablas, los algoritmos están ordenados por competencias, y dentro de cada competición por su posición en su competición (de mejor a peor). Se ha elegido ese orden para visualizar mejor si los nuevos algoritmos mejoran a los anteriores o no.

También se han aplicado test estadísticos, usando el software KEEL [1]. En particular, se han usado los test no-paramétricos, por ser más adecuados para estos benchmarks [8]. Tras detectar diferencias significativas (siempre ha habido) hemos aplicado el método *post-hoc* Holchberg, que compara en cada caso

¹ <https://github.com/framg/EvolutionaryAlgorithms.git>

cada algoritmo con el mejor (menor ranking medio) con un p-value adaptado para mantener controlado el error acumulado. Se incluyen las tablas generados por KEEL.

4 Análisis de los algoritmos por benchmark

A continuación mostramos el análisis *benchmark* por *benchmark*, y luego comentaremos unas conclusiones generales. Para cada *benchmark* los algoritmos se han comparado usando el ranking promedio, y luego se han aplicado test estadísticos para detectar diferencias significativas.

4.1 Resultados para el benchmark CEC'2013

Table 1: Resultados por ranking en el *benchmark* de CEC'2013

(a) Ranking Medio				(b) Posiciones de los algoritmos			
Algoritmo	D10	D30	D50	Algoritmo	D10	D30	D50
IPOP-CMAES	5.52	6.18	6.23	IPOP-CMAES	7	7	8
GAMPC	8.11	8.04	7.79	GAMPC	10	10	10
ICMAES-ILS	4.71	5.04	4.95	ICMAES-ILS	4	5	4
NBIPOPcCMA	6.68	6.77	6.80	NBIPOPcCMA	9	9	9
DRMA	5.12	5.30	5.64	DRMA	5	6	6
L-SHADE	4.32	4.27	4.16	L-SHADE	1	2.5	2
GAAPPADE	4.64	4.27	5.12	GAAPPADE	3	2.5	5
MVMO14	6.12	6.41	6.07	MVMO14	8	8	7
SPS-L-SHADE-EIG	4.52	4.70	4.34	SPS-L-SHADE-EIG	2	4	3
L-SHADE-ND	5.25	4.04	3.89	L-SHADE-ND	6	1	1

La Tabla 1 muestra el ranking medio y sus posiciones, para cada dimensión del *benchmark* CEC'2013. Se muestra que los mejores valores para las distintas dimensiones son L-SHADE (dimensión 10), y L-SHADE-ND (dimensión 30 y 50). Los resultados de SPS-L-SHADE-EIG y GAAPPADE también son bastante buenos. ICMAES-ILS, el ganador de esa competición, mantiene una posición media para cada valor de dimensión, al ser mejorado por los nuevos algoritmos.

La tabla 2 muestra los resultados de los test estadísticos. Para dimensión 10, sólo se detecta como estadísticamente peores a GAMPC y NBIPOPcCMA. Para dimensiones 30 y 50, no se detecta diferencias significativas entre el mejor, L-SHADE-ND, y los algoritmos L-SHADE, SPS-L-SHADE-EIG, ICMAES-ILS, DRMA, y GAAPPADE.

4.2 Resultados para el benchmark CEC'2014

La Tabla 3 muestra los algoritmos ordenados por sus ranking medios. Se puede observar que los mejores algoritmos son L-SHADE, L-SHADE-ND and ICMAES-ILS. L-SHADE, el ganador de esa competición, no ha sido mejorado por los algoritmos propuestos en el CEC'2015, aunque L-SHADE-ND mejora para dimensión 10. Mientras que L-SHADE y L-SHADE-ND son los mejores para dimensión 10 y 30, es CMAES-ILS el ganador para dimensión 50. SPS-L-SHADE-EIG obtiene buenos resultados, a diferencia del *benchmark* CEC'2013.

La Tabla 4 muestra los resultados de los test estadísticos. Para dimensión 10, el mejor es L-SHADE-ND y mejora significativamente a GAMP, NBIPOPcCMA, y GAAPPADE. Para dimensión 30, L-SHADE mejora a GAMPC, MVMO14, NBIPOPcCMA, y SPS-L-SHADE-EIG. Para dimensión 50, ICMAES-ILS mejora a GAMPC, MVMO14 and NBIPOPcCMA.

Table 2: Post Hoc comparación Hochberg para CEC'2013

(a) D=10, comparando con L-SHADE			(b) D=30, comparando con L-SHADE-ND		
Algoritmo	unadjusted p	$p_{Hochberg}$	Algoritmo	unadjusted p	$p_{Hochberg}$
GAMPC	0.000003	0.000026	GAMPC	0.000001	0.000007
NBIPOPcMA	0.003579	0.028636	NBIPOPcMA	0.000734	0.005874
MVMO14	0.02582	0.180741	MVMO14	0.003334	0.023341
IPOP-CMAES	0.139252	0.808197	IPOP-CMAES	0.008092	0.048552
L-SHADE-ND	0.251152	0.808197	DRMA	0.117149	0.585744
DRMA	0.320673	0.808197	ICMAES-ILS	0.216522	0.774197
ICMAES-ILS	0.627319	0.808197	SPS-L-SHADE-EIG	0.414197	0.774197
GAAPPADE	0.691197	0.808197	GAAPPADE	0.774197	0.774197
SPS-L-SHADE-EIG	0.808197	0.808197	L-SHADE	0.774197	0.774197

(c) D=50, comparando con L-SHADE-ND		
Algoritmo	unadjusted p	$p_{Hochberg}$
GAMPC	0.000002	0.000014
NBIPOPcMA	0.000322	0.002574
IPOP-CMAES	0.003841	0.026885
MVMO14	0.007095	0.042571
DRMA	0.030564	0.152821
GAAPPADE	0.127829	0.511318
ICMAES-ILS	0.192905	0.578714
SPS-L-SHADE-EIG	0.581148	0.740625
L-SHADE	0.740625	0.740625

Table 3: Resultados por ranking en el *benchmark* de CEC'2014

(a) Ranking Medio				(b) Posición de los algoritmos			
Algoritmo	D10	D30	D50	Algoritmo	D10	D30	D50
IPOP-CMAES	5.62	5.48	5.18	IPOP-CMAES	7	5	5
GAMP	7.67	7.50	8.03	GAMP	10	10	10
ICMAESILS	5.17	4.43	4.07	ICMAESILS	4	3	1
NBIPOPcMA	6.88	6.35	6.02	NBIPOPcMA	9	8	8
DRMA	5.23	5.53	5.65	DRMA	5	6	7
L-SHADE	4.18	3.93	4.27	L-SHADE	2	1	2
GAAPPADE	6.18	5.82	5.53	GAAPPADE	8	7	6
MVMO14	5.45	7.08	7.05	MVMO14	6	9	9
SPS-L-SHADE-EIG	4.57	4.65	4.57	SPS-L-SHADE-EIG	3	4	3
L-SHADE-ND	4.05	4.22	4.63	L-SHADE-ND	1	2	4

Table 4: Post Hoc comparación Hochberg para CEC'2014

(a) D=10, comparando con L-SHADE-ND			(b) D=30, comparando con L-SHADE		
Algoritmo	unadjusted p	$p_{Hochberg}$	Algoritmo	unadjusted p	$p_{Hochberg}$
GAMPC	0.000004	0.000033	GAMPC	0.000004	0.000033
NBIPOPcCMA	0.00029	0.002317	MVMO14	0.000056	0.000447
GAAPPADE	0.006353	0.044473	NBIPOPcCMA	0.001384	0.009687
IPOP-CMAES	0.045061	0.270365	SPS-L-SHADE-EIG	0.003737	0.022423
MVMO14	0.073312	0.366558	GAAPPADE	0.020132	0.100658
DRMA	0.130096	0.459492	IPOP-CMAES	0.036675	0.115907
ICMAES-ILS	0.153164	0.459492	DRMA	0.038636	0.115907
SPS-L-SHADE-EIG	0.508662	0.864569	ICMAES-ILS	0.405701	0.749119
L-SHADE	0.864569	0.864569	L-SHADE-ND	0.749119	0.749119

(c) D=50, comparando con ICMAES-ILS		
Algoritmo	unadjusted p	$p_{Hochberg}$
GAMPC	0.000005	0.000045
MVMO14	0.000056	0.000447
NBIPOPcCMA	0.001992	0.013945
GAAPPADE	0.015989	0.095934
DRMA	0.040685	0.189574
IPOP-CMAES	0.047393	0.189574
SPS-L-SHADE-EIG	0.359267	0.717022
ICMAES-ILS	0.522431	0.717022
L-SHADE-ND	0.717022	0.717022

4.3 Resultados para el benchmark CEC'2015

La Tabla 5 muestra los algoritmos ordenados por ranking. Se observa que el mejor algoritmo es GAAPPADE, y los siguientes mejores son SPS-L-SHADE-EIG y L-SHADE-ND.

Table 5: Resultados por ranking en el *benchmark* de CEC'2015

(a) Ranking Medio				(b) Posición ordenada de los algoritmos			
Algoritmos	D10	D30	D50	Algoritmo	D10	D30	D50
IPOP-CMAES	7.68	7.00	6.07	IPOP-CMAES	9	7	8
GAMPC	7.73	7.47	8.60	GAMPC	10	10	10
ICMAES-ILS	5.93	5.40	5.27	ICMAES-ILS	7	6	6
NBIPOPcCMA	7.67	7.20	5.93	NBIPOPcCMA	8	8	7
DRMA	5.10	5.30	4.67	DRMA	4	5	3
L-SHADE	5.17	5.23	5.23	L-SHADE	5	4	4.5
GAAPPADE	3.00	2.60	2.93	GAAPPADE	1	2	1
MVMO14	5.80	7.27	7.80	MVMO14	6	9	9
SPS-L-SHADE-EIG	3.30	2.37	3.27	SPS-L-SHADE-EIG	2	1	2
L-SHADE-ND	4.23	5.17	5.23	L-SHADE-ND	3	3	4.5

En la Tabla 6 se muestran los resultados del test estadístico. Para dimensiones 10 y 50 el mejor fue GAAPPADE, y solo para dimensión 30 SPS-L-SHADE-EIG. Es destacable este hecho, que GAAPPADE mejore para dimensión 10 y 50 al ganador de dicha competición, SPS-L-SHADE-EIG. Esta mejora es más interesante aún, si se valora que los parámetros de SPS-L-SHADE-EIG fueron especialmente

optimizados para este benchmark, a diferencia de GAAPPADE.

Table 6: Post Hoc comparación Hochberg para CEC'2015

(a) D=10, comparando con L-SHADE-ND			(b) D=30, comparando con SPS-L-SHADE-EIG		
Algoritmo	unadjusted p	$p_{Hochberg}$	Algoritmo	unadjusted p	$p_{Hochberg}$
GAMPC	0.000019	0.000167	GAMPC	0.000004	0.000036
NBIPOPcCMA	0.000024	0.000194	MVMO14	0.000009	0.000075
IPOP-CMAES	0.000235	0.001643	NBIPOPcCMA	0.000012	0.000086
ICMAES-ILS	0.007971	0.047825	IPOP-CMAES	0.000028	0.000167
MVMO14	0.011319	0.056595	ICMAES-ILS	0.006074	0.022638
L-SHADE	0.050016	0.172488	DRMA	0.007971	0.022638
DRMA	0.057496	0.172488	L-SHADE	0.009514	0.022638
L-SHADE-ND	0.264597	0.529194	L-SHADE-ND	0.011319	0.022638
SPS-L-SHADE-EIG	0.786114	0.786114	GAAPPADE	0.832842	0.832842

(c) D=50, comparando con GAAPPADE		
Algoritmo	unadjusted p	$p_{Hochberg}$
GAMPC	0	0.000003
MVMO14	0.000011	0.000086
IPOP-CMAES	0.004594	0.032158
NBIPOPcCMA	0.006656	0.039934
ICMAES-ILS	0.034808	0.112459
L-SHADE	0.037486	0.112459
L-SHADE-ND	0.037486	0.112459
DRMA	0.116914	0.233828
SPS-L-SHADE-EIG	0.763025	0.763025

4.4 Resultados Generales

Una vez que hemos detectado los algoritmos con mejores resultados, podemos obtener una serie de interesantes conclusiones, sin orden concreto:

El algoritmo GAMP obtiene en la mayoría de casos los peores resultados. Una posible razón es que al ser tan diferente el benchmark del CEC'2011 del resto, el GAMP, siendo muy bueno en ese *benchmark*, obtiene un comportamiento pobre en el resto.

L-SHADE y L-SHADE-ND son los algoritmos más robustos, obteniendo los mejores resultados en más benchmarks. DRMA y ICMAES-ILS ofrecen buenos resultados, sin que se detectase en la mayoría de casos ser significativamente peores que el mejor. GAAPPADE es el algoritmo con mejores resultados en el CEC'2015 *benchmark*, siendo de un año anterior.

SPS-L-SHADE-EIG obtiene muy buenos resultados en CEC'2015 pero no tanto en otras competiciones, aún siendo buenos. Sus parámetros optimizados le permiten obtener los mejores resultados en el CEC'2015 pero no así en el resto.

Desde un punto de vista estadístico, no hay diferencias significativas entre L-SHADE, SPS-L-SHADE-EIG, L-SHADE-ND, ICMAES-ILS y DRMA en la mayoría de las comparativas, por lo que podemos considerarlos algoritmos bastante robustos. Los otros algoritmos detectados han sido estadísticamente peor en alguna ocasión, y los algoritmos como IPOP-CMAES, NBIPOPcCMA pueden considerarse como ya mejorados al cabo de los años, mostrando una mejora desde su propuesta.

ICMAES-ILS, ganador del CEC'2013 ha conseguido los mejores resultados en el *benchmark* de CEC'2014 para dimensión 10, y GAAPPADE para el CEC'2015, mostrando que son algoritmos competitivos a seguir teniendo en cuenta.

5 Conclusiones y Lecciones Aprendidas

En los últimos años se han ido proponiendo distintas competiciones de optimización de parámetros reales. En cada una se evaluaban las propuestas usando un *benchmark* ordenando las propuestas por su resultado, obteniendo unos ganadores. Sin embargo, al considerar únicamente los algoritmos presentados en ese año, no se puede afirmar que los ganadores de una competición hayan sido mejorados por las siguientes, ya que no se ha comprobado. Para comprobarlo, hemos comparado los ganadores de las competiciones del IEEE del 2013, 2014 y 2015 con los distintos *benchmarks*.

Hemos aprendido que, aunque los resultados obtenidos por un *benchmark* no son generalizables, al comparar distintos *benchmarks* podemos identificar la existencia de algoritmos bastante robustos. Además, al aplicar los test estadísticos, hemos obtenido un grupo de algoritmos (L-SHADE-ND, L-SHADE, GAAPPADE, ICMAES-ILS, DRMA) con los que no se detectamos que el mejor los mejorase significativamente.

También hemos aprendido que no siempre los algoritmos con mejor resultados en un *benchmark* son los propuestos para dicho *benchmark*. En respuesta a la pregunta formulada anteriormente, *¿A través de los años, los ganadores de una competición obtienen mejores resultados que los de competiciones anteriores?* hemos visto que no es siempre el caso. Los resultados del ICMAES-ILS y GAAPPADE muestran que ganadores previos puede mejorar a los nuevos en las nuevas competiciones. Por tanto, a veces son injustamente ignorado a pesar de ser competitivos (GAAPPADE es el mejor algoritmo para CEC'2015 y ICMAES-ILS lo es para CEC'2014 para dimensión 50).

Por tanto, recomendamos a los organizadores de futuras competiciones, como lección aprendida, que incluyan al ganador (o ganadores) de competiciones previas como algoritmos de referencia de las nuevas. De esta forma, se podría comprobar que las nuevas propuestas mejoran a las anteriores en el nuevo *benchmark*. Esto se hace en algunas competiciones, como la de alta dimensionalidad (con una propuesta invicta varios años), pero su uso no es tan generalizado como debería de ser.

Agradecimientos

Este trabajo ha sido apoyado por los proyectos nacionales TIN2014-57251-P y TIN2016-81113-R.

References

- [1] J. Alcalá-Fdez, A. Fernández, J. Luengo, J. Derrac, and S. García. Keel data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework. *Multiple-Valued Logic and Soft Computing*, 17(2-3):255–287, 2011.
- [2] A Auger and N Hansen. A Restart CMA Evolution Strategy with Increasing Population Size. In *2005 IEEE Congress on Evolutionary Computation*, pages 1769–1776, 2005.
- [3] N. Awad, M. Z. Ali, and R. G. Reynolds. A differential evolution algorithm with success-based parameter adaptation for CEC2015 learning-based optimization. In *2015 IEEE Congress on Evolutionary Computation (CEC)*, pages 1098–1105, May 2015.
- [4] T. Bäck, D. B. Fogel, and Z. Michalewicz, editors. *Handbook of Evolutionary Computation*. IOP Publishing Ltd., Bristol, UK, 1997.
- [5] S. M. Elsayed, R. A. Sarker, and D. L. Essam. Ga with a new multi-parent crossover for solving iee-cec2011 competition problems. In *2011 IEEE Congress of Evolutionary Computation (CEC)*, pages 1034–1040, June 2011.
- [6] S. M. Elsayed, R. A. Sarker, D. L. Essam, and N. M. Hamza. Testing united multi-operator evolutionary algorithms on the cec2014 real-parameter numerical optimization. In *2014 IEEE Congress on Evolutionary Computation (CEC)*, pages 1650–1657, July 2014.

- [7] I. Erlich, J. L. Rueda, S. Wildenhues, and F. Shewarega. Evaluating the mean-variance mapping optimization on the ieeec-ec 2014 test suite. In *2014 IEEE Congress on Evolutionary Computation (CEC)*, pages 1625–1632, July 2014.
- [8] S. Garcia, D. Molina, M. Lozano, and F. Herrera. A study on the use of non-parametric tests for analyzing the evolutionary algorithms’ behaviour: A case study on the CEC’2005 Special Session on Real Parameter Optimization. *Journal of Heuristics*, 15(6):617–644, 2009.
- [9] C. García-Martínez, P.D. Gutiérrez, D. Molina, and F. Herrera. Since cec 2005 competition on real-parameter optimisation: a decade of research, progress and comparative analysis’s weakness. *Soft Computing*, pages 1–11, 2017.
- [10] S. M. Guo, J. S. H. Tsai, C. C. Yang, and P. H. Hsu. A self-optimization approach for L-SHADE incorporated with eigenvector-based crossover and successful-parent-selecting framework on CEC 2015 benchmark set. In *2015 IEEE Congress on Evolutionary Computation (CEC)*, pages 1003–1010, May 2015.
- [11] N Hansen, S D Müller, and P Koumoutsakos. Reducing the Time Complexity of the Derandomized Evolution Strategy with Covariance Matrix Adaptation (CMA-ES). *Evolutionary Computation*, 1(11):1–18, 2003.
- [12] B. Lacroix, D. Molina, and F. Herrera. Dynamically updated region based memetic algorithm for the 2013 cec special session and competition on real parameter single objective optimization. In *2013 IEEE Congress on Evolutionary Computation*, pages 1945–1951, June 2013.
- [13] J.J. Liang, Q. Chen, B.Y. Qu, B. Liu, P.N. Suganthan, and Q. Chen. Problem definitions and evaluation criteria for the cec 2015 competition on learning-based real-parameter single objective optimization. Technical report, Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou, China and Technical Report, Nanyang Technological University, Nov 2014.
- [14] J.J. Liang, B-Y. Qu, and P.N. Suganthan. Problem Definitions and Evaluation Criteria for the CEC 2014 Special Session and Competition on Single Objective Real-Parameter Numerical Optimization. Technical report, Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou, China and Technical Report, Nanyang Technological University, Singapore, 2013.
- [15] J.J. Liang, B-Y. Qu, P.N. Suganthan, and A.G. Hernández-Díaz. Problem definitions and evaluation criteria for the cec 2013 special session and competition on real-parameter optimization. Technical report, Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China and Technical Report, Nanyang Technological University, Singapore, 2013.
- [16] T. Liao, D. Molina, and T. Stützle. Performance evaluation of automatically tuned continuous optimizers on different benchmark sets. *Applied Soft Computing Journal*, 27:490–503, 2015.
- [17] T. Liao and T. Stützle. Benchmark results for a simple hybrid algorithm on the cec 2013 benchmark set for real-parameter optimization. In *2013 IEEE Congress on Evolutionary Computation*, pages 1938–1944, June 2013.
- [18] I. Loshchilov. Cma-es with restarts for solving cec 2013 benchmark problems. In *2013 IEEE Congress on Evolutionary Computation*, pages 369–376, June 2013.
- [19] R. Mallipeddi, G. Wu, M. Lee, and P. N. Suganthan. Gaussian adaptation based parameter adaptation for differential evolution. In *2014 IEEE Congress on Evolutionary Computation (CEC)*, pages 1760–1767, July 2014.
- [20] D. Molina, M. Lozano, C. García-Martínez, and F. Herrera. Memetic algorithms for continuous optimisation based on local search chains. *Evolutionary Computation*, 18(1):27–63, 2010.

- [21] J. L. Rueda and I. Erlich. Testing MVMO on learning-based real-parameter single objective benchmark optimization problems. In *2015 IEEE Congress on Evolutionary Computation (CEC)*, pages 1025–1032, May 2015.
- [22] K. M. Sallam, R. A. Sarker, D. L. Essam, and S. M. Elsayed. Neurodynamic differential evolution algorithm and solving CEC2015 competition problems. In *2015 IEEE Congress on Evolutionary Computation (CEC)*, pages 1033–1040, May 2015.
- [23] R. Tanabe and A. Fukunaga. Evaluating the performance of shade on cec 2013 benchmark problems. In *2013 IEEE Congress on Evolutionary Computation*, pages 1952–1959, June 2013.
- [24] R. Tanabe and A. S. Fukunaga. Improving the search performance of shade using linear population size reduction. In *2014 IEEE Congress on Evolutionary Computation (CEC)*, pages 1658–1665, July 2014.