Análisis Empírico del *No Free Lunch* en Problemas Binarios Reales

Carlos García-Martínez, Francisco J. Rodriguez, Manuel Lozano

Resumen— **Los teoremas del *No Free Lunch* para optimización (*no hay comida gratis*, en español) establecen que el rendimiento de cualquier par de algoritmos es equivalente cuando se considera el rendimiento medio sobre el conjunto completo de posibles problemas. Sin embargo, estudios más recientes sugieren que los algoritmos pueden ofrecer rendimientos diferentes cuando se consideran únicamente el conjunto de problemas reales, es decir, aquellos con interés práctico. En este trabajo, hacemos la suposición de que los problemas binarios que se han tratado en la literatura científica representan, al menos mínimamente, los verdaderos problemas binarios de interés en el mundo real. A partir de ahí, analizamos el rendimiento medio de varios algoritmos en subconjuntos de este banco de pruebas. En particular, consideramos problemas de optimización combinatoria estáticos sin restricciones de un solo objetivo y un solo jugador y cuyas soluciones pueden codificarse directamente como vectores de variables binarias. Nuestros resultados muestran que existe evidencia empírica para rechazar los teoremas del No Free Lunch en el conjunto de problemas binarios del mundo real.**

Palabras clave— **No free lunch, optimización binaria, estudio empírico, problemas reales**

I. Introducción

En 1997, Wolpert y Macready presentaron los teoremas para optimización del No Free Lunch (NFL) [1], los cuales establecen, en pocas palabras, que todos los algoritmos tienen un rendimiento equivalente cuando se consideran todos los posibles problemas, o más concretamente, conjuntos de problemas cerrados bajo permutaciones (c.b.p) [2] (Es posible reducir aún más el conjunto de problemas cuando se considera un conjunto concreto de algoritmos [3]). Es importante destacar que en este contexto se entiende por algoritmo cualquier procedimiento que realiza un muestreo sin reemplazamiento de las soluciones candidatas al problema. Este resultado impactó profundamente a investigadores que pretendían diseñar procedimientos eficientes de propósito general para problemas de optimización, porque se demostraba que no serían mejores que la búsqueda aleatoria sin reemplazamiento. Recientemente, Marshall y Hinton han demostrado que cuando se consideran procedimientos que pueden visitar soluciones más de una vez los teoremas del NFL no pueden aplicarse, sin embargo, aún existe una elevada probabilidad de que el rendimiento de cualquier algoritmo no sea superior al de la búsqueda aleatoria con o sin reemplazamiento [4].

En 2003, Igel y Toussaint mostraron que los teoremas del NFL posiblemente no tienen validez cuando se consideran conjuntos de problemas de interés práctico [5]. En particular, demostraron que los conjuntos de problemas a los que se les pueden asociar restricciones basadas en relaciones de similitud no triviales entre soluciones no son c.b.p. Según los autores, este tipo de restricciones son muy comunes en problemas de interés práctico, lo cual permite concebir procedimientos de propósito general para la optimización de problemas reales. En 2010, Jiang y Chen aportaron resultados empíricos en favor de las afirmaciones de Igel y Toussaint [6] (aunque el trabajo no se citaba), considerando una clase específica de funciones matemáticamente definidas que incorporaban restricciones basadas en la similitud de las soluciones. Sin embargo, la asunción de que los problemas reales incorporaban las mencionadas restricciones se mantenía aún sin demostrar.

En este trabajo, pretendemos aportar evidencias empíricas a favor de las afirmaciones de Igel y Toussaint en el conjunto específico de problemas con interés en el mundo real. Para ello, hacemos la siguiente suposición: los problemas tratados en la literatura científica representan, superando al menos un umbral mínimo, a los problemas del mundo real. En caso contrario, debe entenderse que los estudios científicos correspondientes habrían sido inútiles. A partir de ahí, seleccionamos un conjunto extenso, potencialmente infinito, de instancias de problemas binarios representativo de los que se han tratado en la literatura, y para el cual no se conoce si el NFL se cumple o no (entiéndase que el conjunto completo de posibles problemas binarios sí es c.b.p). Finalmente, realizamos un estudio empírico comparando varios algoritmos en el banco de problemas considerado con la intención de determinar si existen o no diferencias de rendimiento significativas entre ellos.

El resto del trabajo se estructura de la siguiente forma. En la Sección II, describimos el marco experimental bajo el cual desarrollamos nuestro estudio. En la Sección III, presentamos los resultados del estudio. En la Sección IV interpretamos las implicaciones de los resultados obtenidos. Finalmente, la Sección V ofrece las conclusiones del trabajo.

II. Marco Experimental

En esta sección, definimos el marco experimental en el que se realizan los experimentos. En la Sec-

Dpto. Informática y Análisis Numérico. Universidad de Córdoba. E-mail: cgarcia@uco.es .

Dpto. Ciencias de la Computación e I.A (CITIC-UGR). Universidad de Granada. E-mail: fjrodriguez@decsai.ugr.es .

Dpto. Ciencias de la Computación e I.A (CITIC-UGR). Universidad de Granada. E-mail: lozano@decsai.ugr.es .

ción II-A y II-B describimos el conjunto de problemas y algoritmos considerados en este estudio, respectivamente. La Sección II-C define la metodología de comparación.

A. Problemas

En esta sección, presentamos el conjunto de problemas que usamos para alcanzar nuestro objetivo, ofrecer evidencia empírica a favor de las afirmaciones de Igel y Toussaint [5]. En este punto, debemos asumir la hipótesis que dice que los problemas tratados en la literatura representan al menos mínimamente el verdadero conjunto de problemas con interés práctico en el mundo real. El lector debe entender que esta hipótesis, aunque no se haya probado, es completamente necesaria. El argumento opuesto establece que los problemas tratados en la literatura no representan a los problemas del mundo real a ningún nivel significativo, y por tanto, la investigación hasta el momento habría sido simplemente un juego desafiante, pero inútil. Hay que mencionar que en la literatura se han presentado ejemplos de problemas artificialmente construidos para confirmar las implicaciones del NFL y que carecen de significado e incluso definición fuera de este contexto. Su falta de definición ha impedido que se consideren en este estudio.

En este trabajo, hemos considerado un subconjunto restringido y bien definido de problemas de la literatura, en particular, *problemas de optimización combinatoria sin restricciones de un solo objetivo y jugador cuyas soluciones pueden representarse directamente como vectores de variables binarias* (en adelante, simplemente *problemas binarios*). Antes de nada, es importante aclarar las características del conjunto de problemas considerado:

- *Estático*: Significa que el problema permanece intacto a lo largo de la ejecución del procedimiento optimizador, es decir, el valor objetivo de las soluciones candidatas al problema no depende del tiempo o de la aplicación del algoritmo.
- *Optimización combinatoria binaria*: El espacio de búsqueda se compone de combinaciones de valores para las verdaderas variables de decisión del problema, las cuales pueden tomar únicamente los valores del conjunto $\{0, 1\}$. De esta definición, debe entenderse que problemas de optimización entera o de parámetros reales, cuyas variables de decisión no son binarias, no se consideran incluso a pesar de que éstas se traten como tal al nivel del computador o existiesen transformaciones polinomiales entre éstos y problemas de optimización binaria.
- *Un solo objetivo*: Sólo hay un criterio que debe maximizarse o minimizarse. Sin pérdida de generalidad, consideraremos problemas a maximizar, reformulando aquellos que sean de minimización.
- *Un solo jugador*: El agente que aborda el problema controla todas las variables de decisión para las cuales el problema interpreta una única solución. En contra, el dilema del prisionero [7] es un ejemplo de problema de más de un jugador donde el resultado depende de la decisión de más de un agente. Aunque los problemas de múltiples jugadores se puedan tratar gestionando conjuntamente las variables de decisión de todos los jugadores, éstos no se han considerado para mantener el estudio y análisis de los resultados lo más simple posible.
- *Sin restricciones*: Todas las soluciones del espacio de búsqueda son factibles. Por el contrario, problemas como el de la mochila, para el cual una fracción del espacio de búsqueda se compone de soluciones no factibles, no se consideran.

Hemos seleccionado un conjunto extenso, potencialmente infinito, de problemas binarios de la literatura, asumiendo que éstos representan, al menos mínimamente, el conjunto de problemas binarios (estáticos, sin restricciones...) del mundo real. La Tabla I enumera las clases de problemas considerados: sus nombres, referencias a su definición detallada, dimensiones de las instancias consideradas, es decir, número de variables binarias (L), y una aproximación al número potencial de instancias diferentes que podrían generarse. A continuación, proveemos algunos comentarios sobre estos problemas:

- Cada simulación de los algoritmos se realiza seleccionando al inicio la clase de problemas a abordar ($\{1, \ldots, 10\}$), y después, la instancia particular se genera o se selecciona aleatoriamente de un banco de pruebas para el caso de Maxcut, BQP o Un-knapsack. Como se indica, los problemas *deceptive* (en español, *engañosos*) ($\{2.a, 2.b, 2.c, 2.d\}$) se han agrupado en una sola clase. Es importante saber que la i-ésima simulación de cada algoritmo abordará exactamente la misma instancia del mismo problema. Esto es así porque la semilla del generador de números aleatorios que produce la instancia del problema es la misma para la i-ésima simulación de todos los algoritmos y probabilísticamente diferente a la semilla de la j-ésima simulación de éstos con $i \neq j$.
- Para problemas en los que el óptimo global tiene el valor 1 en todas las variables binarias ($\{1, 2.a, 2.b, 2.c, 2.d, 6, 7\}$), crearemos primeramente una cadena binaria aleatoria S que se convertirá en el nuevo óptimo, para aplicar una traslación al problema. El valor objetivo de una solución candidata X se calculará como el valor objetivo de la solución $\overline{X \oplus S}$ (es decir, el resultado de aplicar las operaciones a nivel de bit: primero O exclusivo entre X y S y después inversión) en el problema original. Esta es la razón, entre otras, de que el número potencial de instancias del problema *onemax* sea $\sum_L 2^L$. La intención es evitar la ventaja parcial de algunos algoritmos que tienden a evaluar la solución con sólo unos o sólo ceros al inicio de la simulación.
- Para evitar un tiempo de computación excesivo, algunas instancias se han reducido expresamente:
  - Si el problema *Nk-land* generado tiene $k \geq 6$ y $L > 100$, entonces, L se vuelve a asignar escogiendo aleatoriamente un número del conjunto

TABLA I
PROBLEMAS BINARIOS CONSIDERADOS

Problema	Nombre	Ref.	L	Instancias
1	Onemax	[8]	$\{20,\ldots,1000\}$	$\sum_L 2^L$
2.a	Simple deceptive	[9]	$\{20,\ldots,1000\},\ L \equiv 0 \pmod 3$	$\sum_L 2^L$
2.b	Trap	[10]	$\{20,\ldots,1000\},\ L \equiv 0 \pmod{36}$	$\sum_L 2^L$
2.c	Overlap deceptive	[11]	$\{20,\ldots,1000\},\ L \equiv 1 \pmod 2$	$\sum_L 2^L$
2.d	Bipolar deceptive	[11]	$\{20,\ldots,1000\},\ L \equiv 0 \pmod 6$	$\sum_L 2^L$
3	Max-sat	[12]	$\{20,\ldots,1000\}$	$< \sum_{L,C_l,N_c} (2L)^{C_l \cdot N_c}$, $C_l \in \{3,\ldots,6\}$, $N_c \in \{50,\ldots,500\}$
4	NK-land	[13]	$\{20,\ldots,500\}$	$<< \sum_{L,k,r} r^{(2^k)\cdot L}$ $r \in [0,1]$ $k \in \{2,\ldots,10\}$
5	PPeaks	[13]	$\{50,\ldots,500\}$	$< \sum_{L,N_p} 2^{L \cdot N_p}$ $N_p \in \{10,\ldots,200\}$
6	Royal-road	[14]	$\{20,\ldots,1000\},\ L \equiv 0 \pmod{L_{BB}}$ $L_{BB} \in \{4,\ldots,15\}$	$\sum_{L,L_{BB}} 2^L$
7	HIFF	[15]	$\{4.\ldots,1024\},\ L = k^p$ $k \in \{2,\ldots,5\},\ p \in \{4,5\}$	$\sum_L L! \cdot 2^L$
8	Maxcut	[16]	$\{60,\ldots,400\}$	178
9	BQP	[17]	$\{20,\ldots,500\}$	165
10	Un-knapsack	[18]	$\{10,\ldots,105\}$	54

$\{20,\ldots,100\}$.

- Si el problema *PPeaks* generado tiene $NP \geq 100$, entonces, L se vuelve a generar aleatoriamente dentro del conjunto $\{50,\ldots,100\}$
- Si el problema *HIFF* tiene $k = 5$ y $p = 5$, entonces, p se asigna igual a 4.

▪ Las instancias de los problemas *BQP* y *Maxcut* se obtuvieron de la *Biq Mac Library* (http://biqmac.uniklu.ac.at/).

▪ Las instancias del problema de la mochila con penalización (*Un-knapsack*) se obtuvieron de la SAC-94 Suite (http://elib.zib.de/pub/Packages/ /mp-testdata/ip/sac94-suite/). Estos problemas no tienen restricciones porque consideran un mecanismo de penalización concreto y explícito en su publicación original [18].

▪ Debemos avanzar que, aunque el conjunto completo de posibles problemas binarios (estáticos, sin restricciones...) es c.b.p, el subconjunto considerado de problemas que se han tratado en la literatura no lo es (lo cual se demuestra para algunos de los problemas en [5] y [19]). Esto significa que, mientras que el NFL es válido para el conjunto completo de problemas, éste puede no serlo en nuestro banco de problemas. De todas formas, es importante analizar su viabilidad en nuestro conjunto extenso de casos de prueba que representa al menos mínimamente el conjunto de problemas binarios con interés en el mundo real.

B. Algoritmos

Hemos seleccionado cinco algoritmos bien conocidos y con claras diferencias entre ellos para analizar si el NFL es válido en el conjunto de pruebas considerado.

Es importante destacar que no es nuestra intención presentar una propuesta ganadora o identificar el mejor algoritmo para problemas de optimización combinatoria binaria, sino encontrar evidencia empírica a favor o en contra de los teoremas del NFL. Por ello, los parámetros de los algoritmos no se han ajustado en ningún momento, ni se han considerado diseños avanzados de los algoritmos clásicos que recientemente hayan mostrado una clara superioridad con respecto a sus predecesores. En vez de eso, hemos considerado algoritmos estándares y bien conocidos con la asignación de valores razonables a sus correspondientes parámetros. Los algoritmos son:

▪ *Búsqueda Aleatoria* (BA): Ésta simplemente muestrea soluciones del espacio de búsqueda (con reemplazamiento) y devuelve la mejor solución encontrada. Cuando el espacio de búsqueda es suficientemente grande, BA no suele generar las mismas soluciones más de una vez en simulaciones limitadas, es decir, es similar a una búsqueda sin reemplazamiento. Por ello, no se deben esperar diferencias de rendimiento entre BA y cualquier otro algoritmo en conjuntos de problemas c.b.p., siempre que la probabilidad de generar soluciones más de una vez sea baja.

▪ *Búsqueda Local con Multi-arranque* (BLM): Los

algoritmos de búsqueda local explotan la idea de vecindario para mejorar iterativamente una solución dada. Éstos se utilizan muy frecuentemente en problemas de optimización combinatoria. En el caso de problemas binarios, la estructura de vecindario más utilizada es la que se produce cambiando el valor de una única variable binaria, es decir, dos soluciones binarias son vecinas si una solución puede obtenerse cambiando el valor de una variable de la otra. En este estudio aplicaremos la estrategia *primer-mejor* que explora el vecindario de la solución actual hasta encontrar la primera solución que la mejora. BLM aplicará iterativamente búsqueda local a soluciones aleatorias del problema cada vez que éstas queden atrapadas en óptimos locales. Al final de la ejecución, se devuelve la mejor solución encontrada.

- *Enfriamiento Simulado* (ES) [20]: Se considera que ES es el primer algoritmo que extiende los métodos de búsqueda local con una estrategia explícita para escapar de óptimos locales. La idea fundamental es permitir decisiones que provoquen una reducción en la calidad de la solución actual para escapar de estos óptimos locales. La probabilidad de tomar estas decisiones se controla mediante un parámetro que decrece durante la ejecución y que simula la temperatura del sistema. En este estudio hemos aplicado un ES estándar con el método de aceptación logístico y esquema de enfriamiento geométrico. La temperatura se reduce un 1 % cada cien iteraciones. La temperatura inicial se calcula de la siguiente forma: primero se establece la probabilidad de aceptar soluciones peores a la actual al inicio del algoritmo (0.4) y se generan dos soluciones aleatorias; después, se calcula la temperatura necesaria para que la probabilidad de aceptar la peor de las dos soluciones sea igual a la probabilidad deseada, considerando el método de aceptación indicado. Debemos indicar que, aunque este método de inicialización de la temperatura requiere la evaluación de dos soluciones, estas evaluaciones no se han considerado al comparar el rendimiento de ES con respecto a los otros algoritmos.
- *Búsqueda Tabú* (BT) [21]: Ésta es la metaheurística más usada y citada para la optimización de problemas combinatorios. BT promueve la aplicación de numerosas estrategias para realizar una búsqueda efectiva dentro del espacio de soluciones candidatas, y el lector interesado puede encontrar más información en la referencia anterior. Nuestro método BT implementa una memoria a corto plazo que registra los cambios de valor de las variables binarias de la solución actual más recientes. La tenencia tabú será $L/4$. Utiliza un criterio de aspiración que selecciona las soluciones que sean mejores que la mejor encontrada hasta el momento. Además, si la solución actual no se ha mejorado tras un número máximo de iteraciones (100), la memoria a corto plazo se vacía y se inicia una nueva BT a partir de otra solución aleatoria del espacio de búsqueda.
- *CHC* (en inglés, *Cross-generational elitist selection, Heterogeneous recombination, and Cataclysmic mutation*) [22]. Es un algoritmo evolutivo que combina una estrategia de selección con una elevada presión selectiva y varios componentes que inducen diversificación. CHC se comparó con varios algoritmos genéticos, dando mejores resultados especialmente en problemas difíciles [23]. Por ello, CHC se ha convertido en un punto de referencia en la literatura de algoritmos evolutivos para problemas de optimización combinatoria binaria. Su población se compone de 50 individuos.

Todos los algoritmos considerados son métodos estocásticos, y por tanto, aplican generadores de números aleatorios inicializados con una semilla específica. Debemos indicar que la semilla que utilizan estos generadores es diferente a la semilla utilizada para generar el problema de nuestro banco de pruebas que será abordado.

C. Metodología de Comparación

Para determinar la validez de los teoremas del NFL en nuestro banco de problemas es importante que los algoritmos se ejecuten bajo un mismo marco de comparación, que asegure que todos ellos tienen el mismo número de oportunidades para generar soluciones candidatas al problema. Por ello, nuestra metodología de comparación consistirá en analizar la mejor solución alcanzada tras un número máximo de evaluaciones para todos los algoritmos. En particular, cada simulación de cada algoritmo se ejecutara con un número máximo de 10^6 evaluaciones.

Los análisis estadísticos no paramétricos [24], [25] se han aplicado en la literatura para detectar la existencia o no de diferencias significativas en el rendimiento de varios algoritmos. En particular, nosotros calcularemos primero el ranking medio de cada algoritmo según el test de Friedman [26], [27]. Esta medida se obtiene calculando, para cada problema, el ranking r_j del resultado para el algoritmo j asignando al mejor de ellos el ranking 1 y al peor J (J es el número de algoritmos, cinco en nuestro caso). Después se calcula la media de los valores de ranking para todos los problemas. Por ejemplo, si un algoritmo consigue los rankings 1, 3, 1, 4 y 2, en cinco problemas, la media es $(1+3+1+4+2)/5 = 11/5$. Como puede entenderse, cuanto menor sea el ranking medio, mejor será el algoritmo correspondiente. Seguidamente, aplicaremos el test de Iman y Davenport [28] para comprobar la existencia de diferencias de rendimiento entre los algoritmos considerados, y finalmente, el test de Holm [29], para detectar diferencias de rendimiento entre el algoritmo con mejor ranking medio y el resto. Estos dos últimos análisis estadísticos toman como entrada los valores de ranking medios generados según el test de Friedman, y se aplicarán con 5 % como factor de significancia.

Es interesante remarcar que los teoremas del NFL implican que cualquiera dos algoritmos tienen exactamente el mismo rendimiento medio sobre con-

juntos de funciones c.b.p para cualquier medida de rendimiento, y en particular, valores medios de ranking según el test de Friedman.

Se han realizado un número de 1000 simulaciones por algoritmo, cada una sobre un problema aleatorio del banco de problemas. Como se avanzó en la Sección II-A, toda simulación i-ésima de cualquier algoritmo abordará exactamente el mismo problema del banco de pruebas, y diferente probabilísticamente a la simulación j-ésima con $i \neq j$. Esto es porque la secuencia de semillas para las diferentes simulaciones se genera por otro generador de números aleatorios cuya semilla se asigna al inicio del experimento. Dado que nuestros resultados podrían estar influenciados por esta semilla inicial, consideraremos posteriormente la repetición del experimento al completo con diferentes semillas (50 repeticiones).

III. Resultados

Esta sección agrupa los resultados de los experimentos. La Sección III-A compara los resultados de los algoritmos en la primera repetición del experimento. La Sección III-B analiza la dependencia de los resultados con respecto a la secuencia aleatoria de problemas abordados repitiendo el experimento 50 veces con diferentes semillas.

Debemos avanzar que los resultados se presentarán en forma de análisis estadísticos y gráficas resumen. En particular, los resultados de los algoritmos en cada problema no se presentan por ser demasiados (1000 simulaciones por 5 algoritmos por 50 repeticiones).

A. Primer Análisis

La Tabla II muestra el ranking medio de los algoritmos sobre las 1000 simulaciones del experimento. El test de Iman Davenport encuentra diferencias de rendimiento significativas entre los algoritmos considerados porque su estadístico (756.793) es mayor que el valor crítico correspondiente (2.374) al 5 %. Aplicamos entonces el test de Holm para detectar diferencias entre el algoritmo con mejor ranking (BT) y el resto. El signo más (+) en cada fila de la Tabla II indica que el test de Holm encuentra diferencias significativas entre BT y el algoritmo correspondiente. Como puede observarse, BT consigue el mejor ranking y el test de Holm encuentra diferencias de rendimiento entre éste y cualquier otro algoritmo. Además, es interesante destacar que BA obtiene el peor ranking medio.

TABLA II
Valores de ranking y test de Holm

Algoritmo	Ranking	Sig
BT	2.052	Mejor ranking
ES	2.546	+
BLM	2.758	+
CHC	2.874	+
BA	4.769	+

B. Dependencia con la semilla inicial

Dado que los resultados anteriores pueden estar influenciados por la semilla inicial que genera la secuencia de 1000 problemas a abordar, hemos repetido el mismo experimento con diferentes semillas iniciales 50 veces, es decir, cada experimento ha considerado una secuencia de 1000 problemas probabilísticamente diferentes. La Figura 1 representa las distribuciones de los valores de ranking medio de los algoritmos sobre las 50 repeticiones por medio de diagramas de caja, los cuales muestran los valores máximo y mínimo y los cuartiles primero y tercero.

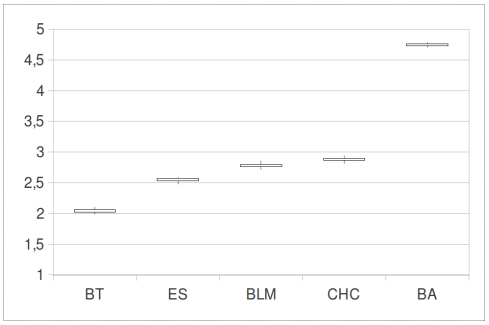

Fig. 1. Distribuciones de los valores de ranking

Puede verse que los valores de ranking de los algoritmos son muy similares entre las diferentes repeticiones y por tanto, su dependencia con respecto a la semilla que genera la secuencia de problemas a abordar es muy pequeña. Además, debemos destacar que los análisis de Iman Davenport y Holm siempre encontraban diferencias significativas entre el algoritmo con mejor ranking y cualquier otro algoritmo.

IV. Interpretación de los Resultados

De acuerdo a los resultados anteriores, podemos concluir que los teoremas del NFL no son válidos en nuestro conjunto de problemas, porque claramente aparecen diferencias significativas entre los algoritmos considerados. Además, BA ha obtenido siempre los peores valores de ranking. Sin embargo, dado que nuestro banco de problemas no era c.b.p., las diferencias de rendimiento eran ciertamente posibles. Lo que es más significativo es que, dado que habíamos considerado un conjunto de problemas de optimización binaria representativo de los que aparecen en la literatura científica, según nuestro conocimiento, podemos formular la siguiente conclusión: si el conjunto de problemas de optimización binaria que aparece en la literatura es mínimamente representativo del conjunto de problemas de optimización binaria con interés práctico en el mundo real, entonces, los teoremas del NFL tampoco son válidos para este último conjunto (como ya se intuía en [5]).

Según nuestros resultados, BT obtiene normalmente los mejores resultados tras 10^6 evaluaciones. Sin embargo, otros algoritmos podrían ser los mejores al considerar otras metodologías comparativas. Esto será objeto de futuros estudios.

V. Conclusiones

En este trabajo, hemos ofrecido evidencia empírica de que existen diferencias de rendimiento entre diferentes algoritmos cuando se considera únicamente el conjunto de problemas de optimización binaria con interés práctico en el mundo real. Estos resultados se han obtenido asumiendo que los problemas que se han abordado en la literatura son representativos de los problemas del mundo real, en al menos un mínimo nivel de significancia. Seguidamente, hemos desarrollado un estudio empírico comparando los rendimientos medios de cinco algoritmos en un conjunto representativo de los problemas binarios que han aparecido en la literatura. El análisis estadístico llevado a cabo ha concluido que ciertamente existen diferencias de rendimiento significativas entre los algoritmos considerados, proveyendo evidencia empírica a las hipótesis de Igel y Toussaint [5].

Agradecimientos

Este trabajo se ha financiado con los proyectos de investigación TIN2008-05854 y P08-TIC-4173.

Referencias

[1] D. Wolpert and W. Macready, "No free lunch theorems for optimization", *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 67–82, 1997.

[2] C. Schumacher, M. D. Vose, and L. D. Whitley, "The No Free Lunch and Problem Description Length", in *Proc. of the Genetic and Evolutionary Computation Conference*, 2001, pp. 565–570.

[3] D. Whitley and J. Rowe, "Focused no free lunch theorems", in *Proc. of the Genetic and Evolutionary Computation Conference*, 2008, pp. 811–818.

[4] J. A. R. Marshall and T. G. Hinton, "Beyond No Free Lunch: Realistic Algorithms for Arbitrary Problem Classes", in *IEEE Congress on Evolutionary Computation*, vol. 1, 2010, pp. 18–23.

[5] C. Igel and M. Toussaint, "On classes of functions for which no free lunch results hold", *Inf. Process. Lett.*, vol. 86, no. 6, pp. 317–321, 2003.

[6] P. Jiang and Y. Chen, "Free lunches on the discrete Lipschitz class", *Theoretical Comput. Sci.*, vol. 412, no. 17, pp. 1614–1628, Dec. 2010.

[7] B. Grofman, "The prisoner's dilemma game and the problem of rational choice: Paradox reconsidered", in *Frontiers of Economics*. Blacksburg, Virginia: University Publications, 1975, pp. 101–119.

[8] J. Schaffer and L. Eshelman, "On crossover as an evolutionary viable strategy", in *Proc. of the Int. Conf. on Genetic Algorithms*, R. Belew and L. Booker, Eds. Morgan Kaufmann, 1991, pp. 61–68.

[9] D. Goldberg, B. Korb, and K. Deb, "Messy genetic algorithms: motivation, analysis, and first results", *Complex Syst.*, vol. 3, pp. 493–530, 1989.

[10] D. Thierens, "Population-based iterated local search: restricting neighborhood search by crossover", in *Proc. of the Genetic and Evolutionary Computation Conf.*, K. Deb *et al.*, Eds., vol. LNCS 3103. Springer, 2004, pp. 234–245.

[11] M. Pelikan, D. Goldberg, and E. Cantú-Paz, "Linkage problem, distribution estimation, and bayesian networks", *Evol. Comput.*, vol. 8, no. 3, pp. 311–340, 2000.

[12] K. Smith, H. Hoos, and T. Stützle, "Iterated robust tabu search for MAX-SAT", in *Proc. of the Canadian Society for Computational Studies of Intelligence Conf.*, J. Carbonell and J. Siekmann, Eds., vol. LNCS 2671. Springer, 2003, pp. 129–144.

[13] S. Kauffman, "Adaptation on rugged fitness landscapes", *Lec. Sci. Complex.*, vol. 1, pp. 527–618, 1989.

[14] S. Forrest and M. Mitchell, "Relative building block fitness and the building block hypothesis", in *Foundations of Genetic Algorithms 2*, L. Whitley, Ed. Morgan Kaufmann, 1993, pp. 109–126.

[15] R. Watson and J. Pollack, "Hierarchically consistent test problems for genetic algorithms", in *Proc. of the Congress on Evolutionary Computation*, vol. 2, 1999, pp. 1406–1413.

[16] R. Karp, "Reducibility among combinatorial problems", in *Complexity of Computer Computations*, R. Miller and J. Thatcher, Eds. Plenum Press, 1972, pp. 85–103.

[17] J. Beasley, "Heuristic algorithms for the unconstrained binary quadratic programming problem", The Management School, Imperial College, Tech. Rep., 1998.

[18] D. Thierens, "Adaptive mutation rate control schemes in genetic algorithms", in *Proc. of the Congress on Evolutionary Computation*, 2002, pp. 980–985.

[19] M. J. Streeter, "Two Broad Classes of Functions for Which a No Free Lunch Result Does Not Hold", in *Genetic and Evolutionary Computation Conference*, vol. LNCS 2724, 2003, pp. 1418–1430.

[20] S. Kirkpatrick, C. Gelatt Jr, and M. Vecchi, "Optimization by simulated annealing", *Sci.*, vol. 220, no. 4598, pp. 671–680, 1983.

[21] F. Glover and M. Laguna, *Tabu Search*. Kluwer Academic Publishers, 1997.

[22] L. Eshelman and J. Schaffer, "Preventing premature convergence in genetic algorithms by preventing incest", in *Int. Conf. on Genetic Algorithms*, R. Belew and L. Booker, Eds. Morgan Kaufmann, 1991, pp. 115–122.

[23] D. Whitley, S. Rana, J. Dzubera, and E. Mathias, "Evaluating evolutionary algorithms", *Artif. Intell.*, vol. 85, pp. 245–276, 1996.

[24] 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", *J. Heuristics*, vol. 15, no. 6, pp. 617–644, 2009.

[25] S. Garcia, A. Fernández, J. Luengo, and F. Herrera, "A study of statistical techniques and performance measures for genetics-based machine learning: accuracy and interpretability", *Soft Comput.*, vol. 13, no. 10, pp. 959–977, 2009.

[26] M. Friedman, "A comparison of alternative tests of significance for the problem of m rankings", *Ann. Math. Stat.*, vol. 11, no. 1, pp. 86–92, 1940.

[27] J. Zar, *Biostatistical Analysis*. Prentice Hall, 1999.

[28] R. Iman and J. Davenport, "Approximations of the critical region of the Friedman statistic", in *Communications in Statistics*, 1980, pp. 571–595.

[29] S. Holm, "A simple sequentially rejective multiple test procedure", *Scand. J. Stat.*, vol. 6, pp. 65–70, 1979.