SIMPOSIO INTERNACIONAL DE CONSTRUCCIONES

Estrategias para mejorar el rendimiento de las meta-heurísticas en la optimización (discreta) del diseño de estructuras de HA

Strategies for improving the performance of meta-heuristics in the design (discrete) optimization of RC structures

Iván A. Negrin Diaz[1], Ernesto L. Chagoyén Méndez[2]

1- Facultad de Construcciones, UCLV, Cuba. E-mail: indiaz@uclv.cu

2- Facultad de Construcciones, UCLV, Cuba. E-mail: chagoyen@uclv.edu.cu

Resumen: La optimización estructural es usualmente un proceso complejo y extremadamente costoso desde el punto de vista computacional, especialmente cuando un software profesional de modelación, análisis y diseño es utilizado como motor de cálculo. Por tanto, la utilización de estrategias de optimización competentes es esencial para obtener resultados satisfactorios en un tiempo razonablemente corto. En este trabajo se proponen algunas alternativas para mejorar el rendimiento de los métodos más empleados en estos procesos de optimización: las meta-heurísticas. Se presentan algunos resultados de aplicar las estrategias propuestas y además se introducen algunas ideas para mejorar dichas estrategias en futuras líneas de investigación.

Abstract: *Structural optimization is usually a complex and extremely computationally expensive process, especially when a professional modeling, analysis and design software is used as the calculation engine. Therefore, the use of competent optimization strategies is essential to obtain satisfactory results in a reasonably period of time. In this paper, some alternatives are proposed to improve the performance of the most used methods in these optimization processes: meta-heuristics. Some results of applying the proposed strategies are presented and some ideas for improving these strategies are introduced for future work.*

Palabras Clave: optimización estructural; meta-heurísticas; modelos sustitutivos; ajuste de parámetros

Keywords: *structural optimization; meta-heuristics; surrogate models; parameter tuning*

1. Introducción

La optimización del diseño de obras de ingeniería civil es una temática que ha venido cobrando un gran auge en los últimos años, debido al continuo desarrollo de las herramientas computacionales y la creciente necesidad de ahorrar recursos y ser respetuosos con el medio ambiente.

Debido a determinadas características de estos problemas de optimización, usualmente de naturaleza discreta y combinatoria, el uso de métodos meta-heurísticos es la principal vía de darles solución. Estos métodos aplican operadores estocásticos para explorar el espacio de soluciones y guiar la búsqueda hacia mejores diseños en dependencia del o los objetivos analizados (Dillen et al, 2021a) (Negrin et al, 2021b). El rendimiento de estos métodos, por tanto, depende de tres instancias principales: (1) el problema de optimización, (2) el valor o estrategia utilizada por cada uno de estos operadores y (3) el carácter aleatorio inherente a los algoritmos estocásticos (Dillen et al, 2021). Por tanto, varias estrategias se pueden aplicar para mejorar este rendimiento, como el ajuste de parámetros (*parameter tuning* en inglés) (Smith and Eiben, 2010a) (Smith and Eiben, 2010b) (Eiben and Smith, 2011a) (Eiben and Smith, 2011b) (Dillen et al, 2018) (Dillen et al, 2021) (Negrin et al, 2020) (Negrin et al, 2021a) (Negrin et al, 2021b) o el uso de hibridaciones para mejorar, la estrategia en general (el uso de dos o más métodos en un mismo procedimiento de optimización) (Negrin et al, 2019a), o los propios métodos combinando conceptos de varios para crear una nuevo.

Uno de los grandes problemas de chequear el rendimiento de los métodos en los problemas de optimización estructural, es lo costoso, desde el punto de vista computacional, que son estos procedimientos. Alternativas como el uso de modelos sustitutivos (Negrin et al, 2021a) (Negrin et al, 2021b) se pueden utilizar para hacer factible el testeo de estas estrategias.

En este trabajo se exponen algunas de estas alternativas y se demuestran algunos resultados de aplicarlas, evidenciando como se pueden mejorar los resultados al

utilizarlas. En el siguiente capítulo se exponen las principales características de las meta-heurísticas y se explica el funcionamiento de algunas de las más utilizadas. Además se desarrollan algunas ideas acerca del ajuste de parámetros. En el capítulo 3 se exponen algunos resultados obtenidos, así como las líneas futuras de investigación basadas en estos resultados.

2. Metodología

En este capítulo se exponen las principales características de las meta-heurísticas en general, y de algunas de las más utilizadas por la comunidad científica y en los estudios mostrados en el capítulo 3 en particular. Además se ofrece un resumen del ajuste de parámetros, una de las principales estrategias propuestas en este estudio para mejorar el rendimiento de estos métodos. También se desarrolla la propuesta de crear los modelos sustitutivos que permiten estudiar los métodos de optimización en procesos mucho más "ligeros" que los reales.

2.1 Formulación del problema de optimización del diseño de estructuras de HA

La formulación del problema de optimización consiste en identificar la o las funciones objetivo, las variables/parámetros asignados y las restricciones.

Los *objetivos de optimización* de estructuras de hormigón armado (HA) más comunes son: costo económico (Negrin et al, 2019a) (Negrin et al, 2019b) (Negrin et al, 2021a) (Negrin et al, 2021b) (Negrin et al, 2021c), criterios medioambientales como emisión de CO_2 (Negrin et al, 2021c) y energía incorporada, o EE por sus siglas en inglés (Negrin et al, 2021c), además de otros relacionados con la edificabilidad (Payá et al, 2008) (o cuan fácil es de construir una estructura), la seguridad, y más recientemente la durabilidad de estas estructuras, teniendo en cuenta el Análisis de Ciclo de Vida (LCA) de la obra. Como se ha mencionado, estas *funciones objetivo* se expresan, generalmente, como funciones no lineales, con la característica de ser funciones difíciles de optimizar, con la presencia de muchos óptimos locales, ya que la configuración del acero de refuerzo (tipo y colocación de las barras) varía fuertemente con las dimensiones de la sección transversal.

Por su parte, *las variables* en estos problemas son, usualmente, las dimensiones de las secciones transversales de los elementos, la rectangularidad de los cimientos en caso que se tengan en cuenta en el proceso, las propiedades de los materiales, la solución de refuerzo en caso que se desee medir objetivos constructivos o de durabilidad. Para

ofrecer soluciones prácticas desde el punto de vista ingenieril, estas variables son usualmente discretas. Por ejemplo: las dimensiones de las secciones transversales se establecen como múltiplo de 5 cm (30, 35, 40…cm) o las propiedades, digamos del hormigón se establecen en intervalos de 5 MPa (25, 30, 35, 40…MPa). Lo mismo pasa con la rectangularidad de los cimientos y con la distribución del refuerzo. Por tanto, estos problemas se convierten en problemas de optimización discreta o combinatoria, lo cual tiene como aspecto positivo que reduce notablemente el espacio de soluciones, pero a la vez inhabilita el uso de muchas estrategias de optimización, siendo las heurísticas y las meta-heurísticas las más indicadas para resolver dichos problemas.

Las restricciones, por su parte, son aquellas que se imponen para que el diseño de la estructura sea correcto, es decir, que cumpla con todas las normas establecidas en cuanto a resistencia, deformación, ductilidad, durabilidad, etc. Se pueden dividir en dos grandes grupos. Las *restricciones de diseño* (o *explícitas*) son aquellas impuestas a las variables de diseño directamente, y tienen en cuenta varios objetivos como funcionalidad, transporte o estético, y se presentan como $X_{min} \leq X \leq X_{max}$, es decir, establecen el límite de movimiento de las variables. Por su parte, las *restricciones de comportamiento* (o *implícitas*), usualmente llamadas *ecuaciones de estado*, son aquellas que gobiernan el diseño de los elementos, en nuestro caso, para que estos cumplan los criterios de los estados límites. Estas ecuaciones son usualmente complejas, con un alto grado de no-linealidad y son difíciles de evaluar por parte del método de optimización. En el caso de las ecuaciones de estado, estas siempre se van a cumplir, porque el diseño (o solución del refuerzo en función del área de hormigón) se basa en su cumplimiento. Lo que puede pasar es que esa solución no cumpla otros criterios como constructivos (la cantidad de acero obtenida no es permitida para la sección) o de rigidez (para las dimensiones obtenidas hay un elemento que se deforma más de lo permitido, o el desplazamiento de la edificación en el tope del edificio supera el límite). En este caso, se penaliza el valor obtenido por la función objetivo, de forma tal que esta solución quede fuera de las posibles seleccionadas como "óptima". En el caso de los modelos sustitutivos, el uso de las restricciones es mucho más fácil, aunque muy importante. Esto se explica en el apartado donde se desarrolla esta estrategia.

Después de analizados los elementos más importantes de la formulación matemática del problema de optimización, la Fig. 1 resume los conceptos expuestos.

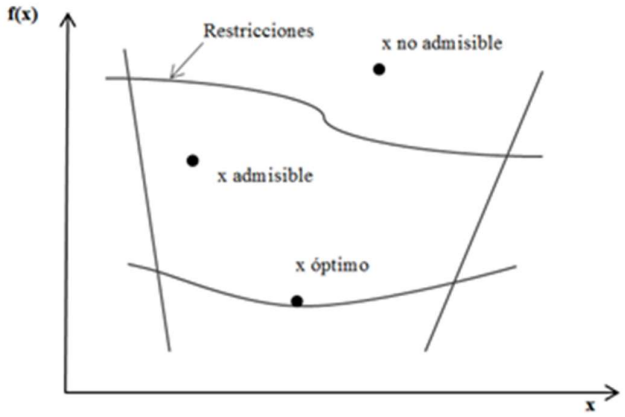


Figura 1. Resumen gráfico del problema de optimización (Negrin et al, 2019).

2.2 Métodos de optimización

Una vez definido cierto problema de optimización, es necesario encontrar el método apropiado para darle solución. Existen una gran cantidad de estos métodos, los cuales se pueden agrupar de varias formas. En este artículo se utilizará la adoptada por (Rahmanian et al, 2014), donde se dividen en dos grandes grupos: programación (1) lineal y (2) no-lineal. El enfoque de la programación lineal puede ser aplicado a problemas donde la función objetivo y las restricciones se expresan como funciones lineales, el cual no suele ser el caso de la optimización estructural, y menos aún el de la optimización del diseño de estructuras de hormigón armado, procesos en los cuales se enfoca este trabajo. Por su parte, los métodos con enfoque no-lineal se pueden dividir en tres grandes subgrupos: (1) enfoque enumerativo, (2) determinístico (o exacto) y (3) estocástico o (meta) heurístico.

El primero consiste en recorrer todo el espacio de soluciones y, utilizando la función objetivo, se obtiene la mejor solución. Esto garantiza encontrar el óptimo global, pero a costa de un gran número de evaluaciones de la función objetivo. Se han propuesto otras alternativas basadas en esta idea, tratando de hacer la búsqueda más "efectiva, como el denominado "ramificación-y-acotamiento" (*branch-and-bound* en inglés), aunque sigue siendo no factible para problemas de elevada complejidad. Este enfoque, también denominado programación lineal entera (o mixta) resuelve, en un tiempo de cómputo razonable, solo problemas "pequeños" de optimización (Van Mellaert et al, 2018).

Por su parte, los métodos determinísticos o exactos realizan una búsqueda sucesiva dentro del espacio de soluciones, basándose en información brindada por el valor de la función objetivo, su gradiente, o ambos. Por tanto, estas estrategias son usualmente de búsqueda local, es decir, son buenos *explotadores* y no *exploradores*, siendo este último un término empleado para métodos de búsqueda global. Además, estos métodos se enfocan en la optimización utilizando variables continuas, por lo cual quedan excluidos como posible solución a los problemas de optimización estructural discreta, como los abordados en este estudio.

2.2.1 Métodos heurísticos y meta-heurísticos

Quedando inhabilitados los métodos de enfoque enumerativo, debido a su gran costo computacional, y los exactos, debido a las limitaciones que ofrecen en cuanto al tipo de variables y su estrategia de búsqueda, es necesario recurrir a una serie de métodos que se basan, principalmente, en conceptos derivados de procesos naturales: las heurísticas y meta-heurísticas. Estos basan su búsqueda aplicando reglas de probabilidades y una forma de búsqueda estocástica (Sivanandam y Deepa, 2008). Cuando los métodos determinísticos no son los ideales para encontrar óptimos globales en problemas de optimización complejos, o son computacionalmente muy costosos de abordar, los métodos estocásticos son capaces de proveer buenas soluciones en un período razonable de tiempo (Rahmanian et al, 2014). Por tanto, estos métodos son ideales para problemas de optimización de alta complejidad, con muchas variables y superficies de respuesta con la presencia de muchos óptimos locales. Los enfoques más comunes están fundados en computación evolutiva (o *Evolutionary Algorithms*, *EA*) y en los algoritmos basados en enjambres.

Ahora bien ¿cuál es la diferencia entre una heurística y una meta-heurística? Los métodos heurísticos son los mencionados con anterioridad, es decir, todos aquellos que se inspiran en fenómenos naturales y utilizan operadores estocásticos para guiar la búsqueda hacia mejores soluciones. Por tanto, las meta-heurísticas son estrategias generales que mejoran los conceptos de las heurísticas originales, tratando de combinar inteligentemente estos conceptos para mejorar la estrategia de búsqueda. Muchos de los métodos que se utilizan hoy día se podrían catalogar entonces como meta-heurísticas, porque parten de los conceptos tradicionales y han sido transformados con el paso del tiempo para mejorar su rendimiento.

A continuación se explica el funcionamiento de dos heurísticas básicas como son los Algoritmos Genéticos (GA) y la Optimización por Enjambre de Partículas (PSO). Además se introduce una tercera meta-heurística creada hace relativamente poco tiempo, la cual demuestra ser muy efectiva para resolver problemas de optimización discreta.

2.2.1.1 Algoritmos Genéticos

Los algoritmos genéticos (*Genetic Algorithms*, *GA*) son posiblemente la heurística más difundida y utilizada. Fue originalmente propuesta por (Holland, 1975) y ha sufrido varias modificaciones a lo largo de la historia. A continuación se brindan los principales pasos empleados por la estrategia para realizar la búsqueda de mejores soluciones (Matlab, 2015):

1. El algoritmo crea una población aleatoria conformada por *PopSize* individuos.
2. A partir de esta población inicial, el algoritmo crea nuevas poblaciones. En cada paso se utilizan individuos de la población actual para crear futuros individuos, según la siguiente metodología:
    a. Puntúa cada individuo de la población en función de su valor de aptitud (o valor correspondiente a la función objetivo)
    b. Escala las puntuaciones brutas de aptitud para convertirlas en un rango de valores más utilizable.
    c. Selecciona dos soluciones, denominadas "padres" para generar una nueva solución (hijo). Esta estrategia de solución (*SelFcn*) es uno de los operadores estocásticos mencionados con anterioridad. Existen diferentes tipos de funciones para seleccionar los padres, aunque generalmente los padres seleccionados son los mejores adaptados (o que brindan un mejor valor de función objetivo)
    d. Un determinado número de los mejores individuos de la población "sobreviven" a la iteración (o generación) y pasan a la próxima exactamente como son. Este proceso se denomina "elitismo". El valor de *ECountFract* es una fracción entre 0 y 1, y representa el porciento del tamaño de población que sobrevivirá hacia la próxima.
    e. Se producen los "hijos" de los "padres" seleccionados. Los hijos se producen o por "cruzamiento" o por "mutación". Las funciones de cruzamiento y de

mutación son otros operadores de suma importancia. *CrossFract* es una fracción entre 0 y 1 que representa el porciento de la nueva población que será generada (además de las soluciones elite) a partir de cruzamiento. Los demás se generarán por mutación. Las funciones de cruzamiento y de mutación son otros operadores de suma importancia.

f. Se reemplazan las nuevas soluciones por las anteriores

3. El algoritmo continúa hasta que se cumple uno de los criterios de parada.

Se puede decir que esta metodología es la base de las EAs. A partir de aquí se han generado un sinnúmero de nuevas estrategias, como la que se describe a continuación.

2.2.1.2 Optimización por Enjambre de Partículas

La Optimización por Enjambre de Partículas simula el comportamiento de las bandadas de pájaros. Fue propuesto por (Kennedy y Eberhart, 1995). Las partículas vuelan a través del espacio del problema de D dimensiones aprendiendo de las mejores experiencias de todas las partículas. Por lo tanto, las partículas tienen una tendencia a volar hacia mejores áreas de búsqueda en el transcurso del proceso de búsqueda.

El algoritmo consta básicamente de los siguientes pasos:

1. El algoritmo PSO comienza creando las partículas iniciales. Cada partícula tiene una posición y una velocidad inicial.
2. El algoritmo evalúa la función objetivo (o función de aptitud) en cada ubicación de las partículas. Luego determina el mejor valor de la función y la mejor ubicación.
3. A continuación, actualiza iterativamente las ubicaciones de las partículas, las velocidades y los vecindarios.
   a. La información global se obtiene de la vecindad. El tamaño inicial del vecindario (N) se obtiene de acuerdo a:

$$N = \max(1, floor(SwarmSize * MinFractNeigh)) \quad (1)$$

   aquí *SwarmSize* es el número de partículas en el enjambre y *MinFractNeigh* es el tamaño mínimo de vecindad adaptable. El tamaño de la vecindad se actualiza en cada iteración, de acuerdo con la mejora o no del valor de la mejor función.

   b. La velocidad de cada partícula se actualiza:

$$v_{upd} = W * v + y1 * u1 * (p - x) + y2 * u2 * (g - x) \quad (2)$$

aquí, *W* es la inercia de la partícula. Se inicializa con el valor máximo de *InertiaRange* (o mínimo si *InertiaRange* tiene valores negativos), que establece los límites inferior y superior de la inercia adaptativa; v es la velocidad anterior; *y1* (*SelfAdj*) e *y2* (*SocAdj*) son constantes de aceleración que representan la ponderación de los términos de aceleración estocástica que empujan a cada partícula hacia las mejores posiciones personales y globales, respectivamente; *u1* y *u2* son números aleatorios distribuidos uniformemente y sujetos a [0,1]; *(p-x)* es la diferencia entre la posición actual y la mejor posición que la partícula ha visto y *(g-x)* es la diferencia entre la posición actual y la mejor posición en la vecindad actual.

c. La posición se actualiza según:

$$x_{upd} = x + v_{upd} \quad (3)$$

aquí *x* es la posición previa.

Las iteraciones continúan hasta que el algoritmo alcanza un criterio de parada.

2.2.1.3 Optimización Basada en Biogeografía

Como se ha mencionado, este método pertenece a las EAs. Fue propuesta por (Simon, 2008), y representa modelos matemáticos de cómo las especies migran de una isla a otra, cómo surgen nuevas especies y cómo se extinguen las especies. Se dice que las áreas geográficas que son adecuadas como residencia para especies biológicas tienen un alto índice de idoneidad del hábitat (HSI). La correspondencia entre la terminología BBO y la terminología clásica de las EAs es la siguiente: conjunto de hábitats, hábitats, especies y HSI corresponden respectivamente a población, individuos (soluciones), genes y valor de aptitud. Por tanto, el número de especies en cada hábitat es igual al número de variables en el problema de optimización. El número de hábitats es igual al tamaño de la población.

El algoritmo trabaja básicamente siguiendo los siguientes pasos:

1. El procedimiento comienza con un conjunto inicial aleatorio de hábitats con una distribución HSI uniforme. En cada iteración, los coeficientes de emigración e inmigración, denotados respectivamente por μ y λ, se asignan a cada hábitat. Las

soluciones o hábitats con un HSI alto reciben valores altos de μ y valores bajos de λ, y viceversa.

2. El algoritmo procesa los hábitats en orden decreciente de HSI, utilizando los parámetros μ, λ y probabilidad de mutación de la siguiente manera:
   a. Dentro de cada hábitat, para cada especie se analiza la posibilidad de realizar el proceso de migración: cada especie se verifica en función del coeficiente de inmigración del hábitat λ. Por lo tanto, las especies de los mejores hábitats tienen pocas posibilidades de entrar en este proceso, mientras que esta posibilidad aumenta cuando se consideran hábitats con una λ más alta.
   b. Una vez que una especie ingresa al proceso de migración, se selecciona otra especie de otro hábitat mediante la selección de la rueda de la ruleta (para seleccionar el hábitat) en función de μ para inmigrar al hábitat en el que se está trabajando.
   c. Una vez seleccionadas las especies se inicia la inmigración, que no es la sustitución de una por otra, sino una combinación de ambas, que se realiza como:

$$NewSpecies^{i}_{k} = Species^{i}_{k} + \alpha\,(Species^{j}_{k} - Species^{i}_{k}) \tag{4}$$

   dónde $Species^{i}_{k}$, es decir la *k-th* especie del hábitat *i*, es la especie analizada y $Species^{j}_{k}$, es decir la *k-th* especie del hábitat *j*, es la especie seleccionada para inmigrar, y α (*Alpha*) es el coeficiente de aceleración.
   d. Además, las especies pueden mutar con cierta probabilidad según:

$$NewSpecies^{i}_{k} = NewSpecies^{i}_{k} + \sigma * N(0,1) \tag{5}$$

   donde σ es el tamaño del paso de mutación y *N(0,1)* es un número aleatorio con media 0 y desviación estándar 1. Después de cada iteración, σ disminuye, modificado por la amortiguación del tamaño del paso de mutación.

Una vez analizada toda la población, la nueva se forma seleccionando los mejores hábitats de la población anterior (antes de ser transformada) y los mejores de la nueva población. *KeepRate* denota la fracción de la población anterior que sobrevive. Esto es similar al elitismo utilizado en los AG. Este proceso iterativo termina cuando se cumple un criterio de parada.

2.3 Estrategias para aumentar el rendimiento

En este epígrafe se desarrollará la idea central del artículo. Primeramente se expone una forma de lograr un paso extremadamente importante antes de mejorar una estrategia: probarla. Se había mencionado que los procesos de optimización estructural son extremadamente costosos desde el punto de vista computacional, y aún más si se utiliza como motor de cálculo un software de modelación, análisis y diseño estructural. Por tanto, testear las estrategias en estos procesos es casi imposible.

Seguidamente se introduce la idea de un procedimiento usualmente ignorado en la optimización estructural como el ajuste de parámetros. Además se expone la combinación de estrategias para crear una mejor como posible alternativa.

2.3.1 Creación de modelos sustitutivos

Para poder hacer posible el testeo de los diferentes métodos es necesario crear modelos sustitutivos. Para este caso se proponen dos alternativas relativamente sencillas. La primera es la creación de una base de datos que contenga todo el espacio de soluciones de un problema real (Negrin et al, 2021b). Es decir, se computan todas las posibles soluciones y se almacenan, y una vez que se quiera optimizar este modelo, se accede a la base de datos sin necesidad de utilizar el procedimiento convencional de modelación, análisis y diseño. La gran desventaja de esta estrategia es que se sólo pueden hacer bases de datos de estructuras relativamente sencillas, con pocas variables. Aunque, para la optimización de modelos más complejos, se puede adoptar la táctica de probar el o los métodos de optimización en ejemplos sencillos, y sacar conclusiones para aplicarlas en estructuras más complejas (Negrin et al, 2021b).

La segunda alternativa se trata de utilizar funciones analíticas de referencias (Suganthan et al, 2005) para crear modelos de características similares a los reales, en el cual probar los métodos de optimización (Negrin et al, 2021ª) (Negrin et al, 2021b) (ver Fig. 2). Para esto se utiliza la función de referencia con un igual número de variables y cada variable puede tomar un número igual de valores que el problema real. Entonces, con las restricciones explícitas, se toma un sub-espacio discreto de la función de referencia continua, creando un problema de optimización discreto con la complejidad que se desee.

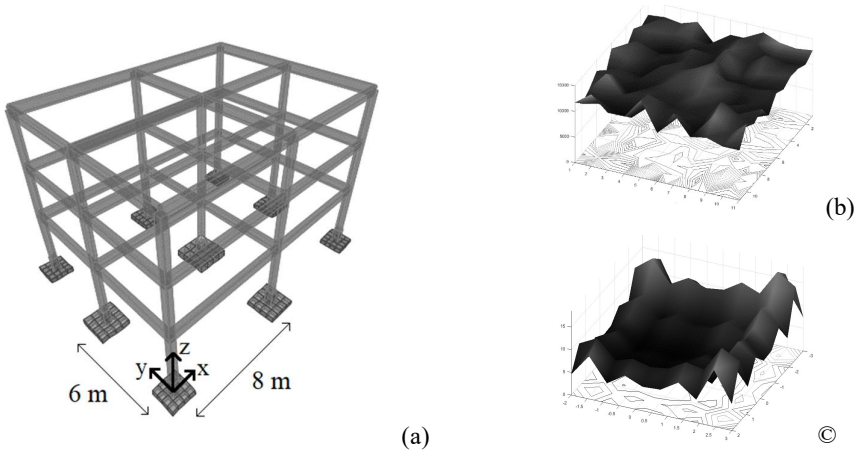


Figura 2. Segunda estrategia para crear modelos sustitutivos, (a) es una caso de estudio real, con 14 variables y un total de 1.5 x 10^{10} posibles soluciones, (b) es la superficie de respuesta real del costo en función del peralto de dos grupos de vigas y (c) es la utilización de la función *Ackley*, una función de referencia, para crear un modelo simulando a (a) y (b) (Negrin et al, 2021[a]).

Así, se evaluarán los procesos de optimización en una función analítica y no en un procedimiento de modelación, análisis y diseño estructural. Este modelos sustituto no será el mismo que el real, pero tendrá características similares, y los resultados obtenidos con el serán perfectamente aplicables a la optimización de la estructura real.

2.3.2 Ajuste de parámetros

El ajuste de parámetros (*parameter tuning*), como se ha mencionado, es un proceso extremadamente costoso desde el punto de vista computacional, mediante el cual, se encuentran valores adecuados de los operadores estocásticos para enfrentar un determinado problema. Este proceso puede ser muy útil cuando estamos, por ejemplo, probando varios métodos de optimización para resolver un determinado problema, como se aprecia en la Fig. 3.

Algunas de las estrategias más utilizadas para permitirse este proceso son: La meta-optimización, el diseño de experimentos (DOE), la optimización basada en modelos y el aprendizaje automático, y la configuración de algoritmos sin modelos (Dillen et al, 2021).

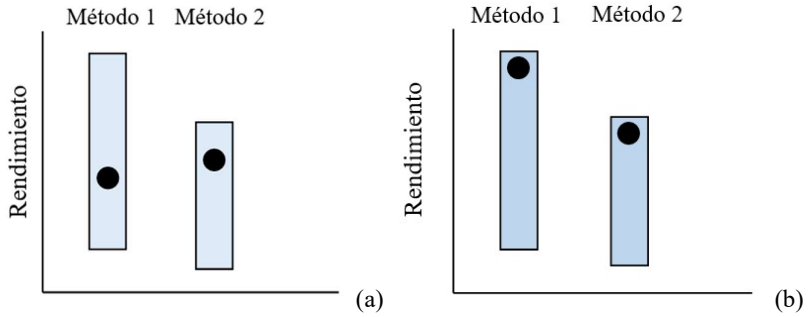


Figura 3. Representación gráfica del beneficio del ajuste de parámetros; (a) métodos con parámetros sin ajustar, se puede apreciar como el método 1, aun siendo más apropiado para enfrentar el problema, presenta un rendimiento inferior que el número 2; (b) con el ajuste de parámetros se puede llegar a una configuración óptima, o cercana a esta que permita un buen rendimiento (Eiben and Smith, 2011a)

La meta-optimización consiste en utilizar los parámetros del algoritmo como variables en un problema de optimización. Así, la función objetivo es la métrica de utilidad del rendimiento del algoritmo. Los enfoques más comunes de DOE incluyen diseños factoriales, junto con análisis de varianza o análisis de regresión (Adenso-Diaz y Laguna, 2006) (Coy et al, 2001) (Rardin et al 2001). Estas estrategias se han ido mejorando, con la aparición de otras nuevas como Diseño y Análisis de Experimentos por Ordenador (DACE), Optimización de Parámetros Secuenciales (SPO) o Configuración de Algoritmos Basados en Modelos Secuenciales (SMAC).

Las configuraciones de algoritmos sin modelo se basan principalmente en pruebas de hipótesis estadísticas para comparar diferentes ajustes de parámetros (Czarn et al, 2004) (Castillo-Valdivieso et al, 2002) (Xu and Glober, 1998). El inconveniente de esta propuesta es el gran número de ejecuciones experimentales necesarias para obtener una precisión estadística suficiente. Para superar este inconveniente, otras estrategias han surgido para obtener información no trivial sobre la interrelación de los parámetros y su influencia en el rendimiento del método, prediciendo posibles resultados y acelerando la convergencia del algoritmo, disminuyendo considerablemente el costo computacional. Entre estos nuevos algoritmos tenemos: el algoritmo F-Race (Birattari et al, 2002), el F-Race mejorado (Balaprakash et al, 2007), combinación de F-Race y meta-optimización (Yuan and Gallagher, 2007), y otros.

La propuesta realizada en este artículo de basa en la configuración de algoritmos sin modelo, aunque sin tener en cuenta la compleja interacción entre los parámetros y sus

valores. Por tanto, cada parámetro se establece con sus valores, y se combinan cada una de las posibles soluciones. Este procedimiento, como se ha mencionado, es costoso, por lo que en las líneas de investigaciones futuras se proponen alternativas para aligerar considerablemente estos procedimientos.

2.3.2.1 Definiendo la utilidad

Para medir el rendimiento del algoritmo, se debe establecer una métrica o utilidad. Las formas clásicas de realizar estas mediciones incluyen la calidad de las soluciones y el tiempo de cómputo empleado. Las tres utilidades clásicas son (Eiben and Smith, 2011b): mejor rendimiento promedio (MBF), número medio de evaluaciones para la solución (AES) y tasa de éxito (SR). Sin embargo, en este trabajo se muestra la propuesta de (Negrin et al, 2020) (Negrin et al, 2021b), la cual engloba al unísono el MBF y la velocidad de convergencia, utilizando la curva de rendimiento promedio.

Suponemos que la meta-heurística se ejecuta N veces con un vector de parámetros fijo dado para reducir los efectos estocásticos. La curva de rendimiento promedio, o APC (aptitud media en función del número de iteraciones), denotada como *fa(x)*, se obtiene promediando las curvas de rendimiento de las N ejecuciones del método (véase la Fig. 4). Esta curva ofrece más información sobre el rendimiento que las simples medidas de rendimiento, como MBF o AES, y permite una fácil comparación entre dos o más procedimientos. Por lo tanto, la utilidad propuesta se basa en el uso del APC (Negrin et al, 2021b).

Esta utilidad fue propuesta en (Negrin et al, 2021b), y tiene en cuenta tanto la velocidad de convergencia como el resultado final del proceso, o calidad de las soluciones. Esto se logra, al utilizar como utilidad el área debajo de la APC, o un promedio ponderado de esta, asignándole más peso al último punto (o MBF). Para encontrar una información más detallada, consultar el trabajo citado.

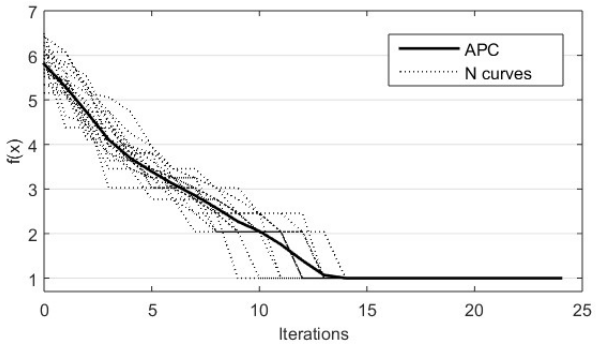


Figura 4. Obtención de la APC (Negrin et al, 2021a) (Negrin et al, 2021b)

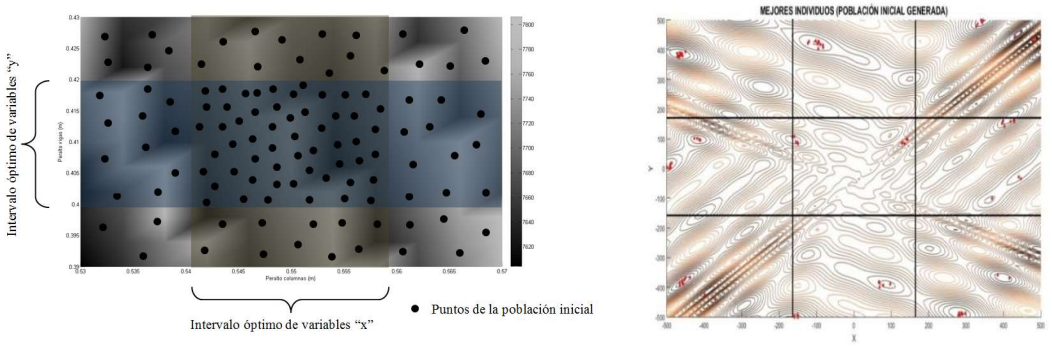
2.4 Hibridaciones

Otra forma clásica de mejorar el rendimiento de una estrategia es mediante hibridaciones. Estas se pueden catalogar de "fuertes" y "débiles". Las primeras se consisten en hibridar conceptos de varias (meta) heurísticas para crear una nueva estrategia, es decir, utilizar operadores propios de un determinado métodos para mejorar el rendimiento de otro. También se puede crear una estrategia que utilice dos o más métodos, combinados e manera ingeniosa para mejorar la búsqueda por el espacio de soluciones. Las segundas son más sencillas, y consisten en utilizar dos o más métodos de una forma menos sofisticada. Por ejemplo, es conocido que las (meta) heurísticas son generalmente buenas exploradoras (optimización global), pero malas explotadoras (optimización local). Por tanto, se puede establecer un proceso de búsqueda que se inicie con una algoritmo estocástico, el cual encuentre una buena solución, pero mejorable. A partir de aquí se utiliza un método de búsqueda local que explote el espacio cercano a la solución encontrada por el primero, como se realizó en (Negrin et al, 2019b).

2.4.1 Creación de una población inicial

Todos estos métodos estocásticos tienen la posibilidad de "obtener" una población para iniciar el proceso, creada a partir de otros métodos más eficientes que los procedimientos aleatorios tradicionales. Esta población se puede crear a partir de criterios de pre-dimensionamiento previamente obtenidos (para la optimización de diseño estructural) (ver Fig. 5ª) (Negrin et al, 2019b). También mediante hibridaciones "débiles", utilizando un método para crear una población inicial para que otro realice la

búsqueda tradicional (Negrin et al, 2019ª), o utilizando una sola estrategia que cree su propia población inicial (Negrin et al, 2020). La desventaja de este procedimiento es el costo computacional extra que supone explorar el espacio de soluciones para crear la población inicial, pero esta última puede garantizar una búsqueda más eficiente por el espacio de soluciones en el proceso tradicional de optimización (Negrin et al, 2020).



(a) (b)

Figura 5. Ejemplo de poblaciones iniciales generadas, (a) concentrando la mayor parte de los individuos en la zona de mayor probabilidad de albergar las mejores soluciones, obtenida a través de los posibles intervalos óptimos de cada variable atendiendo a criterios de pre-dimensionamiento (Negrin et al, 2019b), (b) división del espacio de soluciones en sub-espacios para realizar la búsqueda, se puede apreciar cómo la población inicial (puntos rojos) se concentra en los óptimos locales de cada sub-espacio (Negrin et al, 2020)

En (Negrin et al, 2020) se logra realizar la búsqueda de esta población inicial de una manera más eficiente, dividiendo el espacio de soluciones en sub-espacios, y realizando búsquedas locales (ver Fig. 5b).

3. Resultados y discusión

Los resultados se brindarán acorde a investigaciones previas, agrupados en función de las estrategias descritas en el epígrafe anterior, y los beneficios de utilizarlas. Primero se mostrarán resultados de aplicar lasa hibridaciones y la creación de poblaciones iniciales. Después se analiza el beneficio de ajustar los parámteros de los métodos para cada problema.

3.1 Hibridación y creación de población inicial

Aquí se muestran resultados de realizar hibridaciones de búsqueda global con búsqueda local, para crear poblaciones iniciales y la creación de estas poblaciones utilizando el propio algoritmo de búsqueda tradicional.

3.1.1 Hibridación de búsqueda global con búsqueda local

En (Negrin et al, 2019b), el uso de GA simple brindaba soluciones diferentes en cada proceso. Estas soluciones no eran muy diferentes, lo cual indica que GA simple encontraba una zona de óptimos locales cerca del óptimo global. Por esta razón se propuso la siguiente estrategia: una vez terminada la búsqueda local con GA, se utilizaba el resultado brindado como punto de partida para un algoritmo de búsqueda local el algoritmo Nelder-Mead, lográndose la estabilización de los resultados. La fig. 6 representa una solución utilizando esta estrategia. Además en este trabajo se utilizó la estrategia de brindar la población inicial en función de recomendaciones de diseño (Fig. 5ª), con lo que se obtuvo un aumento significativo de la velocidad de convergencia de la estrategia.

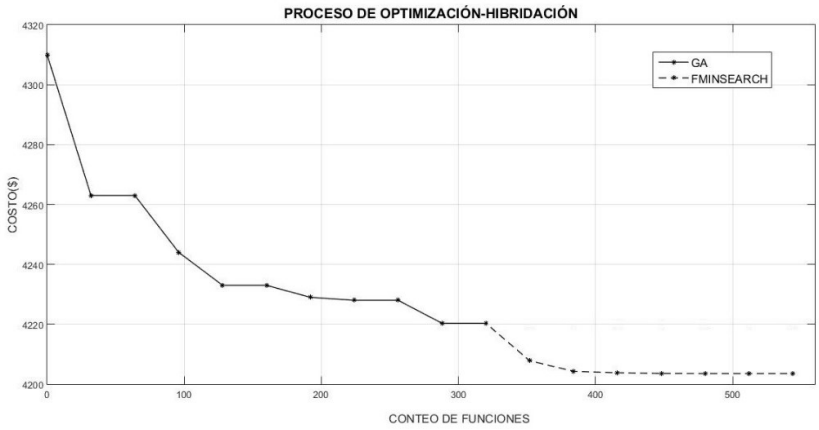


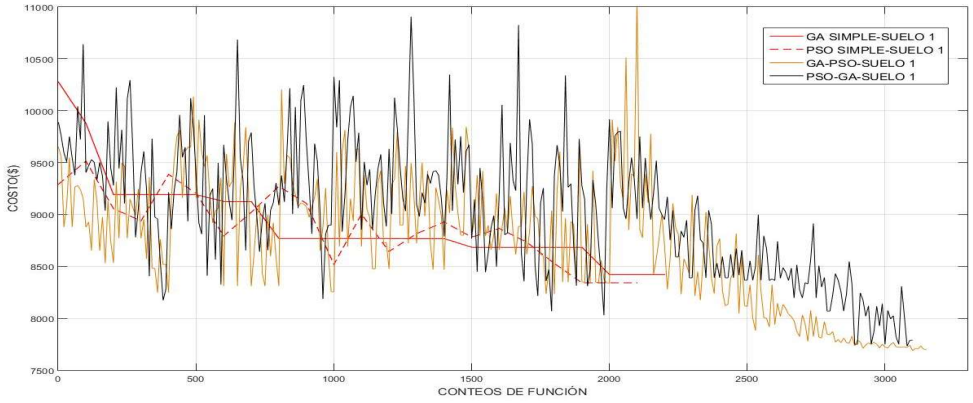
Figura 6. Resultados al aplicar la hibridación de GA con el algoritmo Nelder-Mead (fminsearch en Matlab). Se puede apreciar como el algoritmo "local" mejora la solución encontrada por el "global" (Negrin et al, 2019b).

3.1.2 Hibridación para crear una población inicial

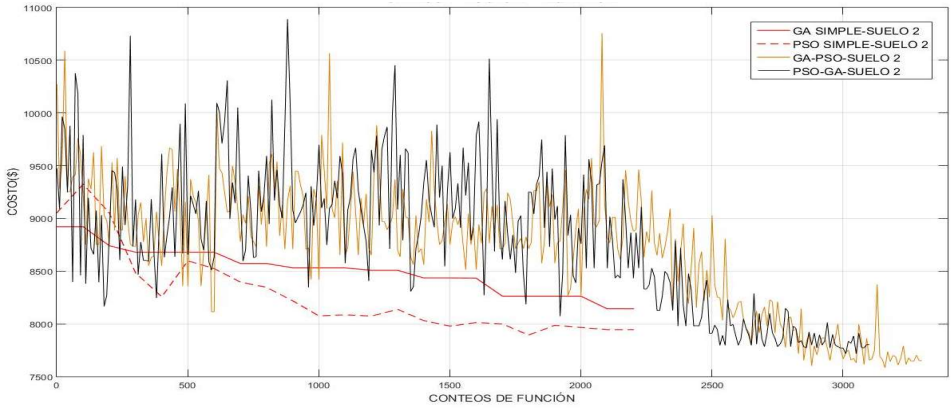
La idea de crear una población inicial personalizada partía de la hipótesis de que esto no solo mejoraría la velocidad de convergencia del algoritmo en la búsqueda tradicional, sino que mejoraría la calidad de las soluciones en problemas de optimización de elevada complejidad. La desventaja, como se ha mencionado, es el costo computacional extra para crear dicha población.

En (Negrin et al, 2019a) se presentaba un problema de optimización sumamente difícil de optimizar, no solo por el consumo computacional, sino también por el número de variables y el gran espacio de soluciones en general. Para darle solución, se probó la estrategia de utilizar GA para realizar una búsqueda general y crear una población

inicial para PSO, y viceversa. En la Fig. 7 se puede apreciar como la primera opción resultó ser más efectiva, mejorando las soluciones alcanzadas por las metodologías simples hasta en un 10 %, aunque a costa de un mayor consumo computacional.



(a)

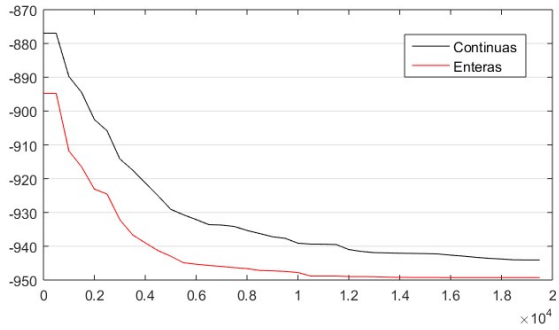


(b)

Figura 7. Resultados de aplicar diferentes estrategias a un caso de estudio para dos tipos de suelos (Negrin et al, 2019a). El "desorden" de las estrategias compuestas obedece a la búsqueda de la población inicial, donde se grafica el mejor resultado encontrado cada 10 conteos de función.

3.1.3 Creación de población inicial utilizando el propio algoritmo

En (Negrin et al, 2020) se utilizó GA para la optimización de la función de referencia Eggholder. En este caso se realizó un ajuste de parámetros y se probaron estrategias como optimizar utilizando variables enteras en vez de continuas, y la creación de una población inicial utilizando el propio GA, siguiendo la estrategia representada en la Fig. 5b.



(a)

Método	MBF	Cont fun
Variables cont.	-944.08	19500
Variables disc.	-949.29	19500
Compuesto	-956,92	22610

(b)

Figura 8. Resultados alcanzados en Negrin et al. (2020), (a) APCs al optimizar variables enteras y continuas y (b) MBF y consumo computacional al utilizar además, el método compuesto (creación de la población inicial).

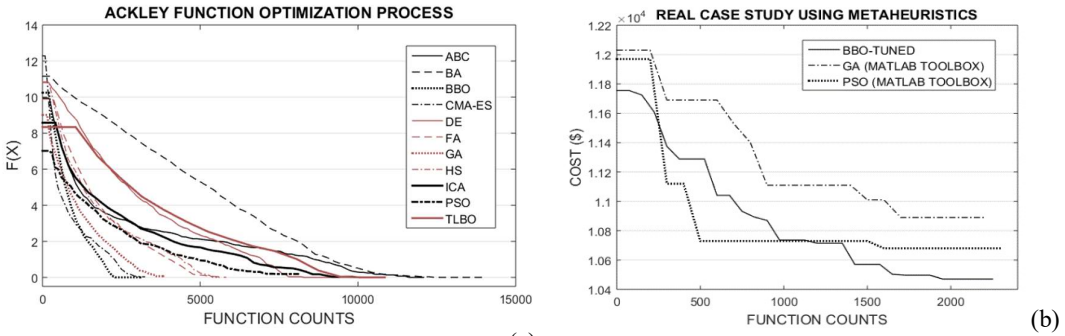
En la Fig. 8 se puede apreciar cómo, utilizar variables enteras puede ser más efectivo que utilizar variables continuas, ya que se reduce notablemente el espacio de soluciones. Por otra parte se puede observar la superioridad del método compuesto. GA simple con los parámetros ajustados, utilizando variables continuas encontró el óptimo global el 30% de las 20 pruebas, con variables enteras el 90% de las veces. El método compuesto fue capaz de encontrarlo el 100% de las veces, con un simple aumento del 16% del costo computacional.

3.2 Ajuste de parámetros

En el epígrafe 2.3.2.1 se había definido una forma de medir el rendimiento de un método basado en el uso de la APC. Sin embargo, un procedimiento más sencillo pudiera ser el uso de la misma APC. En (Negrin et al, 2021a) se utilizó la función Ackley adaptada para probar 11 meta-heurísticas en la optimización discreta, para posteriormente optimizar una estructura real. En la Fig. 9a se puede apreciar la superioridad de BBO sobre las demás. Después, mediante el uso de las propias APCs, se ajustaron sus parámetros y se optimizó el diseño de la estructura real. En la Fig. 9b se puede apreciar la superioridad de BBO sobre GA y PSO.

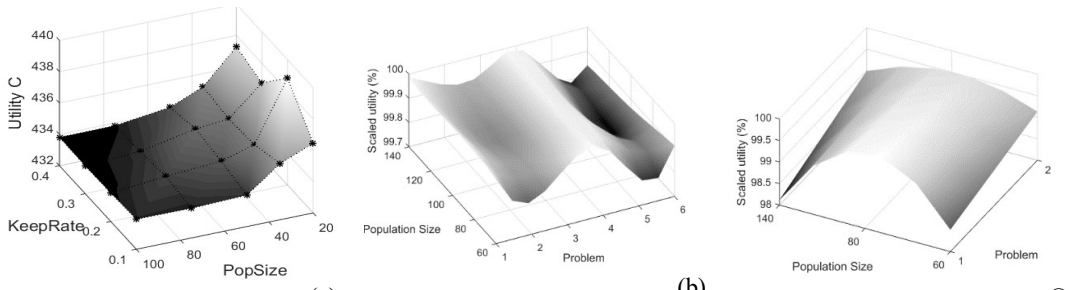
En (Negrin et al, 2021b) se ajustaron los parámetros de BBO utilizando la utilidad propuesta y los denominados paisajes de utilidad, los cuales nos permiten "visualizar" el comportamiento de un método en función de diferentes combinaciones de parámetros (ver Fig. 10a). Estos paisajes de utilidad se pueden utilizar, además, para desarrollar la estrategia de testear los métodos en modelos sencillos para sacar conclusiones sobre su

uso en modelos de gran complejidad, donde es imposible probar los métodos (ver Fig. 10b). En este trabajo se utilizaron 6 modelos simples para ajustar los parámteros de BBO y ser utilizado en la optimización del diseño de una estructura mucho más compleja. Los resultados simples sugerían que un tamaño de población de 60 era la mejor opción. Sin embargo, con el uso del paisaje de utilidad, se pudo comprobar que usar 80 como tamaño de población parecía más racional, puesto que brindaba resultados estables para todos los casos, mientras que 60 había sido la mejor solución para una mayor cantidad de modelos, pero bajaba su rendimiento en los más complejos (ver Fig. 10b). En la optimización de la estructura real se pudo comprobar que esta conclusión era cierta (ver Fig. 10c).



(a) (b)

Figura 9. Uso de las propias APC para medir el rendimiento de las meta-heurísticas, (a) optimización de la función Ackley adaptada utilizando 11 métodos, (b) optimización del caso de estudio real (Negrin et al, 2021a).



(a) (b) ©

Figura 10. Uso de los paisajes de utilidad para medir el rendimiento de un determinado método, (a) en el propio ajuste de parámteros, (b) utilizando 6 modelos sencillos para sacar conclusiones sobre su rendimiento, aquí se puede apreciar como un tamaño de población de 60 brinda buenos resultados pero baja su rendimiento en el problema 5, sin embargo 80 mantiene su estabilidad y en (c) se puede apreciar como ciertamente 80 era la mejor opción, ya que aquí se representa el rendimiento en la optimización de la estructura real (Negrin et al, 2021b).

El problema de usar la propia APC para medir el rendimiento de los métodos no permite la implementación de una estrategia automatizada para ajustar sus parámetros. Sin embargo, el uso de la utilidad propuesta brinda otras alternativas. Trabajos recientes se están enfocando en el desarrollo de una estrategia completamente automatizada de ajuste de parámetros. Esto requiere un análisis estadístico profundo para medir, además, la interacción entre los parámetros y sus valores. La Fig. 11 muestra un estudio reciente (Negrin et al, 2021d), donde se utilizó la utilidad para probar cuatro meta-heurísticas en cuatro modelos simulando problemas reales. La estrategia de ajuste fue bastante sencilla: probar todas las posibles combinaciones de los parámetros y sus valores, almacenar los resultados, y realizar el análisis estadístico correspondiente. Los gráficos de caja representan (box plots) el *box bottom whisker*, *box bottom*, *middle*, *top* and *top whisker* y denotan, respectivamente, el mínimo, percentil 25, mediana, percentil 75 y máximo valor de la utilidad de cada proceso. Se puede apreciar la gran superioridad de BBO sobre el resto, lo cual es uno de los grandes resultados de nuestras investigaciones: el descubrimiento de la gran habilidad de esta estrategia para enfrentar problemas de optimización discreta de optimización estructural.

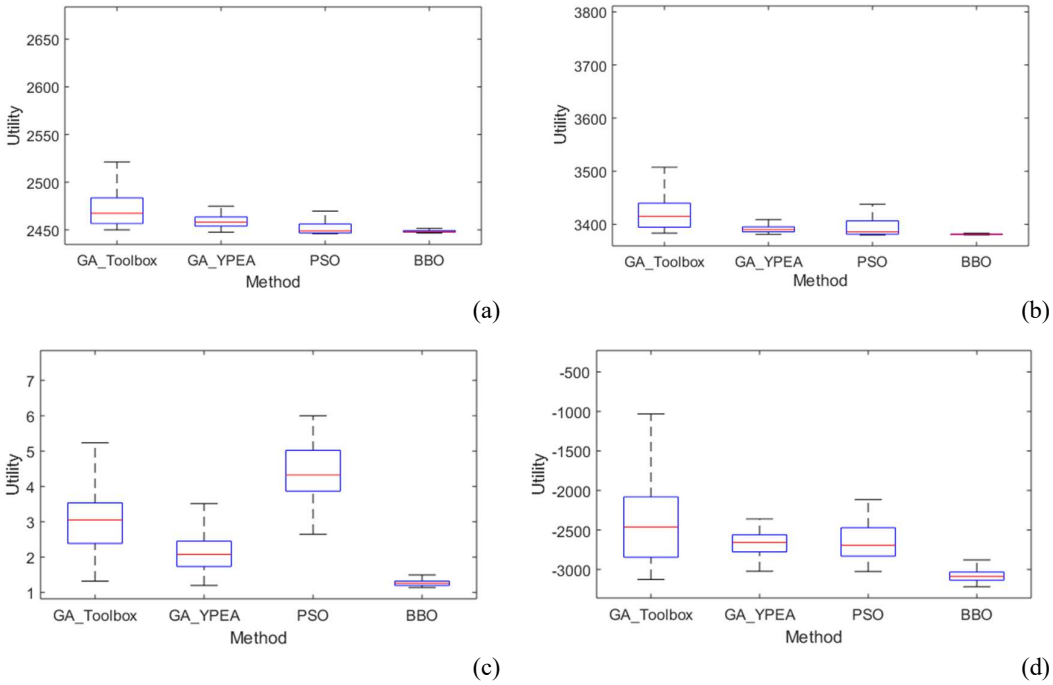


Figura 11. Análisis estadístico utilizando *box plots* de los resultados, medidos por la utilidad propuesta, de cuatro métodos aplicados a cuatro casos de estudio (o modelos sustitutivos) simulando la optimización de

estructuras reales (Negrin et al, 2021d).

3.3 Trabajo futuro

Partiendo de los resultados obtenidos, y de sus limitaciones, se proponen varias estrategias para superarlas en el futuro de esta línea de investigación.

- Existen grandes limitaciones referidas al consumo computacional de estos procesos. Por tanto, el uso de meta-modelos (modelos Kriging, Redes Neuronales, etc.) se pueden utilizar para mejorar, tanto la velocidad de convergencia como la calidad de los resultados, al "predecir" soluciones sin necesidad de evaluar el proceso convencional.
- El proceso de ajuste de parámetros resulta muy costoso. Este se puede mejorar de varias maneras. La primera es utilizar los meta-modelos para evitar la excesiva cantidad de evaluaciones necesarias para realizar esta tarea. Además, se pueden implementar otras alternativas como las simulaciones Monte Carlo para realizar un proceso de combinación de parámetros y sus valores mucho más eficiente.
- Se ha mencionado la gran habilidad de BBO para lidiar con problemas de optimización discreta. Sin embargo, esta estrategia se encuentra desarrollada de una forma muy simple, por lo que se pueden utilizar operadores de otros métodos (de selección, de cruzamiento), los cuales pudieran ser más eficientes en la búsqueda, mejorando aún más el rendimiento del método.
- Debido a que las (meta)heurísticas son procesos usualmente computacionalmente costos, se pudieran explorar otras alternativas, como la implementación del problema como de optimización continua (omitiendo o limitando el alcance de los resultados del diseño estructural, por ejemplo) para utilizar métodos clásicos que recorran el espacio de soluciones, y una vez encontrado un óptimo global, *explotar* la zona con una meta-heurística de búsqueda local considerando las soluciones prácticas que complejizan el problema.

4. Conclusiones

Debido a las particularidades de la optimización estructural, principalmente del diseño de estructuras de HA, el uso de (meta)heurísticas es la práctica más común. Estos métodos presentan una serie de operadores estocásticos que pueden ser regulados, o mejorados, para ajustar e incrementar su rendimiento. Sin embargo, estos procesos de optimización son usualmente muy costosos desde el punto de vista computacional,

especialmente cuando se utiliza un software de modelación, análisis y diseño estructural como motor de cálculo. En este trabajo se brindan una serie de alternativas para hacer viable el testeo de estos métodos, y así mejorarlos. Esto se puede lograr a través de modelos sustitutivos, es decir, modelos mucho más "ligeros" que permitan probar las estrategias de optimización en un proceso simple en vez de los reales de gran complejidad.

Una vez que es posible comprobar el funcionamiento de los métodos, se proponen varias metodologías para mejorar las estrategias de optimización. Las hibridaciones son una de estas, mediante las cuales se unen varios métodos para crear una herramienta de búsqueda mucho más robusta. Por otra parte, el ajuste de parámetros permite regular los operadores estocásticos en función del tipo de problema a resolver. En este trabajo se muestran varios ejemplos de aplicación y se comprueba cuan efectivas pueden ser las estrategias propuestas.

5. Referencias bibliográficas

1. Adenso-Díaz B. y Laguna M. (2006) Fine-tuning of algorithms using fractional experimental designs and local search. Operations Res. 54, 99–114. https://doi.org/10.1287/opre.1050.0243
2. Balaprakash P., Birattari M. y Stützle, T. (2007) Improvement strategies for the F-Race algorithm: sampling design and iterative refinement, in International workshop on hybrid metaheuristics, Dortmund, Germany, October, 2007 (Springer), 108–122.
3. Birattari M., Stützle T., Paquete L. y Varrentrapp K. (2002) A racing algorithm for configuring metaheuristics, in Proceedings of the 4th annual conference on genetic and evolutionary computation, New York, United States (New York, NY: Morgan Kaufmann Publishers Inc.), 11–18.
4. Castillo-Valdivieso P., Merelo J., Prieto A., Rojas I., y Romero G. (2002) Statistical analysis of the parameters of a neuro-genetic algorithm. IEEE Trans. Neural Netw. 13, 1374–1394. https://doi.org/10.1109/TNN.2002.804281
5. Coy S. P., Golden B., Runger G. y Wasil, E. (2001) Using experimental design to find effective parameter settings for heuristics. J. Heuristics 7, 77–97. https://doi.org/10.1023/a:1026569813391
6. Czarn A., MacNish C., Vijayan K., Turlach B., y Gupta R. (2004) Statistical exploratory analysis of genetic algorithms. IEEE Trans. Evol. Computat. 8, 405–421. https://doi.org/10.1109/tevc.2004.831262
7. Dillen W, Lombaert G y Schevenels M (2021) Performance Assessment of Metaheuristic Algorithms for Structural Optimization Taking Into Account the Influence of Algorithmic Control Parameters. Front. Built Environ. 7:618851. https://doi.org/10.3389/fbuil.2021.618851

8. Dillen, W., Lombaert, G., Voeten, N. y Schevenels, M. (2018) Performance assessment of metaheuristic algorithms for structural optimization taking into account the influence of control parameters, 6th International Conference on Engineering Optimization, Lisbon. https://doi.org/10.1007/978-3-319-97773-7_9
9. Eiben A.E. y Smit S.K. (2011a) Evolutionary Algorithm Parameters and Methods to Tune Them. In: Hamadi Y., Monfroy E., Saubion F. (eds) Autonomous Search. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-21434-9_2
10. Eiben A.E. y Smith S.K. (2011b) Parameter tuning for configuring and analyzing evolutionary algorithms, Swarm and Evol. Comp. pp. 19–31. https://doi.org/10.1016/j.swevo.2011.02.001
11. Holland J.H. 1975 Adaptation in Natural and Artificial Systems. University of MichiganPress, Ann Arbor, MI.
12. Kennedy y R. Eberhart (1995) Particle swarm optimization, Proceedings of ICNN'95 - International Conference on Neural Networks, pp. 1942-1948 vol.4. https://doi.org/10.1109/ICNN.1995.488968
13. Matlab (2015) The MathWorks, Inc.
14. Negrin I., Chagoyén E. y Negrin A. (2019a) Meta-heuristic optimization of structural sets of reinforced concrete, Revista Ingeniería de Construcción, Vol. 35 No 2, pp. 181-192. https://doi.org/10.4067/S0718-50732019000200181
15. Negrin I., Chagoyén E. y Negrin A. (2019b) Optimization of reinforced concrete plane frames using a hybridization of Genetic Algorithms and the Nelder-Mead algorithm (in Spanish), Rev. Ob. y Proy. Vol. 36, pp. 74-86. https://doi.org/10.4067/S0718-28132019000200074
16. Negrin I., Negrin, L.I. y Chagoyén E. (2020) Ajuste de parámetros de algoritmos genéticos: propuesta de método compuesto, Revista Cubana de Ciencias Informáticas, Vol. 14, No. 3, ISSN: 2227-1899 | RNPS: 2301
17. Negrin I., Chagoyén E. y Negrin A. (2021a) Parameter tuning in the process of optimization of reinforced concrete structures, DYNA, 88(216), pp. 87-95. https://doi.org/10.15446/dyna.v88n216.87169
18. Negrin, I., D. Roose, E. Chagoyén y G. Lombaert (2021b) Biogeography-Based Optimization of RC structures including static soil-structure interaction, Structural Engineering and Mechanics, (preprint). Available at: http://arxiv.org/abs/2103.05129
19. Negrin I. y Chagoyén E (2021c) Economic and environmental design optimization of RC structures: a comparative study. Structures. (En revisión)
20. Negrin, I., D. Roose y E. Chagoyén (2021d) Parameter tuning strategies for metaheuristic methods applied to discrete optimization of structural design, Investigación Operacional. (En revisión)
21. Payá, I., Yépes, V., González-Vidosa, F. y Hospitaler, A. Multiobjective Optimization of Concrete Frames by Simulated Annealing. Computer-Aided Civil

and Infrastructure Engineering 23 (2008) 596–610. http://dx.doi.org/10.1111/j.1467-8667.2008.00561.x

22. Rahmanian, I., Lucet Y. y Tesfamariam S. (2014) Optimal design of reinforced concrete beams: A review. Computers and Concrete, Vol.13, No.4 pp. 457-482. http://dx.doi.org/10.12989/cac.2014.13.4.457
23. Rardin R. L. y Uzsoy R. (2001) Experimental evaluation of heuristic optimization algorithms: a tutorial. J. Heuristics 7, 261–304. https://doi.org/10.1023/a:1011319115230
24. Simon D. (2008) Biogeography-Based Optimization, IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, VOL. 12(NO. 6.), pp. 702-713. https://doi.org/10.1109
25. Sivanandam, S.N. y Deepa, S.N. (2008), Introduction to Genetic Algorithms, Springer, Berlin Heidelberg, Germany.
26. Smith S. y Eiben A. (2010a) Parameter Tuning of Evolutionary Algorithms: Generalist vs. Specialist, Applications of Evolutionary Computation, EvoApplications 2010, Lecture Notes in Computer Science, vol. 6024. Springer, Berlin, Heidelberg, pp. 542-551. https://doi.org/10.1007/978-3-642-12239-2_56
27. Smith S. y Eiben A. (2010b) Beating the 'world champion' evolutionary algorithm, IEEE Congress on Evolutionary Computation, IEEE Computational Intelligence Society, IEEE Press, Barcelona, Spain, pp 1-8. https://doi.org/10.1109/CEC.2010.5586026
28. R. Van Mellaert, K. Mela, T. Tiainen, M. Heinisuo, G. Lombaert y M. Schevenels, (2018) Mixed-integer linear programming approach for global discrete sizing optimization of frame structures, Struct Multidisc Optim 57: 79-593.
29. Xu J.,Chiu S. Y., y Glover F. (1998) Fine-tuning a tabu search algorithm with statistical tests. Int. Trans. Oper. Res. 5, 233–244. https://doi.org/10.1111/j.1475-3995.1998.tb00117.x
30. Yuan B. y Gallagher M. (2007) Combining meta-EAs and racing for difficult EA parameter tuning tasks, in Parameter setting in evolutionary algorithms (Berlin, Germany: Springer), 121–142.