

# Optimización de Problemas Multiobjetivo de Ingeniería Civil con jMetal

Antonio J. Nebro   Gustavo R. Zavala   Juan J. Durillo   Francisco Luna

*Resumen*— Este artículo describe el uso del framework de optimización multiobjetivo jMetal para afrontar la resolución de problemas de ingeniería civil; en particular, lo que se ha hecho ha sido integrar un software Open Source para el diseño de estructuras, denominado Ebes, con jMetal. De esta forma los ingenieros civiles tienen a su disposición una herramienta que les permite diseñar estructuras que luego pueden ser optimizadas con metaheurísticas multiobjetivo atendiendo a varios criterios, como minimizar el peso y minimizar la deformación. Por otro lado, este tipo de problemas pueden ser objeto de estudios por parte de investigadores del área de las metaheurísticas, que pueden usarlos como casos de estudio. Tras presentar tanto jMetal como Ebes, se detalla la integración de ambas herramientas, se presentan tres casos de estudio y se proponen algunas líneas abiertas de investigación.

*Palabras clave*— Optimización Multiobjetivo, Diseño de Estructuras, Framework de Optimización

## I. INTRODUCCIÓN

El framework de optimización multiobjetivo jMetal [1] es una de las herramientas software más usadas dentro de la comunidad de investigadores interesados en problemas de optimización multiobjetivo [2]. Estos problemas se caracterizan por estar compuestos por dos o más funciones contrapuestas a optimizar a la vez, de forma que una mejora en una función implica un empeoramiento de las otras. El framework proporciona una cantidad destacable de algoritmos, operadores y problemas, siendo éstos principalmente problemas sintéticos, por lo que presentan un interés en la práctica muy reducido. Estos problemas incluyen la mayor parte los bancos de prueba más usados, como los ZDT [3], DTLZ [4], WFG [5] y la competición del CEC2009 [6].

Un motivo por el que jMetal es ampliamente usado es su diseño, que se basa en una arquitectura orientada a objetos donde cada entidad (algoritmo, operador, codificación, problema) está claramente definida, así como las relaciones entre ellas. Otras razones son su implementación en Java, el ser código abierto, la atención prestada a la calidad del código y la documentación disponible.

En este trabajo presentamos cómo jMetal se ha integrado con una herramienta del campo de la in-

geniería civil, que tiene la particularidad de no estar hecha en Java, sino en otro lenguaje (Visual Basic .Net). Dicha herramienta, denominada Ebes (de “Estructuras de barras espaciales”), es un proyecto de Gustavo R. Zavala que permite diseñar estructuras 2D y 3D compuestas de barras de distinto tipo (circulares, en forma de I, etc.) y de distinto material (acero, madera, etc.). Ejemplos de estas estructuras son puentes atirantados, naves industriales, gruas, etc.

Para situar el contexto del artículo vamos a enumerar los pasos a seguir para conseguir un diseño optimizado de una estructura de barras: primero se diseña el esquema de la estructura; segundo, se define el problema de optimización multiobjetivo correspondiente (indicando los objetivos a optimizar, típicamente reducir peso y aumentar la robustez); tercero, se aplica una técnica de optimización (metaheurística multiobjetivo en nuestro caso); cuarto, se visualiza la aproximación al frente de Pareto con las soluciones obtenidas; por último, se analizan dichas soluciones y se elige una de ellas.

Por lo que sabemos no existe ninguna herramienta que permita realizar todas estas tareas de forma integrada, y menos aún proporcionando metaheurísticas multiobjetivo para usarlas en el proceso. Esto nos ha llevado a desarrollar un software, llamado Ebes+jMetal, para cubrir esta carencia [7]. En el presente artículo no nos vamos a centrar en la perspectiva desde el punto de vista del ingeniero civil sino en la del investigador en metaheurísticas multiobjetivo. En un survey que realizamos sobre la aplicación de este tipo de técnicas a problemas de diseño estructural [8], tras analizar 51 artículos se llegó a la conclusión de que en la mayor parte de los estudios se han usado sobre todo algoritmos evolutivos, principalmente NSGA-II [9] y en menor medida SPEA2 [10], siendo la utilización de otras técnicas (PSO, búsqueda tabú, etc.) muy marginal. Por tanto, se puede afirmar que el desarrollo de algoritmos multiobjetivo para resolver estos problemas es un campo abierto, en el que se podrían realizar muchos estudios si se dispusiese de problemas reales que se pudiesen usar fácilmente. Uno de los objetivos de nuestro trabajo se centra en este punto.

El resto del artículo se estructura de la siguiente forma. En la sección II se presenta el tipo de problemas de diseño estructural que se abordan en este trabajo. La sección IV se dedica a describir las herramientas software Ebes y jMetal. La siguiente sección detalla la integración de dichas herramientas.

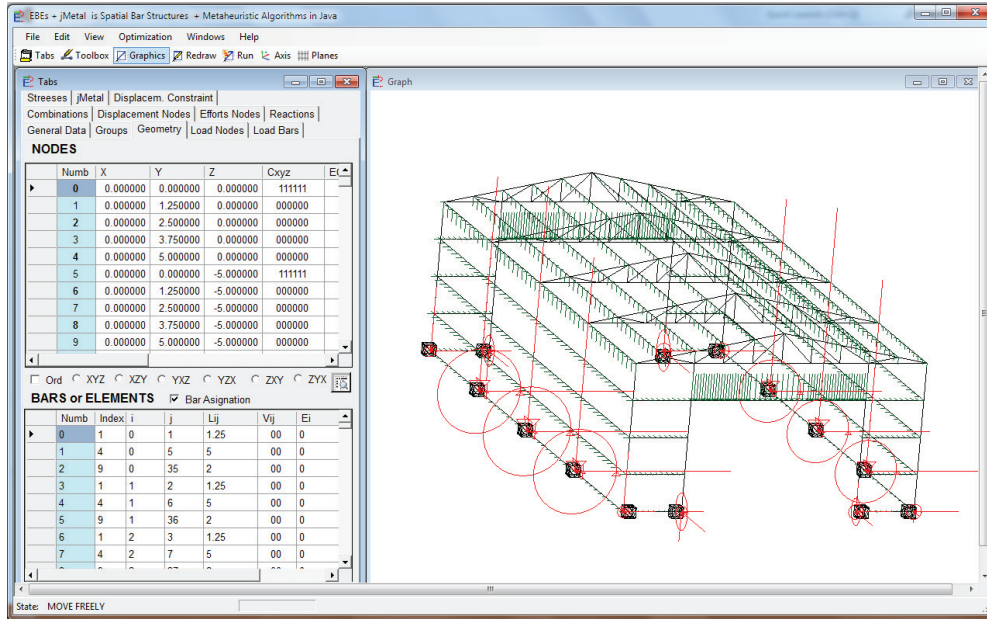
Departamento de Lenguajes y Ciencias de la Computación. Universidad de Málaga. E-mail: antonio@lcc.uma.es

Khaos Research Group. Universidad de Málaga. E-mail: grz@lcc.uma.es

Institute of Computer Science. University of Innsbruck. E-mail: juan@dps.uibk.ac.at

Departamento de Sistemas Informáticos y Telemáticos, Centro Universitario de Mérida. Universidad de Extremadura. E-mail: fluna@unex.es

Fig. 1. Diseño de una nave industrial con el software Ebes.

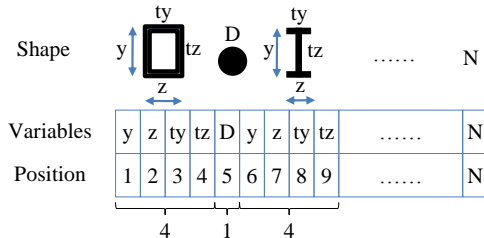


En la sección V se analizan tres casos de estudio y se dan ideas sobre líneas de investigación abiertas. Por último, la sección VI presenta las conclusiones del artículo.

## II. PROBLEMAS DE DISEÑO DE ESTRUCTURAS

Para ilustrar el tipo de problemas que estamos considerando tomemos como ejemplo la Fig. 1. Mediante una herramienta de diseño (Ebes en nuestro caso) se puede hacer un diseño de una estructura, como la nave industrial que se muestra. La nave tiene unas dimensiones prefijadas, y está compuesta por una serie de barras o elementos longitudinales que están unidas entre sí a través de unos nudos. El diseño debe cumplir una serie de restricciones para que, por ejemplo, la nave no se colapse por elegir elementos con tamaños insuficientes.

Fig. 2. Tipos de barras y representación de las variables.



Existen varios tipos de barras, como se muestra en la Fig. 2, cada uno de los cuales tiene diferentes dimensiones. Así, una barra circular está caracterizada por su diámetro, mientras que la que tiene forma de I tiene cuatro medidas. El problema de optimización se puede formular de forma informal cómo determinar cuáles son las dimensiones adecuadas de todas las barras que componen la estructura de for-

ma que se satisfagan las restricciones del problema y se satisfagan algunos criterios. Entre éstos vamos a considerar dos: minimizar el peso de la estructura (para abaratar el coste de construcción) y minimizar la deformación de la misma en determinados puntos (para aumentar su robustez). Estos objetivos son contrapuestos, luego se trata de un problema de optimización multiobjetivo (bi-objetivo en particular).

La Fig. 2 muestra cómo sería la representación de los individuos en vistas a usar una técnica metaheurística. Se trata de un problema de optimización continua, donde el conjunto de variables estará compuesto por los valores que caracterizan a todas las barras de la estructura. La complejidad del problema vendrá dada por el número de barras y por los tipos de las mismas (por ejemplo, la barras circulares son más simples que el resto).

Hay que reseñar que existen otros tipos de problemas de diseño de estructuras. En el survey [8] se propuso la siguiente clasificación:

1. Diseño de barras o elementos
  - Optimización de área (1.y)
  - Optimización de tamaño (2.y)
  - Optimización de forma (3.y)
  - Optimización de la topología de las secciones transversales (4.x)
2. Diseño de topología
  - Sin optimización (x.1)
  - Optimización discreta (x.2)
  - Optimización continua (3)

Dado que ambas clasificaciones no son excluyentes, se usa la notación (x.y) asociada a cada tipo, donde la x representa el tipo de diseño de barras (por lo que puede tomar, por tanto, valores entre 1 y 4) y la y el tipo de diseño topológico (pudiendo tomar los valores 1 y 2). La excepción es el caso (3),

Fig. 3. Variantes de problemas de diseño de barras: (1) Área sin tamaño, (2) tamaño, (3) forma, and (4) topología de la sección transversal

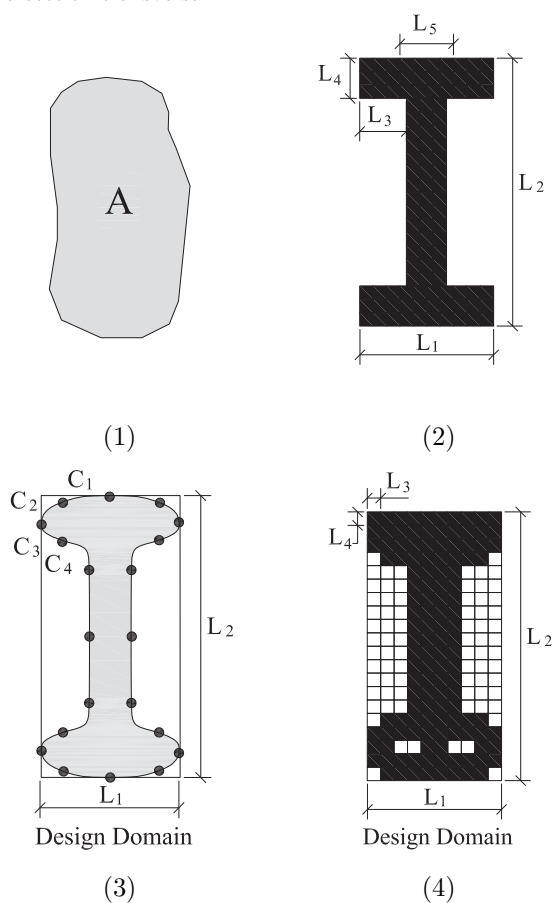


Fig. 4. Contenido de un fichero “.ebe”.

```
* EBES
GENERAL DATA
Name: Mobile_Bridge_25N_35B_8G_160rdZXY
Description: units of measure MN, m, MPa, MN/m3
Amount of NODES: 25
Amount of SUPPORTS: 4
Amount of ELEMENTS GROUPS: 8
Amount of ELEMENTS: 35
Groups of element same ELASTICITY: 2
Groups of element same SECTION: 8
Groups of element same INERTIA in Z axis: 8
Total load HYPOTHESIS: 1
Hypothesis COMBINATION of LOADS: 0
WEIGHT OWN OF ELEMENTS: True
STATIC OVERLOAD in elements: 28
STATIC LOAD in nodes: 0
LOAD for FREE FALL in node: 0
DISPLACEMENTS amount in nodes: 0
ASSEMBLY ERROR: 0
MAXIMUM DISPLACEMENTS to Nodes: 20
CUTTING Effect: False
SECOND ORDER: False
BUCKLING Check: True
Precision for EQUILIBRIUM of FORCES: 0.01
Precision for EQUILIBRIUM of MOMENTUMS: 0.01
NODES
0 0 0 0 111000 0 0 0 0 0 0
1 3 0 0 111000 0 0 0 0 0 0
2 3 3 0 000000 0 0 0 0 0 0
3 4 0.05 0 000000 0 0 0 0 0 0
4 5 0.1 0 000000 0 0 0 0 0 0
5 6 0.125 0 000000 0 0 0 0 0 0
6 7 0.15 0 000000 0 0 0 0 0 0
7 8 0.175 0 000000 0 0 0 0 0 0
```

de segundo orden no-lineal geométrico y la verificación al pandeo en elementos comprimidos o flexo-comprimidos.

Una vez que una estructura se ha diseñado, el resultado se puede almacenar en un fichero de texto (con extensión “.ebe”). La mayoría de los datos se almacenan como pares (clave: valor), excepto aquéllos relacionados a los nodos, grupos y barras (elementos), que se incluyen de forma tabular. Un ejemplo se muestra en la Fig. 4.

### B. jMetal

jMetal es un framework de optimización multiobjetivo con metaheurísticas implementado en Java [1]. El proyecto jMetal comenzó en 2006, cuando el análisis de los paquetes de optimización multiobjetivo existentes mostró que ninguno cumplía una serie de requisitos básicos, siendo los principales que ofreciesen una arquitectura orientada a objetos que fuese extensible y fácil de usar, y que los programas tuviesen un mínimo de calidad para facilitar su comprensión. Desde entonces jMetal se ha convertido en una de las herramientas más usadas por la comunidad de optimización multiobjetivo.

Las principales características de jMetal se incluyen en la siguiente lista:

- Algoritmos: 20 multiobjetivo y 4 monoobjetivo.
- Problemas: 5 benchmarks (ZDT, DTLZ, WFG, LZ09, CEC2009), problemas clásicos (5), con restricciones (6), combinatorios (2).
- Indicadores de calidad: hypervolume, spread, generational distance, inverted generational distance, additive epsilon.

ya que aquí se optimiza toda la estructura.

En la Fig. 3 se ilustran los tipos de problemas del primer grupo.

Los problemas que aquí se consideran pertenecen a la categoría (2.1), es decir, se busca optimizar el tamaño de las barras de la estructura, sin optimización de la topología de la misma, ya que ésta se ha definido previamente.

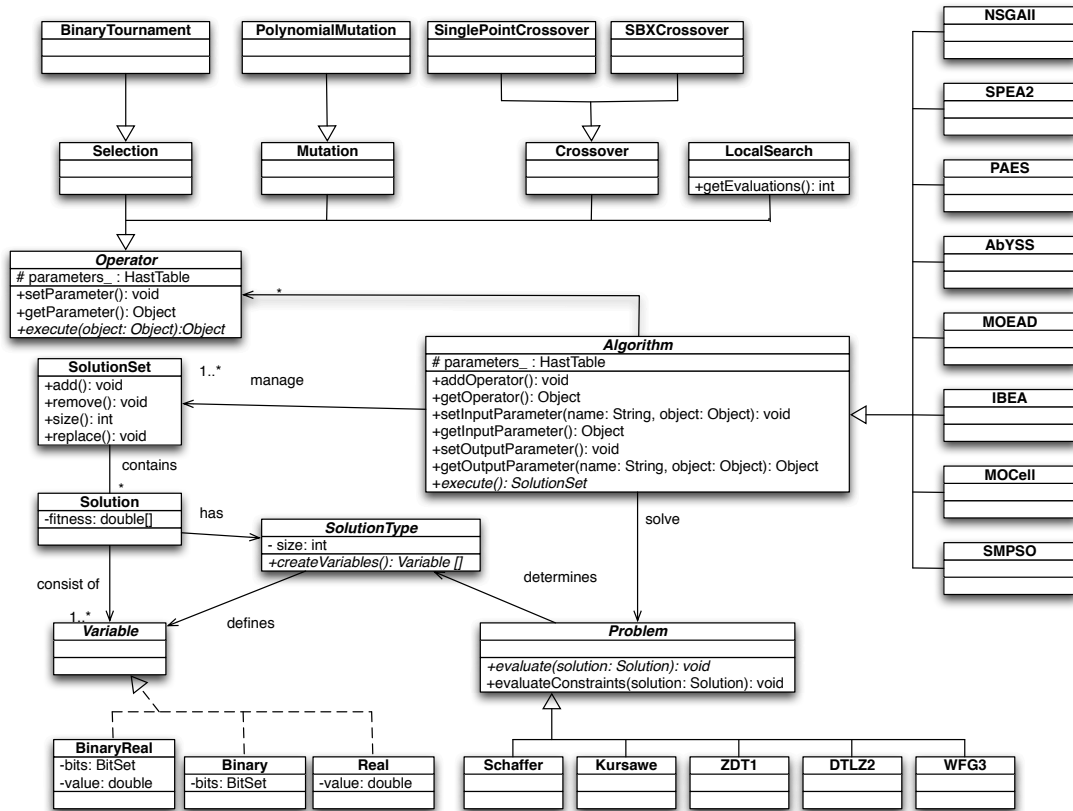
## III. DESCRIPCIÓN DE LAS HERRAMIENTAS

En esta sección se describen las dos herramientas software, Ebes y jMetal, que se han usado en este trabajo.

### A. Ebes

Ebes es una aplicación escrita en Visual Basic .Net que está enfocada al diseño, cálculo y análisis de estructuras de barras, tanto 2D como 3D. Tiene una interfaz gráfica (como se aprecia en la Fig. 1) que ofrece diferentes formas de visualización (escalado, desplazamiento, etc), permite la edición de nodos y barras, y muestra los datos de las estructuras (geometría, carga de nodos, carga de barras, agrupaciones de barras, etc.). Las capacidades de solución de Ebes también incluyen la relación elástica entre tensión-deformación para los materiales, el efecto

Fig. 5. Arquitectura de jMetal.



- Codificaciones: binaria, real, real codificada en binario, entera, permutación.

La arquitectura de jMetal se muestra en la Fig. 5, en la que se incluye el diagrama de clases en UML. Es una arquitectura orientada a objetos que es flexible y extensible, siendo fácil el añadir nuevos algoritmos, codificaciones, operadores y problemas.

#### IV. INTEGRACIÓN DE EBES Y jMETAL

Dado que Ebes y jMetal son dos sistemas independientes, que pertenecen a disciplinas distintas y que están implementados en lenguajes de programación diferentes, combinarlos planteaba una serie de retos.

La solución adoptada se basó en usar ficheros de texto para interconectar los dos componentes. En concreto, dado que los diseños de Ebes se almacenan en ficheros “.ebe”, la idea fue hacer que jMetal fuese capaz de leer este tipo de ficheros. Sin embargo, para computar los valores de las funciones objetivos hacía falta el método de cálculo que estaba incluido en Ebes, por lo que se optó por traducir ese código a Java e incluirlo en jMetal en forma de problema (llamado *Ebes.java*). Como resultado, dicho problema es con diferencia el más complejo que incluye el framework, conteniendo más de 3000 líneas de código.

Una vez que se disponía del problema se podía usar cualquiera de los algoritmos de optimización continua que existen en jMetal. Para facilitar la elección de la metaheurística, Ebes se extendió para po-

der elegir un algoritmo de entre los disponibles y así poder invocarlo. Este proceso se llevó a cabo mediante la ejecución de un fichero .jar que contiene el framework jMetal completo. La Fig. 6 muestra una captura de la ejecución del algoritmo MOCcell, que se ha seleccionado para optimizar el diseño de un puente levadizo.

La ejecución de una metaheurística multiobjetivo devuelve una aproximación al frente del Pareto del problema en forma de dos ficheros de texto: uno que almacena los valores de variables y otro que almacena los correspondientes valores de las funciones. Ambos ficheros contienen tantas líneas como soluciones devueltas por el algoritmo. El último paso de la integración consistió en hacer que Ebes pudiese leer esos ficheros y mostrar los frentes para que el ingeniero civil pudiese elegir las soluciones apropiadas. Este paso requirió además que los valores de las soluciones seleccionadas se incorporasen a la estructura para así poder disponer del diseño optimizado.

En la Fig. 7 se muestra un frente de soluciones obtenido con jMetal importado en Ebes. La interfaz gráfica permite seleccionar cualquier solución del frente, mostrar los valores de cada variable del problema y pasar esas variables a las barras. Este proceso aporta las medidas de la forma geométrica de cada sección de las barras, definiendo el volumen de las mismas. Con la selección de los datos se determinan automáticamente las características físicas para cada

Fig. 6. Ejecución de una metaheurística en Ebes+jMetal.

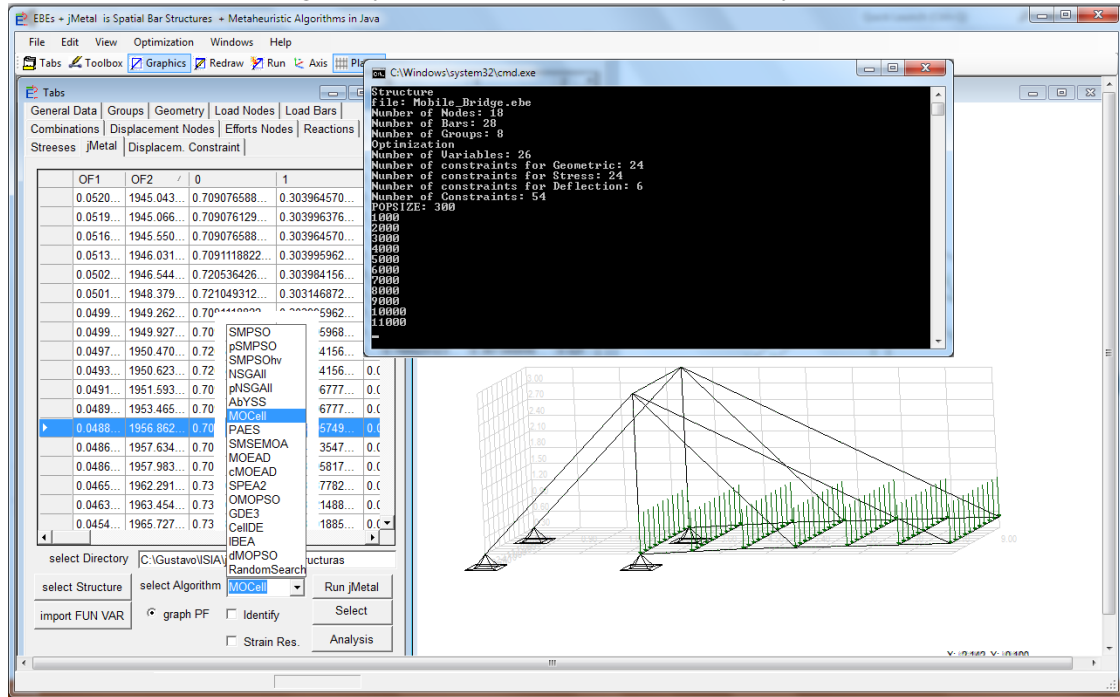
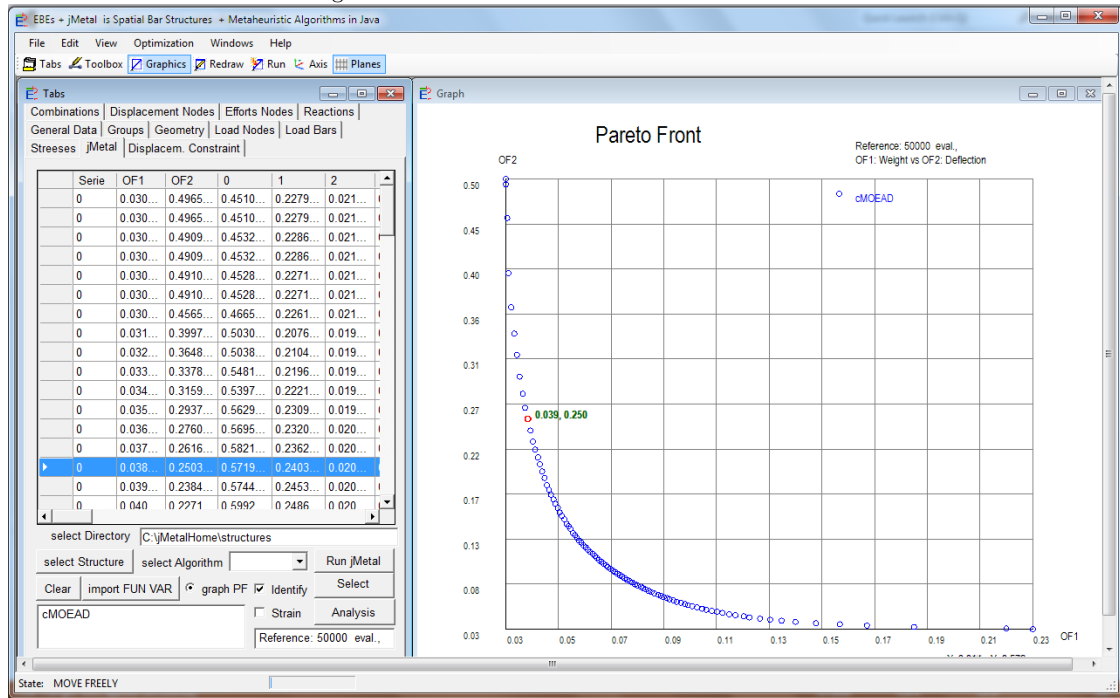


Fig. 7. Análisis de un frente de soluciones en Ebes.



tipo de sección, siendo éstas área, inercia, momento estático, etc. Con las variables junto a otros datos definidos previamente se obtiene un diseño tentativo completo de la estructura.

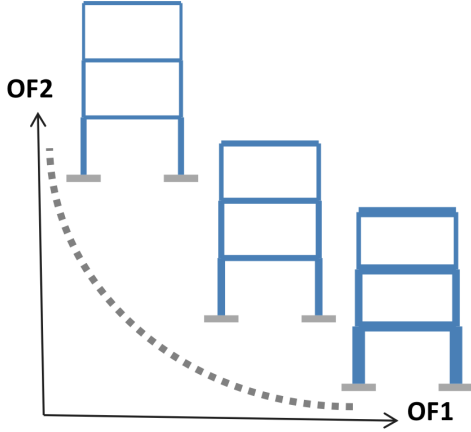
El resultado final es una herramienta Open Source denominada Ebes+jMetal, a la que se puede acceder a través un sitio Web<sup>1</sup>, que contiene enlaces al código fuente, un ejemplo de utilización y los frentes apro-

ximados de Pareto de varios problemas.

Para finalizar esta sección, la Fig. 8 muestra un ejemplo típico de aproximación al frente de Pareto y tres instancias concretas. Se observa claramente que el grosor de las estructuras cambia según las soluciones se obtienen hacia la parte derecha del frente. En este punto sería el experto en el problema (el ingeniero civil) el que deberá tomar la decisión acerca de qué solución será la más adecuada.

<sup>1</sup><http://ebesjmetal.sourceforge.net>

Fig. 8. Ejemplo de aproximación al frente de Pareto.



## V. CONSIDERACIONES PRÁCTICAS SOBRE EL PROCESO DE OPTIMIZACIÓN

El uso de Ebes+jMetal involucra las dos etapas de un proceso de optimización completo: la optimización en sí y la toma de decisiones. Mientras que la segunda fase está claramente en manos del experto del problema, la primera involucra a los investigadores en metaheurísticas. Desde el punto de vista de estos últimos hay mucho que aportar dentro del ámbito en el que se desarrolla el presente trabajo. Los problemas considerados están lejos de ser triviales y su optimización no es nada sencilla, y éste es un tema que queda lejos de los ingenieros civiles, que generalmente no serán expertos en metaheurísticas.

Si prestamos atención a la Fig. 6 podemos observar que existen más de 20 técnicas multiobjetivo en jMetal que se puede seleccionar para ser aplicadas en Ebes, luego estudiar qué algoritmo es el más prometedor para resolver un problema dado es tema abierto, como lo es determinar cuáles son las mejores configuraciones de los mismos.

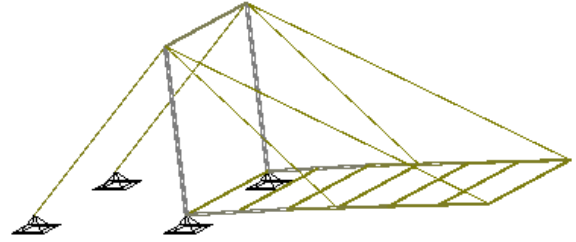
Un aspecto práctico a considerar es que la optimización de los problemas se puede hacer sin la integración con Ebes; basta con tener el fichero “.ebe” que describa la estructura para poder proceder como si fuera un problema más. Como consecuencia, no hace falta la intervención del experto en el problema y el fin a perseguir será obtener aproximaciones a los frentes de Pareto de los problemas de diseño que sean lo más precisos posible, atendiendo a los ya establecidos criterios de convergencia y diversidad. Es decir, se podrían usar los problemas de diseño estructural a modo de *benchmark* real para aplicar en ellos nuevos avances en metaheurísticas multi-objetivo.

Otro tema a tener en cuenta es el coste computacional para evaluar las funciones objetivo. Es obvio que tiempo de cálculo dependerá del número y tipo de las barras. Para dar una idea concreta sobre el orden de magnitud de dicho coste vamos a considerar tres ejemplos, que se corresponden con tres diseños de distinto tamaño de un puente atirantado (*cable-stayed bridge*).

### A. Puente 25N\_35B

El puente que se muestra en la Fig. 9 puede considerarse como una estructura relativamente simple. Está compuesto de 25 nodos y 35 barras, por lo que lo denominamos como puente 25N\_35B. Tiene dos columnas que soportan parte del peso del tablero de tránsito, y el resto del peso es soportado por los cables anclados detrás de las columnas. La longitud del puente es de 9 metros.

Fig. 9. Puente 25N\_35B.



Si se configura el algoritmo NSGA-II para hacer 150.000 evaluaciones el tiempo de ejecución está en torno a seis minutos en un MacBook Pro con procesador Intel Core i7 a 2.2GHz, 8GB de RAM (DDR3 a 1333MHz), disco SSD de 256GB, sistema operativo MacOS X 10.9.2 y Java JDK versión 1.8.0. Este tiempo no es muy elevado y permitiría hacer un estudio que involucrase a varios algoritmos sin la necesidad de muchos recursos computacionales.

La Fig. 10 resume los resultados obtenidos al hacer una comparativa que involucra a siete algoritmos representativos; se muestran los boxplots de los valores del indicador de calidad Hypervolume [11] sobre 30 ejecuciones independientes de cada algoritmo. Además hemos podido hacer un estudio de parámetros para ajustar los algoritmos, pero para ello hemos tenido que usar un cluster de más de 400 cores para terminar todas las pruebas en un par días, dado el elevado número de combinaciones de valores de los parámetros que se tuvo en cuenta.

### B. Puente 133N\_221B

El puente de la Fig. 11, denominado puente 133N\_221B, se caracteriza por tener 133 nodos y 221 barras y una longitud total de 44 metros. Dada la simetría del puente, para simplificar su optimización se ha considerado la mitad del mismo.

El tiempo requerido para ejecutar NSGA-II en este caso asciende a 4,4 horas en el mismo ordenador usado en el puente anterior. Es un tiempo bastante elevado que prácticamente obliga a hacer uso de un sistema paralelo si se quiere hacer un estudio con varios algoritmos.

### C. Puente 837N\_1584B

El puente de mayor tamaño que hemos optimizado se muestra en la Fig. 12. Tiene 837 nodos, 1584 ba-

Fig. 10. Comparativa sobre el puente 25N\_35B.

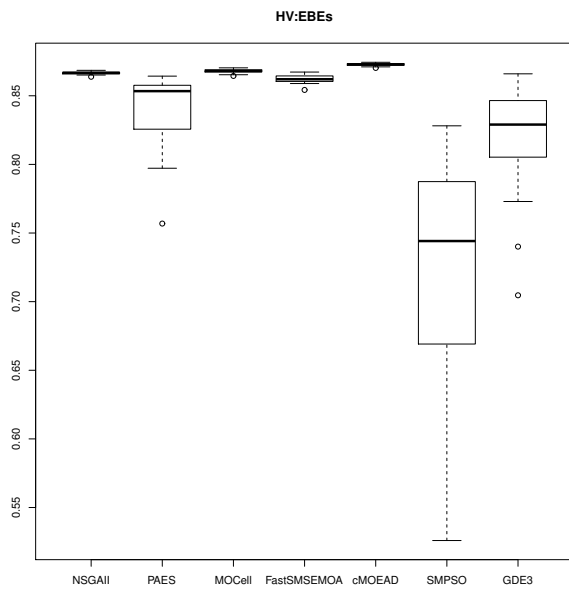
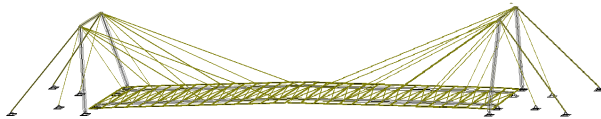
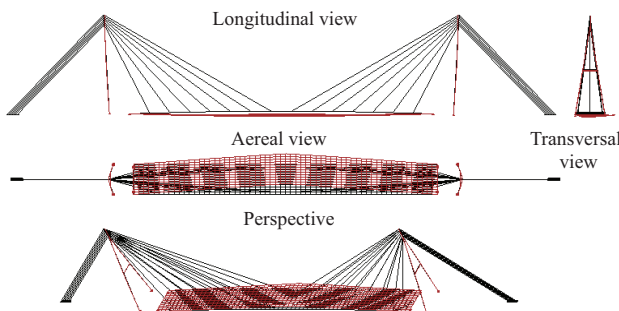


Fig. 11. Puente 133N\_221B.



rras y una longitud total de 162 metros. Para dar una idea de la complejidad de este problema, en nuestro caso fue necesario usar el cluster anterior para poder completar una ejecución del algoritmo NSGA-II en unas 10 horas.

Fig. 12. Puente 837N\_1584B.



Este puente, junto con el anterior, fue sujeto de estudio en el artículo [12].

#### D. Discusión

Los tres puentes comentados dan una idea aproximada de posibles vías de investigación. En el caso de la instancia más pequeña, en la que hacer una ejecución aislada tiene un coste computacional relativa-

mente bajo, hacer un estudio experimental típico que involucre a varios algoritmos puede requerir usar un clúster de al menos algunas decenas de máquinas para tener los resultados en un tiempo razonable. En el caso de la instancia mediana ya estaríamos hablando de necesitar algunos cientos de máquinas para hacer un estudio similar, y si se intentan abordar problemas como el puente de mayor tamaño entonces usar un elevado número de procesadores es poco menos que obligatorio para hacer una única ejecución.

Parece claro que, por un lado, estudiar qué algoritmos son más eficaces para problemas de dimensión reducida o media tendría mucha utilidad para los ingenieros civiles, que tendrían al menos una orientación sobre qué técnicas serían las más prometedoras para resolver un problema concreto. En este sentido también sería de interés determinar qué configuración de parámetros proporcionaría los mejores resultados a un algoritmo determinado.

Por otro lado, dado que el coste de evaluar las funciones objetivo puede ser considerable, reducir el número de éstas para proporcionar resultados aceptables al ingeniero civil en el menor tiempo posible también sería un avance importante. Aplicar técnicas de búsqueda local podría ser una estrategia a considerar aquí, así como el uso adaptativo de diferentes operadores de variación durante el proceso de búsqueda.

Por último, diseñar metaheurísticas que saquen partido de conocimiento del problema, probablemente definiendo operadores específicos, sería un avance significativo en este área.

## VI. CONCLUSIONES

En este artículo hemos presentado la integración del framework de optimización multiobjetivo jMetal con un software de diseño de estructuras civiles denominado Ebes. El resultado es una herramienta que permite al ingeniero civil utilizar los algoritmos incorporados en jMetal de forma que los frentes resultantes se pueden analizar con Ebes y así poder analizar las soluciones de compromiso obtenidas y poder elegir la que más se ajuste a las necesidades o a los requisitos que se hayan establecido.

Los problemas de ingeniería que se pueden abordar son problemas de diseño estructural, en los que la topología ya está definida y se trata de determinar las dimensiones de cada barra para minimizar el peso y la deformación de la construcción resultante. Mediante Ebes+jMetal se pueden abordar las distintas etapas que hay que llevar a cabo, empezando por el diseño inicial, siguiendo por el proceso de optimización y terminando por la toma de decisiones por parte del experto en el dominio.

Dado que el proceso de optimización no requiere necesariamente el uso de Ebes se pueden realizar estudios de optimización usando únicamente jMetal, lo que abre muchas posibilidades a investigadores que trabajen en el campo de la optimización multiobjeti-

vo con metaheurísticas. Para ilustrar las dificultades computacionales que se pueden presentar a la hora de resolver estos problemas se han tomado tres instancias de distinta complejidad y se han ofrecido los tiempos obtenidos al resolverlos con un ordenador concreto.

En base a estos ejemplos se han presentado algunas ideas sobre temas abiertos que se pueden aplicar a los problemas de diseño estructural, lo que daría lugar a resultados de gran interés en ese dominio. En el estado del arte actual la presentación de nuevas propuestas algorítmicas enfocadas a resolver estos problemas de forma eficiente es un campo por explorar.

#### AGRADECIMIENTOS

Este trabajo ha sido financiado por los proyectos TIN2011-25840 (Ministerio de Ciencia e Innovación) y P11- TIC-7529 and P12-TIC-1519 (Plan Andaluz de Investigación, Desarrollo e Innovación).

#### REFERENCIAS

- [1] Juan J. Durillo and Antonio J. Nebro, “jmetal: A java framework for multi-objective optimization,” *Advances in Engineering Software*, vol. 42, pp. 760–771, 2011.
- [2] I. Giagkiozis, R.J. Lygoe, and P.J. Fleming, “Liger: an open source integrated optimization environment,” in *Genetic and Evolutionary Computation Conference, GECCO '13, Amsterdam, The Netherlands, July 6-10, 2013, Companion Material Proceedings*, 2013, pp. 1089–1096.
- [3] E. Zitzler, K. Deb, and L. Thiele, “Comparison of multiobjective evolutionary algorithms: Empirical results,” *Evolutionary Computation*, vol. 8, no. 2, pp. 173–195, Summer 2000.
- [4] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, “Scalable test problems for evolutionary multiobjective optimization,” in *Evolutionary Multiobjective Optimization. Theoretical Advances and Applications*, Ajith Abraham, Lakhmi Jain, and Robert Goldberg, Eds., pp. 105–145. Springer, USA, 2005.
- [5] S. Huband, L. Barone, R.L. While, and P. Hingston, “A scalable multi-objective test problem toolkit,” in *Third International Conference on Evolutionary Multi-Criterion Optimization, EMO 2005*, C.A. Coello, A. Hernández, and E. Zitzler, Eds. 2005, vol. 3410 of *Lecture Notes in Computer Science*, pp. 280–295, Springer.
- [6] Q. Zhang and P. N. Suganthan, “Special session on performance assessment of multiobjective optimization algorithms/cec 09 moea competition,” May 2009.
- [7] Gustavo R. Zavala, Antonio J. Nebro, Juan J. Durillo, and Francisco Luna, “Integrating a multi-objective optimization framework into a structural design software,” *Advances in Engineering Software*, vol. 76, no. 0, pp. 161 – 170, 2014.
- [8] G.R. Zavala, A.J. Nebro, F. Luna, and C.A. Coello Coello, “A survey of multi-objective metaheuristics applied to structural optimization,” *Structural and Multidisciplinary Optimization*, vol. 49, no. 4, pp. 1–22, 2013.
- [9] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, “A fast and elitist multiobjective genetic algorithm: NSGA-II,” *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.
- [10] E. Zitzler, M. Laumanns, and L. Thiele, “SPEA2: Improving the strength pareto evolutionary algorithm,” in *EUROGEN 2001. Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems*, K. Giannakoglou, D. Tsahalis, J. Periaux, P. Papailou, and T. Fogarty, Eds., Athens, Greece, 2002, pp. 95–100.
- [11] E. Zitzler and L. Thiele, “Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach,” *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 4, pp. 257–271, 1999.
- [12] F. Luna, Gustavo R. Zavala, Antonio J. Nebro, J.J. Durillo, and Carlos A. Coello Coello, “Distributed multi-objective metaheuristics for real-world structural optimization problems,” *The Computer Journal*, 2014, In press.