

UNIVERSIDAD POLITÉCNICA DE VALENCIA
Departamento de Ingeniería Mecánica y de Materiales



TESIS DE MASTER

**EVALUACIÓN DE ESTRATEGIAS DE
PARAMETRIZACIÓN GEOMÉTRICA Y DE
LA MEDIDA DE CONTROL DE LA
VELOCIDAD DE CONVERGENCIA EN
METODOLOGÍAS DE OPTIMIZACIÓN DE
FORMA BASADAS EN ALGORITMOS
EVOLUTIVOS.**

Presentada por: Dña. Eva M. Sánchez Orgaz

Dirigida por: Dr. D. José Albelda Vitoria

Valencia, Septiembre de 2011





RESUMEN

La presente Tesis de Máster consistirá en la evaluación de diferentes estrategias para definir paramétricamente la geometría en problemas de optimización de forma de componentes mecánicos. Este estudio se complementará con la evaluación de los métodos de optimización disponibles en el departamento basados en algoritmos evolutivos, en términos de la velocidad de convergencia, teniendo en cuenta sus parámetros de control. Se prestará especial atención al efecto de la precisión en la evaluación de la función objetivo.

La comparación se realizará en base a ciertos problemas de referencia previamente establecidos.

Palabras clave: optimización de forma, algoritmos evolutivos, velocidad de convergencia, función objetivo, parametrización de la geometría.





ABSTRACT

This Master Thesis will consist of evaluating different strategies to define the geometry parametrically in shape optimization problems of mechanical components. This study will complement the evaluation of optimization methods available in the department based on evolutionary algorithms, in terms of speed of convergence, taking into account the parameters of control. Pay special attention to the effect of precision in the evaluation of the objective function.

The comparison was made based on certain pre-established benchmark problems.

Keywords: shape optimization, evolutive algorithms, convergence speed, objective function, parametrization geometry.





RESUM

La present Tesi de Màster consistirà en l'avaluació de diferents estratègies per a definir paramètricament la geometria en problemes d'optimització de forma de components mecànics. Aquest estudi es complementarà amb l'avaluació dels mètodes d'optimització disponibles al departament basats en algorismes evolutius, en termes de velocitat de convergència, tenint en compte els seus paràmetres de control. Es prestarà especial atenció al efecte de la precisió en l'avaluació de la funció objectiu.

La comparació es realitzarà en base a certs conceptes de referència previament establerts.

Paraules clau: optimització de forma, algorismes evolutius, velocitat de convergència, funció objectiu, parametrització de la geometria.





AGRADECIMIENTOS

En primer lugar, me gustaría dar las gracias a Pepe, director de mi Tesis de Máster, por su paciencia, dedicación y por los conocimientos transmitidos.

Querría extender mi agradecimiento a mis compañeros de departamento y de máster, por todo su apoyo.

A todos los profesores y personal administrativo del área de ingeniería mecánica ya que conforman un grupo excelente. Y a Fede, técnico informático del departamento, por la ayuda prestada.

Por último, y no por ello menos importante, a mi familia, por estar siempre ahí.

Septiembre de 2011





Índice general

| | |
|--|-----------|
| Resumen | i |
| Abstract | iii |
| Resum | v |
| Agradecimientos | vii |
| Índice general | ix |
| 1 INTRODUCCIÓN | 1 |
| 2 ANTECEDENTES | 3 |
| 3 CONCEPTOS PREVIOS Y PLANTEAMIENTO | 7 |
| 3.1 INTRODUCCIÓN | 7 |
| 3.2 OPTIMIZACIÓN | 7 |
| 3.2.1 <i>Introducción a los problemas de optimización</i> | 7 |
| 3.2.2 <i>Clasificación de los algoritmos de optimización</i> | 9 |
| 3.2.3 <i>Algoritmos estocásticos</i> | 11 |
| 3.3 OPTIMIZACIÓN DE FORMA | 23 |
| 3.4 MOGA- II: DESCRIPCIÓN Y PARÁMETROS DE CONTROL | 26 |
| 3.4.1 <i>Descripción de algoritmo</i> | 27 |
| 3.4.2 <i>Parámetros</i> | 28 |
| 3.4.3 <i>Tratamiento de las restricciones</i> | 30 |
| 3.5 PROBLEMAS DE REFERENCIA | 30 |
| 3.5.1 <i>Justificación</i> | 30 |
| 3.5.2 <i>Viga</i> | 31 |
| 3.5.3 <i>Cilindro</i> | 33 |
| 3.5.4 <i>Puente</i> | 36 |
| 4 PARÁMETROS DE CONTROL | 39 |
| 4.1 INTRODUCCIÓN | 39 |
| 4.2 AJUSTE DE LOS PARÁMETROS | 40 |
| 4.2.1 <i>Viga</i> | 42 |
| 4.2.2 <i>Cilindro</i> | 49 |
| 4.3 CONCLUSIONES | 55 |
| 5 FUNCIÓN OBJETIVO | 57 |
| 5.1 INTRODUCCIÓN | 57 |



| | | |
|-----------|--|------------|
| 5.2 | COMPARACIÓN DE RESULTADOS | 57 |
| 5.2.1 | <i>Viga</i> | 57 |
| 5.2.2 | <i>Cilindro</i> | 61 |
| 5.3 | PUENTE..... | 63 |
| 5.4 | CONCLUSIONES..... | 65 |
| 6 | PARAMETRIZACIÓN | 67 |
| 6.1 | INTRODUCCIÓN..... | 67 |
| 6.2 | DEFINICIÓN POR PUNTOS. NÚMERO DE COORDENADAS..... | 68 |
| 6.2.1 | <i>Viga</i> | 68 |
| 6.2.2 | <i>Cilindro</i> | 75 |
| 6.3 | PUENTE..... | 78 |
| 6.4 | ACTUALIZACIÓN SECUENCIAL | 82 |
| 6.5 | CONCLUSIONES..... | 83 |
| 7 | CONTROL DE VELOCIDAD DE CONVERGENCIA | 85 |
| 7.1 | INTRODUCCIÓN..... | 85 |
| 7.2 | AJUSTE DE LA CURVA DE CONVERGENCIA..... | 85 |
| 7.2.1 | <i>Ajustes de la curva de convergencia para la viga y el cilindro</i> | 87 |
| 7.2.2 | <i>Ajustes de la curva de convergencia para el puente</i> | 90 |
| 7.3 | ESTRATEGIA DE ESTIMACIÓN DE LA CURVA DE CONVERGENCIA..... | 92 |
| 7.4 | ESTRATEGIAS DE RESTRICCIÓN DE ERROR DE DISCRETIZACIÓN EN FUNCIÓN DE LA CURVE DE CONVEGENCIA DE OPTIMIZACIÓN..... | 92 |
| 7.4.1 | <i>Estrategias utilizadas para el problema del cilindro</i> | 93 |
| 7.4.2 | <i>Estrategia de optimización para el puente</i> | 97 |
| 7.5 | CONCLUSIONES..... | 97 |
| 8 | RESULTADOS OBTENIDOS, CONCLUSIONES Y TRABAJOS FUTUROS..... | 99 |
| 8.1 | CONCLUSIONES..... | 99 |
| 8.2 | DESARROLLOS FUTUROS..... | 100 |
| 9 | ANEXO I: PLANTEAMIENTO DE UN PROBLEMA EN MODEFRONTIER | 103 |
| 10 | ANEXO II: INTERFAZ MODEFRONTIER- GFEM..... | 107 |
| 11 | BIBLIOGRAFÍA..... | 119 |



1 INTRODUCCIÓN

En la presente tesina se va a realizar un estudio de algunas herramientas que permitirán optimizar la forma de componentes estructurales mediante un algoritmo genético de forma más rápida y eficiente.

Las herramientas a desarrollar serán la búsqueda de diferentes estrategias de parametrización de la geometría y de disminución del error de discretización. Y por otro lado, la utilización de distintas funciones objetivo para ver como afectan a la velocidad de convergencia.





2 ANTECEDENTES

El término algoritmo evolutivo (EA) se aplica a un tipo de métodos de optimización estocásticos que simulan el proceso de la evolución natural. Los orígenes de los EA se remontan a los últimos años de la década de los 50 del siglo pasado, y desde 1970 se han propuesto diversas metodologías, principalmente, algoritmos genéticos (GA), programación evolutiva y estrategias de evolución. Todas estas aproximaciones operan sobre un conjunto de soluciones candidatas. Utilizando grandes simplificaciones, este conjunto es subsecuentemente modificado por dos principios básicos: selección y variación. Mientras la selección mimetiza la competencia por la reproducción y los recursos para la supervivencia, el otro principio, la variación, imita la capacidad natural de crear nuevas vidas mediante la recombinación y la mutación.

A pesar de que los mecanismos fundamentales son simples, se ha probado que estos algoritmos son mecanismos de búsqueda generales, robustos, y potentes. En particular, poseen varias características que son deseables para resolver problemas con alguna de las siguientes características:

1. Múltiples objetivos contrapuestos.
2. Espacios de búsqueda amplios y complejos.

Como resultado, se ha propuesto y aplicado numerosas variantes de este tipo de algoritmos a varios tipos de problemas desde 1985.

Si particularizamos la optimización mediante EA para el caso de la optimización de forma, se ha de seleccionar un método de representación de la forma de los individuos en la población de un GA.

Cuando se utiliza un GA o alguna otra técnica de optimización de parámetros para optimización de forma, es necesario representar la forma de cada individuo de una población en forma paramétrica. La elección del método de representación es importante para la eficiencia y la efectividad del algoritmo genético en el cual se basa la optimización de forma, así como para su robustez.



Algunas técnicas de representación de formas geométricas individuales en una población mediante cromosomas de cada individuo, según la referencia [16], son las que utilizan curvas de Bezier y B-splines.

Por otro lado, aparece la necesidad de unir el diseño del modelo o descripción geométrica y el análisis del modelo. El análisis isogeométrico ofrece un potencial camino muy prometedor en este sentido. Con la discretización basada en B-splines racionales no uniformes (NURBS) el modelo de análisis representa la geometría estructural exactamente. Además, los NURBS permiten un control de la geometría con contornos suaves. Son empleados en sistemas CAD, y también ampliamente utilizados en el contexto de diseño de forma óptima para definir la representación geométrica y las variables de diseño. Con esta aproximación isogeométrica a la optimización de forma, el modelo de análisis se funde inherentemente con el modelo de diseño, omitiendo la típica interacción entre ambos.

También es destacable que, la optimización de forma de estructuras mediante el método de elementos finitos ha conocido en los últimos años un desarrollo muy importante. Por una parte, este desarrollo ha sido incentivado por las demandas tecnológicas, cada día más exigentes, en la concepción de una estructura y por otra, ha sido posible gracias al aumento del poder de cálculo experimentado por los ordenadores.

En elasticidad lineal, la calidad de la solución obtenida por elementos finitos depende principalmente del error de discretización lo cual también influye fuertemente en el proceso de optimización de forma.

Esta dependencia trae como consecuencia en muchos casos que la solución obtenida por el proceso de optimización de forma varíe según la discretización inicial del modelo.

Para reducir la dependencia de la solución con respecto a la discretización inicial, se propone un procedimiento automático de optimización de forma con control del error de la solución elementos finitos.

Un número importante de trabajos se han desarrollado en los últimos años. Ellos se diferencian por los métodos de estimación del error relativo, en la manera de generar y controlar el mallado, pero sobre todo en la gestión del nivel de error admisible para los mallados de las etapas intermedias de la optimización de forma.

Este último punto, es decir, el control de los mallados intermedios, es decisivo desde el punto de vista de la eficacia del método. La idea aquí es retardar



lo más posible la adaptación de los mallados con el fin de disminuir los tiempos de cálculo cuando aún se está lejos de la forma óptima de la estructura.

En la referencia [14] se ha estudiado la distribución de errores debido a que no se completa la convergencia del procedimiento de optimización. Los pesos óptimos obtenidos para estructuras alares de transporte civil de alta velocidad (HSCT) calculado por el programa de optimización estructural GENESIS contienen errores sustanciales. Es posible reducir el error estrechando en criterio de convergencia para lograr optimizaciones de mayor calidad. Para estimar el nivel de incertidumbre general de la optimización de baja calidad, el error de optimización se estimó como puntos constituyentes de un diseño experimental en el espacio de diseño del HSCT. Dado que el error de optimización incompleta se comete por una única parte, el modelo de la distribución de Weibull se ajusta al error estimado y se logra un buen ajuste para datos de baja y media calidad. Las distribuciones de ajuste del error pueden proporcionar una guía para la selección del criterio de convergencia, y pueden resultar útiles en la especificación de distribuciones de la incertidumbre de las variables en una optimización robusta.

Con la optimización cada vez más utilizada en aplicaciones de ingeniería, se requieren series de ejecuciones, lo que puede requerir un demasiado coste computacional demasiado elevado para lograr una convergencia precisa.

Por último, la elección de la función objetivo (FO) más adecuada para cada problema influirá en la convergencia a la solución de cada problema. En optimización de forma la FO más frecuente es tratar de obtener el mínimo volumen, no obstante, en la bibliografía ([18]) se han hallado otras como:

1. Conformidad:

$$J(w) = \int_w A \cdot e(u) : e(u) dx = \int_w A^{-1} \cdot \sigma : \sigma dx = \int_{\partial w} f \cdot u ds = c(w)$$

Se trata de una medida global de la rigidez

2. Criterio de Mínimos Cuadrados:

$$J(w) = \left\{ \int_w k(x) \cdot |u(x) - u_0(x)|^2 dx \right\}^{\frac{1}{\alpha}}$$



Donde u_0 es un desplazamiento objetivo dado, $\alpha \geq 2$ y k un multiplicador dado \Rightarrow Sistemas de optimización micromecánicos (MEMS).

3. FO dependiente de la tensión: p.e. $J(w) = \int_w (\sigma)^2$



3 CONCEPTOS PREVIOS Y PLANTEAMIENTO

3.1 INTRODUCCIÓN

En primer lugar, en este apartado se detallarán los principios básicos de la optimización, así como los principales algoritmos y sus parámetros. Y posteriormente se particularizarán para optimización de forma, haciendo especial hincapié en Moga-II, algoritmo utilizado por modeFRONTIER, programa que se ha utilizado en este trabajo de investigación.

A continuación se especificarán los problemas de referencia utilizados y las características particulares de cada uno de ellos.

3.2 OPTIMIZACIÓN

3.2.1 Introducción a los problemas de optimización

En un problema de optimización se pretende tomar una decisión que maximice o minimice un criterio determinado (costes, tiempo, riesgo, error, ganancias, velocidad, eficiencia, etc.). A menudo esta decisión se encuentra condicionada por diferentes restricciones.

Se pueden diferenciar dos tipos de problemas de optimización, el problema de maximización y de minimización. El problema de minimización se puede definir de la siguiente forma: Dado un espacio de decisión (también denominado espacio de búsqueda) X , un espacio objetivo Y (valores objetivo) y un conjunto de funciones objetivo $f: X \rightarrow Y$ sujeto a unas restricciones g_j , hay que encontrar un vector paramétrico $\mathbf{x}^* \in X$ que minimice la función objetivo.

La forma general del problema de optimización es:



$$\begin{aligned} \text{Minimiza: } & f(\mathbf{x}); \quad \mathbf{x} = \{x_i\}; \quad i = 1, \dots, n \\ \text{Con:} & \quad \quad \quad \mathbf{g}(\mathbf{x}) = \{g_j(\mathbf{x})\}; j = 1, \dots, m \\ \text{Cumpliendo:} & \quad \quad \mathbf{g}_j(\mathbf{x}) \leq 0; \quad j = 1, \dots, m \\ & \quad \quad \quad a_i \leq x_i \leq b_i; \quad i = 1, \dots, n \end{aligned}$$

donde x_i son las variables en base a las cuales se define el problema, a_i y b_i son los límites inferiores y superiores de dichas variables.

El problema de maximización puede considerarse como el problema inverso del de minimización, es decir, la minimización de $f(\mathbf{x})$ es igual a maximizar $-f(\mathbf{x})$, por lo tanto, sin pérdida de generalidad, en adelante se tratará el problema de optimización como un problema de minimización.

La secuencia básica de pasos para resolver un problema de optimización es la siguiente:

- Tomar n puntos en el espacio de búsqueda y generar m nuevos puntos en cada iteración (denominada también generación o población en algunos algoritmos de optimización). Cada punto representa una posible solución del problema. En este paso de generación de soluciones el algoritmo debe ser capaz de producir una cierta mejora local ya que en caso contrario se trataría de una mera búsqueda aleatoria.
- Evaluar el cumplimiento de las restricciones y la función objetivo a minimizar para cada una de las soluciones.
- Aceptar las soluciones evaluadas que supongan una cierta mejora y que al mismo tiempo cumplan las restricciones y puedan utilizarse como nuevos puntos de partida rechazando los demás.
- Repetir los pasos anteriores hasta que se cumpla el criterio de parada impuesto y que está relacionado con el cumplimiento de la condición de mínimo o alcanzar el número máximo de iteraciones prefijado.

Con el fin de valorar las soluciones, el usuario define una *función de evaluación*. Para el caso en que la función objetivo se conoce explícitamente, se



puede utilizar ésta como función de evaluación, de lo contrario hay que fijar una función de evaluación que refleje el problema lo mejor posible.

Algunas de las dificultades que aparecen durante la resolución de problemas de optimización son las siguientes:

- En la mayoría de problemas reales, el espacio de búsqueda es demasiado amplio como para realizar una búsqueda precisa y/o tiene que estar resuelto en un plazo determinado, por lo que sólo se puede comprobar un número reducido de soluciones posibles.
- La función objetivo a menudo es altamente no lineal y puede tener varios mínimos, de modo que hay que prestar especial atención de no quedar atrapado en un óptimo local.
- En la mayoría de problemas de optimización prácticos no se conoce la función objetivo. En estos casos se pueden determinar algunos puntos de la función objetivo mediante experimentación o simulación.
- Cuando se utiliza un modelo como aproximación del problema, los resultados pueden encontrarse afectados por el ruido. El ruido puede ser sistemático (se analiza una solución con el modelo más de una vez y se obtienen los mismos resultados) o aleatorio (se analizan dos veces la misma solución y se obtienen resultados diferentes debido a inestabilidades). La suma de errores de redondeo también origina ruido.
- Realizar una simulación o experimento puede requerir mucho tiempo, por lo que hay que intentar reducir el número de soluciones a tratar.
- Algunos puntos del espacio de búsqueda son no factibles. Esto es debido a que los puntos, o bien incumplen alguna de las restricciones objetivo (puntos no factibles) o hacen que la simulación falle (puntos no evaluables).

3.2.2 Clasificación de los algoritmos de optimización

Los algoritmos de optimización pueden clasificarse de varias formas dependiendo del criterio empleado. Se distinguen dos grandes grupos de problemas de optimización:



Optimización combinatoria

Se trata de un subconjunto de la optimización discreta, en el que X es discreta.

En la optimización combinatoria, normalmente X es finita y por lo tanto, existe un número finito de soluciones. Algunos ejemplos son *el problema del vendedor viajante* (TSP) y *el problema de satisfacción* (SAT).

Optimización continua

En los problemas de optimización continua, se distinguen los de optimización lineal y los de optimización no lineal. En la optimización lineal, f , g_j son lineales. Si alguno es no lineal, el problema pasa a ser no lineal. Algunos ejemplos de la optimización continua son la minimización o maximización de funciones, la optimización de la distribución de material o la optimización de forma.

Otra clasificación de los problemas de optimización se centra en el número de objetivos. Se pueden diferenciar varios grupos, los problemas de optimización con objetivo único y los denominados problemas de optimización multiobjetivo que requieren la optimización de varios objetivos.

La forma más popular de clasificar los problemas de optimización es en *algoritmos deterministas* y *algoritmos estocásticos*.

Los algoritmos deterministas proporcionan los mismos resultados si se inicia la búsqueda con los mismos puntos del espacio de búsqueda, en cambio, los algoritmos estocásticos incorporan un cierto grado de aleatoriedad y con los mismos puntos iniciales se obtienen resultados diferentes.

Otra posible clasificación viene determinada por la utilización o no de información relativa al gradiente. Algunos algoritmos requieren la utilización de gradientes de información de la función objetivo y otros solamente necesitan el valor de la función objetivo. *Los métodos basados en el gradiente* convergen más rápidamente que *los métodos no basados en el gradiente*, en caso de que converjan.

Otra cosa que diferencia a los algoritmos de optimización es el número de soluciones utilizadas como base para la futura exploración en cada iteración. Los *algoritmos de solución única* construyen una única mejor solución actual, la cual se



evalúa y se intenta mejorar en el siguiente paso. Por ejemplo, todos los métodos deterministas utilizan un algoritmo de solución única y requieren varias evaluaciones de una función. Por lo tanto, no es posible paralelizar estos métodos. Los *algoritmos de múltiple solución* trabajan con varias soluciones a la vez. La mayoría de algoritmos genéticos son algoritmos de solución múltiple. Estos últimos son más adecuados para la paralelización lo que permite aumentar su baja velocidad.

También es posible combinar ambos métodos (deterministas y estocásticos) en un mismo algoritmo. Estos algoritmos son los denominados algoritmos híbridos y actualmente disfrutan de una gran popularidad, ya que poseen las ventajas de los métodos deterministas (alta velocidad de convergencia) y las ventajas de los algoritmos estocásticos (amplia exploración del espacio de búsqueda). La dificultad radica en establecer una proporción óptima de búsqueda determinista y estocástica.

Otra alternativa es calcular varios casos de un algoritmo de solución única al mismo tiempo desde varios puntos iniciales diferentes, por ejemplo, con el algoritmo hill-climb. Sin embargo, esto no lo convierte en un algoritmo de solución múltiple. La clave en los algoritmos de solución múltiple es el hecho de que las soluciones interactúan. Esta interacción puede aparecer en la construcción de nuevas soluciones o en la selección de las soluciones que deben preservarse.

Hay que poner énfasis en los métodos estocásticos, puesto que este campo está lejos de ser plenamente explorado y poseen varias propiedades favorables que los convierten en una herramienta muy útil para problemas de optimización complejos.

3.2.3 Algoritmos estocásticos

Clasificación y terminología

La optimización estocástica permite minimizar/maximizar una función, incorporando cierto grado de aleatoriedad en el proceso de búsqueda. Por lo tanto, dos ejecuciones de un algoritmo estocástico, en general, no darán el mismo resultado. El uso de la aleatoriedad permite a los algoritmos escapar de los óptimos locales y realizar una búsqueda más amplia en el espacio de búsqueda respecto a los métodos deterministas. También se comportan mejor ante el ruido.



El interés en los métodos estocásticos ha aumentado notablemente en las últimas décadas. Se ha desarrollado muchos métodos y algunas ideas se han tomado prestadas, se han intercambiado y modificado a través de todos estos enfoques. Como consecuencia de ello, es difícil realizar una clasificación de los diferentes métodos sin ambigüedades en las distintas categorías.

Las categorías que a menudo se encuentran en la literatura referentes a los denominados algoritmos metaheurísticos son:

- Algoritmos Genéticos (“genetic algorithm”, GA) y evolutivos (“evolutionary algorithm”, EA)
- Cristalización simulada (“simulated annealing”, SA)
- Algoritmo de hormiguero (“ant colony algorithm”, AC)
- Búsqueda tabú (“tabu search”, TS)

Las definiciones varían significativamente a lo largo de la literatura. Por ejemplo, en [1] el Algoritmo Evolutivo es un término “paraguas” empleado para describir la resolución de problemas basados en algoritmos que utilizan modelos computacionales de algunos de los mecanismos conocidos de evolución como elementos clave en su diseño e implementación. Esta definición no es muy útil, ya que se aplica a todos los métodos evolutivos y genéticos, si se consideran todos “los mecanismos de evolución”, siendo estos:

- recombinación
- mutación
- selección

Los métodos estocásticos se pueden agrupar en dos grandes grupos: Algoritmos Genéticos y Algoritmos No Genéticos. Los algoritmos genéticos modelan la evolución a nivel de los individuos de una especie. Los individuos se recombinan entre ellos, generando descendencia. La descendencia también puede verse sometida a la mutación. Los algoritmos no genéticos modelan el cambio en base a otros conceptos diferentes según el propio algoritmo.

En adelante se presentarán los algoritmos genéticos.



Esquema básico de los algoritmos genéticos

La mayoría de métodos estocásticos son algoritmos de solución múltiple. Ellos trabajan con una población de soluciones, llamadas individuos, que se esfuerzan por sobrevivir y reproducirse. La unidad básica de evolución es el individuo y la unidad básica que caracteriza el individuo es el gen. El tiempo se divide en pasos discretos, llamados generaciones. En cada generación se generan nuevos individuos. Cada individuo se evalúa en el medio ambiente, es decir, en el entorno, mediante la asignación de una medida de su salud o fortaleza física (*fitness*) en el entorno. El objetivo de todo algoritmo es encontrar la máxima fortaleza. El esquema básico del algoritmo genético es el que se muestra en la siguiente figura:

Inicio

Creación de la población inicial $P(t)$

Evaluación de $P(t)$

Repetir

P' : =selección de padres $P(t)$

Recombinación de $P'(t)$

Mutación de $P'(t)$

Evaluación de $P'(t)$

P : = selección($P(t)$, $P'(t)$)

Hasta cumplir la condición de parada

Fin

Los algoritmos no genéticos no incluyen la etapa de recombinación de $P'(t)$. Los distintos algoritmos genéticos se diferencian en la implantación de los operadores y parámetros que se explican a continuación.



Operadores y parámetros

Los algoritmos genéticos se rigen por el siguiente conjunto de parámetros:

- Representación
- Función de evaluación
- Población (tamaño, tipología, inicialización)
- Criterio de parada
- Tratamiento de la restricción

Además, todos los algoritmos emplean un conjunto de operadores. Estos son:

- Generación de descendencia
- Mutación
- Permutación/inversión
- Recombinación
- Selección
- Selección de apareamiento
- Selección natural

Se sabe que no existe un conjunto de parámetros óptimo para todos los problemas. Los parámetros se habrán de ajustar a cada problema concreto ya que, los valores de los parámetros ajustados para un problema específico no funcionarán bien en general para otros problemas.

Todas las elecciones en torno al diseño de un algoritmo genético han de ser realizadas teniendo en cuenta otras muchas cuestiones, por ejemplo, que el diseño del operador mutación sólo puede realizarse considerando la representación del individuo, etc.



Representación

Los algoritmos genéticos pueden incorporar alguna representación que sea apropiada para el problema. Las representaciones más comunes son:

- Representación binaria
- Representación continua
- Permutaciones de variables
- Expresiones simbólicas en forma de árboles de análisis

Función de evaluación o función objetivo

La función de evaluación juzga la calidad de las soluciones generadas. La búsqueda puede ser efectiva sólo si la función de evaluación y los operadores son elegidos para trabajar conjuntamente. Un pequeño cambio en la solución también debería dar lugar a un pequeño cambio en su evaluación. De otro modo, el algoritmo no funcionará correctamente. El criterio más importante en el diseño de la función de evaluación es que la solución óptima debe también recibir la óptima evaluación, de lo contrario, no existe ninguna posibilidad de encontrar siempre una solución óptima. Por lo tanto, también se deben dar altas evaluaciones para soluciones cercanas al óptimo. La forma de la función de evaluación depende de su representación. Por tanto, la representación hay que tenerla presente cuando se realiza la elección de la función de evaluación.

Población

Tamaño de la población

Cuanto mayor es la población, mayor es la probabilidad de encontrar el óptimo global, ya que se realiza una mayor exploración en el espacio de búsqueda. Sin embargo, esta ventaja viene acompañada de un mayor número de evaluaciones de la función objetivo, de forma que el coste computacional del proceso de optimización aumenta.



Población inicial

Si a priori se conoce alguna información sobre el problema, debe incorporarse en la población inicial, por ejemplo, uno o más individuos de la población inicial pueden ser óptimos locales obtenidos mediante un algoritmo de búsqueda local. Para evitar agrupar la población inicial en una zona del espacio de búsqueda, se pueden forzar a los individuos iniciales a situarse a una distancia mínima entre ellos (dependiendo de la representación, la distancia debería ser medida como distancia Euclídea, distancia de Hamming, etc.). Existen muchos algoritmos para tratar de generar una población inicial con suficiente diversidad (diseño de experimentos, ...).

Criterio de parada

El criterio de parada influye significativamente en el tiempo de funcionamiento del algoritmo. El tiempo disponible para resolver el problema de optimización depende de numerosos factores, tales como los recursos computacionales disponibles, la urgencia de encontrar una solución (a veces es suficiente encontrar una solución mejor que la de partida sin necesidad de que sea la solución óptima) o simplemente el tiempo disponible para terminar el proyecto.

En función de estos prerequisites, se pueden elegir diferentes criterios de parada:

- Encontrar una solución factible tan cerca del óptimo como sea posible.
- Encontrar un número de soluciones factibles.
- Ejecutar el algoritmo hasta alcanzar un número predefinido de pasos

Consideración de soluciones no factibles

En la mayoría de los problemas se introducen restricciones, dividiendo el espacio de búsqueda en las regiones factibles y las no factibles. Uno de los desafíos en la optimización con restricciones reside en el tratamiento de las soluciones no factibles. A menos que se tenga alguna información previa sobre la localización del óptimo global, se debe asumir la existencia de una región factible encerrada por una región no factible. Por lo tanto, si se descartan las soluciones no factibles se



puede producir una importante pérdida de información. El principal problema radica en cómo evaluar las soluciones no factibles. Dependiendo del problema, estas soluciones se pueden abandonar completamente, “repararlas” o dejarlas continuar para que participen en el proceso de evolución, asignándoles un *fitness* ficticio que valore el grado de las restricciones.

En la literatura se sugieren varios enfoques:

- Introducir una función de penalización: Cuanto más se introduce una solución en la región no factible, mayor es la penalización. La penalización se obtiene a partir del *fitness* de la solución y del nivel de incumplimiento de las restricciones.
- Métodos de reparación: Se modifican las soluciones no factibles y se introducen en la zona factible más cercana del espacio de búsqueda.
- Métodos barrera: A las soluciones que caen en la región no factible se les asigna un *fitness* $\pm \infty$.
- Modificar los valores de la función en las regiones factibles cerca de regiones no factibles, para prevenir soluciones que deriven hacia regiones no factibles.
- Modificar la función objetivo sobre todo el dominio.
- Convertir cada una de las restricciones en un objetivo separado, que tiene que ser optimizado además de los objetivos reales, de este modo el problema de optimización de objetivo único se convierte en un problema de optimización multiobjetivo.
- Restricción de violación general: Añadir restricciones para obtener un criterio adicional de optimización y de esta forma obtener otra vez un problema multiobjetivo.
- Favorecer más las soluciones factibles que las no factibles cuando se lleva a cabo la selección.



Generación de la descendencia

La descendencia se debe generar de modo que se mantenga el vínculo entre padres e hijos. La descendencia debe asemejarse a los padres, de lo contrario se realiza una búsqueda puramente aleatoria.

Se distinguen varias transformaciones en la descendencia, las unarias y las de mayor orden. Las transformaciones unarias cambian un único individuo para generar uno nuevo (por ejemplo, la mutación y la inversión). Las transformaciones de mayor orden combinan partes de dos o más individuos (por ejemplo, la recombinación).

Mutación

La mutación se logra modificando el valor de algunos o de todos los parámetros de una solución. Cada parámetro x_i sujeto a la mutación se convierte en x_i' :

$$x_i' = x_i + \Delta x_i \quad (3.1)$$

Dependiendo del tipo de mutación, Δx_i se calcula de diferente modo.

Los procedimientos de mutación más comunes son:

Mutación Gaussiana

La distribución Gaussiana tiene la forma:

$$P(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (3.2)$$

Donde σ es la derivación típica (desviación estándar) y μ es la media. En la mutación Gaussiana, Δx_i es un valor obtenido de la distribución Gaussiana con media 0 y desviación típica σ que viene establecida por el tamaño de paso de la mutación. Por lo tanto, $\Delta x_i = N(0, \sigma)$.

Mutación de Cauchy

La distribución de Cauchy tiene la forma:

$$P(x) = \frac{1}{\pi} \frac{\frac{1}{2}\Gamma}{(x - \mu)^2 + (\frac{1}{2}\Gamma)^2} \quad (3.3)$$

Donde Γ es el ancho total a la mitad del máximo y μ es la media. Cuando se utiliza la mutación de Cauchy, Δx_i se obtiene de una distribución aleatoria de Cauchy. Para la mayoría de opciones de Γ , es más probable que se produzca una mutación severa utilizando la mutación de Cauchy que con la mutación Gaussiana, por lo tanto la mutación de Cauchy crea una descendencia más distinta a los padres.

Mutación Uniforme

Δx_i es una variable aleatoria uniforme. Dentro del rango especificado, todos los valores poseen la misma probabilidad, por lo tanto existe la misma posibilidad de que se den mutaciones tanto severas como moderadas.

Mutaciones correlacionadas

Los tamaños de paso se representan en una matriz C de tamaño $n \times n$, siendo n el tamaño de la solución. Los coeficientes de la diagonal de C son las varianzas de la mutación, los coeficientes fuera de la diagonal son las covarianzas. C se utiliza para transformar linealmente un vector aleatorio normalmente distribuido $z \sim N(0, I)$. El vector mutación Δx se obtiene como $\Delta x = \sigma \sqrt{C} z$, donde σ es el tamaño de paso global. Con esta mutación se correlaciona el tamaño de paso de la mutación para el *fitness* de la solución y las soluciones con mayor éxito son sometidas a una menor mutación.

Referente a todos los métodos de mutación hay que señalar que pueden mutar todos los parámetros o se puede dar a cada parámetro una probabilidad de mutación. Esto implica que estos parámetros solo se mutarán con una cierta probabilidad. Se puede elegir aleatoriamente un número de individuos para ser



mutados/recombinados o variar aleatoriamente el número de parámetros de cada individuo que mutan. Otra alternativa es fijar el porcentaje de parámetros a ser mutados, regulando así la semejanza de la descendencia con sus padres. Se puede introducir un tamaño de paso de la mutación global que se aplique a todos los parámetros o dar a cada parámetro su propio tamaño de paso. La última posibilidad probablemente será el método escogido para problemas de optimización en donde los parámetros de individuo tienen diferentes rangos de valor. El procedimiento de mutación debe ser diseñado para poder generar cualquier posible solución en el espacio de búsqueda y así tener la posibilidad de explorar todo el espacio de búsqueda. Esto depende del método de selección y del criterio de parada, pero es muy importante que el procedimiento de mutación sea capaz de dar la oportunidad de generar cualquier solución.

Permutación/Inversión

Estos operadores se utilizan en la optimización combinatoria. Implican la permutación de una parte de la solución. La inversión es una forma específica de permutación que invierte el orden de algunos o de todos los parámetros en un vector solución. Por ejemplo, cuando se resuelve el *TSP* (Problema del vendedor viajero), se puede permutar el orden de visita de algunas o de todas las ciudades mediante la permutación del vector solución. Se puede elegir cambiar el orden en uno o varios segmentos de la solución. Se puede además asignar para cada punto una probabilidad de ser un punto inicial o final de un segmento, determinando estocásticamente el grado de permutación.

Recombinación

Los métodos básicos de recombinación son los *métodos de cruce* y los *métodos de medias ponderadas*.

El *cruce* denota el proceso de creación de una descendencia mediante la combinación de algunas partes de dos o más soluciones padre. Estas partes pueden variar de longitud (varía el número de genes). En el caso extremo, la descendencia recibe un gen de cada padre. Se distingue entre un cruce de un punto y un cruce de n puntos dependiendo del número de los puntos de cruce.



Los *métodos de media ponderada* combinan varias soluciones padre (componente por componente) para crear una descendencia. Los diferentes individuos padre pueden recibir diferentes pesos.

Las soluciones también pueden ser agrupadas dentro de *clusters* dependiendo de su posición en el espacio de búsqueda y de esta forma restringir el proceso de recombinación dentro del *cluster*. Esto evita soluciones muy diferentes dentro de *clusters* exitosos. En este caso hay que especificar un número mínimo de miembros en un *cluster*.

Selección

Hay dos fases de un algoritmo en las que la selección tiene lugar. En la primera, se elige aquellos individuos de la población que van a generar la descendencia. Esto se llama *selección de apareamiento*. Los individuos escogidos se copian en la *charca de apareamiento*. El segundo proceso de selección determina qué nuevos individuos creados y qué individuos de generaciones previas van a formar parte de la nueva generación. Esto se denomina *selección natural*.

Hay infinitas posibilidades de cómo realizar la selección. En general, se puede distinguir entre selección determinista y estocástica. Dada la misma población de padres e hijos, un método de selección determinista siempre elimina los mismos individuos, mientras que el número de individuos eliminados varía en un método de selección estocástica. La selección determinista normalmente converge más rápidamente, aunque aquí se puede manifestar la no deseada convergencia prematura.

Elegir un método de selección es un compromiso entre intentar mejorar la población y mantener una diversidad entre individuos de la población de manera que sea amplíe la exploración en el espacio de búsqueda.

Los diferentes métodos utilizados en la selección natural son:

- Utilizar la *charca de apareamiento* modificada como nueva población
- Combinar la *charca de apareamiento* modificada y la población antigua y aplicar un método de selección
- Elitismo: preservar siempre los mejores individuos a lo largo de las generaciones, convirtiendo las mejores soluciones en inmortales.



- Procedimiento de prueba binario modificado. Se elige aleatoriamente un padre y un hijo y se acepta el que posea un mejor *fitness*.

Existe otro conjunto de métodos que se emplean tanto para la selección de apareamiento como para la selección natural, estos son:

- Ruleta de selección (también llamada selección proporcional)
- Elección determinista de los mejores individuos
- Elección de los i mejores individuos y de otros j individuos aleatoriamente de forma que $i + j = p$, siendo p el tamaño de la charca de apareamiento o de la nueva generación.
- Procedimiento del torneo binario. Elección aleatoria de dos individuos fuera de la población y copia del individuo con mejor *fitness* dentro de la *charca de apareamiento* o la nueva generación hasta alcanzar el número de individuos requerido, de lo contrario se repite el proceso.
- Torneos aleatorios: El mejor individuo de una muestra aleatoria sobrevive. Este proceso se repite hasta que se llega al número de individuos especificado. Dependiendo del tamaño de la muestra, este método varía entre el torneo binario (tamaño de muestra 2) y el método determinista de selección de los mejores individuos (tamaño de muestra = población antigua + descendencia).
- Algoritmo Metrópolis: Siguiendo el ejemplo de la distribución probabilística de Boltzman de energía, la probabilidad de que una solución 1 reemplace a la 2 es:

$$P(E) = e^{\frac{-(E_2 - E_1)}{kT}} \quad (3.4)$$

donde E_1 y E_2 representan la energía (correspondiente al *fitness* negativo) de las dos soluciones, respectivamente. k hace referencia a la constante de Boltzman y T es la actual temperatura. Si $E_2 < E_1$, P es mayor que la unidad. En este caso se fija a la unidad, es decir, se acepta la solución 2. Si la nueva solución es peor que su antecesora ($E_2 > E_1$), ésta se acepta con una cierta probabilidad P . Por lo tanto, este método de selección a veces aceptará una peor solución.



3.3 OPTIMIZACIÓN DE FORMA

Esta tesina se centrará en una parte del campo matemático de diseño de optimización. Es decir, se tratarán estructuras mecánicas cuya principal tarea sea soportar cargas. Este campo recibe el nombre de optimización estructural.

La siguiente función y variables se hallan siempre presentes en los problemas de optimización estructural:

- **Función Objetivo (FO):** Es una función utilizada para clasificar los diseños. Para cada posible diseño, FO devuelve un número que indica la bondad de diseño. Normalmente se escoge una FO para la cual un valor más pequeño sea mejor que uno más grande (problema de minimización). Frecuentemente la FO mide pesos, desplazamientos en una dirección dada, tensiones efectivas o incluso costes de producción.
- **Variable de Diseño (x):** Una función o vector que describe el diseño, la cual puede ser modificada durante la optimización. Puede representar la geometría o la elección de material. Cuando describe la geometría, puede relacionarse con una sofisticada interpolación de forma o puede ser simplemente el área de una barra, o el espesor de una hoja.
- **Variable de Estado (y):** Para una estructura dada, p.e., para un diseño dado x , y es una función o vector que representa la respuesta de la estructura. Para una estructura mecánica, la respuesta significa desplazamiento, tensión, deformación o fuerza.

Un problema de optimización estructural en general tomará la siguiente forma:

- Ha de minimizar $f(x,y)$ con respecto a x e y
- Ha de estar sujeta a: restricciones de comportamiento en y , restricciones de diseño en x y a una restricción de equilibrio.



Este trabajo se centrará exclusivamente en la definición geométrica de la estructura pues, dependiendo de la geometría, se puede dividir la optimización estructural en tres tipos diferentes:

- **Optimización de tamaño:** En este caso x representará algún tipo de espesor de la estructura.
- **Optimización de forma:** En este caso x representa la forma del contorno de alguna parte del límite del dominio estructural. La optimización en general consiste en escoger el dominio de integración de las ecuaciones diferenciales que describen el estado de un sólido de forma óptima.
- **Optimización topológica:** Es la forma más general de optimización estructural. En el caso discreto, se logra tomando secciones transversales de zonas de la estructura, por ejemplo una celosía, como variables de diseño, y entonces se permite a dichas variables tomar valor cero, esas zonas serán eliminadas de la estructura. En este sentido la conectividad de los nodos es variable por lo que se puede decir que la topología de la estructura cambia. Si en cambio, se piensa en una estructura de tipo continuo entonces la topología cambia debido a que se permite al espesor de la pieza tomar el valor cero.

Idealmente, la optimización de forma es una subclase de la optimización topológica, pero prácticamente sus implementaciones están basadas en técnicas muy diferentes, por ello se suelen tratar separadamente. En lo concerniente a la relación entre la optimización topológica y la optimización de tamaño, la situación es justo la contraria: desde un punto de vista fundamental son muy diferentes, pero están muy relacionadas desde un punto de vista práctico.

Cuando el problema es una ecuación diferencial, se puede decir que la optimización de forma controla el dominio de la ecuación, mientras que las optimizaciones de tamaño y forma controlan sus parámetros.

A continuación se detallará la importancia de la optimización de forma que será la que se desarrollará en el presente trabajo.

El uso de técnicas de optimización de forma de componentes industriales es importante puesto que conduce a la obtención de piezas que satisfacen mejor alguno de los requisitos fundamentales del producto, normalmente la capacidad resistente del elemento. El uso de estas metodologías conduce a piezas de menor



peso y de mejores prestaciones. Desde el punto de vista industrial, la disminución del material empleado en una pieza, suele llevar asociada una reducción de costes que es muy importante en sectores donde se fabrican grandes series, tales como el sector del automóvil, por ejemplo. En todos los sectores relacionados con el transporte de personas o mercancías la forma óptima o disminución de peso del vehículo siempre conllevan un cierto ahorro energético. Hay que señalar que en algunos sectores concretos como el aeronáutico, el aeroespacial, el automovilístico y el naval, el requisito de forma óptima y peso mínimo es muy importante y, en algunos casos, absolutamente imprescindible.

Los algoritmos evolutivos (AE) son técnicas emergentes en la resolución de problemas de optimización. Estos métodos son muy atractivos ya que, además de ser robustos, resultan muy eficientes para conseguir mínimos globales, incluso con espacios de diseños altamente no-lineales, discontinuos y multi-dominio, ya que no quedan atrapados en mínimos locales al incorporar cierto grado de aleatoriedad en la creación de las múltiples soluciones a analizar.

Es bien conocido que los AE suelen ser capaces de proporcionar la solución incluso en aquellos problemas en los que los procedimientos de optimización tradicionales tienen grandes dificultades, tales como en el caso de problemas con múltiples restricciones, o con datos de naturaleza estocástica. Sin embargo, las características favorables de este tipo de algoritmos se consiguen a costa de realizar un número elevado de evaluaciones de la función objetivo (FO). Esto puede convertirse en un problema muy importante si la evaluación de la FO implica un alto coste computacional. Este es el caso de los procesos de optimización de forma de componentes en sectores industriales tales como el aeronáutico (formas de alas y fuselaje de aviones), el naval (cascos de barcos), el del automóvil, y otros, donde la evaluación de FO (la resistencia al aire o el impacto sonoro, típicas en los sectores aeronáutico y del automóvil), o la determinación del cumplimiento o no de las restricciones del problema, expresadas en función de desplazamientos o tensiones máximas admisibles, etc., requieren la utilización de técnicas numéricas de cálculo como el *método de los elementos finitos* (MEF), *método de los elementos de contorno* (MEC), *métodos sin malla*, etc., que puedan proporcionar la solución con un nivel de precisión tal que garantice la convergencia adecuada del proceso de optimización.

Los procesos de optimización basados en AE están constituidos por un nivel superior gobernado por el algoritmo de optimización y un nivel inferior que está gobernado por el método numérico utilizado para analizar cada una de las múltiples soluciones. El método numérico utilizado en este trabajo ha sido el MEF.



En los últimos años se han desarrollado diferentes técnicas con el objetivo de superar las dificultades asociadas a los algoritmos evolutivos de optimización y mejorar así el rendimiento computacional del nivel superior.

En lo que se refiere al nivel inferior se han de tener en cuenta dos aspectos fundamentales: velocidad de procesamiento de cada uno de los individuos analizados y precisión de los resultados obtenidos. Resulta evidente la importancia de la velocidad de cálculo de cada individuo sobre el proceso de optimización ya que, en problemas de optimización como los tratados, la mayor parte del coste computacional viene precisamente del análisis de los individuos mediante las técnicas numéricas anteriormente enumeradas (MEF, MEC,...) a fin de obtener los valores de la FO y del grado de cumplimiento de las restricciones del problema. Por otro lado, en lo que a precisión de resultados se refiere, el cálculo de estos valores se realiza solamente de manera aproximada por lo que, si estos valores no son de suficiente precisión, se puede introducir ruido excesivo en el proceso de optimización haciendo que éste no converja hacia el diseño óptimo a la velocidad adecuada, lo haga a solución no óptima, o, simplemente, diverja. En el contexto del MEF las llamadas técnicas *h-adaptativas*, *p-adaptativas*, etc., permiten conseguir una solución con un nivel de error inferior al especificado por el analista en función de la precisión requerida por el proceso de optimización mediante el análisis sucesivo de modelos numéricos progresivamente más precisos, automáticamente adaptados a la solución del problema. Evidentemente estas técnicas implican un considerable esfuerzo computacional que afecta al rendimiento del proceso de optimización.

3.4 MOGA- II: DESCRIPCIÓN Y PARÁMETROS DE CONTROL

Se va a utilizar Moga-II para desarrollar esta tesina puesto que, un análisis previo del problema de referencia de la viga, con diferentes algoritmos de optimización de los que utiliza el programa modeFRONTIER, permitió determinar que era el que mejor se adaptaba al problema de optimización de forma y además se conseguía una mayor velocidad de convergencia.



3.4.1 Descripción de algoritmo

El programa modeFRONTIER utiliza una versión propia del Algoritmo Genético Multi-Objetivo, denominada Moga-II.

Se trata de una versión eficiente que utiliza un elitismo inteligente multi-búsqueda. Este nuevo operador elitismo es capaz de preservar algunas de las mejores soluciones, sin quedarse atrapado en una convergencia prematura en fronteras de óptimos locales.

La utilización del algoritmo Moga-II requiere sólo unos pocos parámetros proporcionados por el usuario y junto con otros parámetros internos definidos en el algoritmo proporcionan robustez y eficiencia al optimizador. El algoritmo realiza un número total de evaluaciones que es igual al número de puntos existentes en la tabla de diseños experimentales (DOE), también denominado población inicial, multiplicado por el número de generaciones. El tamaño de cada ejecución generalmente queda definido por los recursos informáticos disponibles.

Nota: Moga-II sólo funciona con variables discretas. Aparece un error si una base variable de entrada es igual a 0

Dentro de Moga-II existen tres variaciones de algoritmos:

- **Moga – Evolución Generacional** trabaja en un conjunto de configuraciones de diseño que se actualizan periódicamente, cuando una generación se ha completado.
- **Moga – Evolución Fija** utiliza todas las configuraciones calculado tan pronto como estén disponibles en un modo "first in - first out".
- **Moga – Evolución Adaptativa** la elección de los operadores se realiza automáticamente por el método durante la búsqueda. La probabilidad de cruzado direccional y la probabilidad de mutación se toman como condiciones iniciales. Durante la búsqueda el algoritmo de adaptación cambia su ratio de acuerdo a los resultados de evaluación. La evolución adaptativa se puede utilizar cuando no hay una idea clara acerca de las probabilidades de los operadores.



También cabe destacar que aunque se trata de un algoritmo multi-objetivo en la presente tesis se ha utilizado como mono-objetivo.

3.4.2 Parámetros

Los parámetros que debe especificar el usuario para la utilización de MOGA-II son:

- **Número de generaciones:** Este valor define el tamaño máximo de la simulación.
- **Probabilidad de cruzado direccional:** El operador de cruce es un método de recombinación que los padres producen hijos mediante intercambio de información. El objetivo de este operador es la obtención de individuos con mejores características, mientras se mantiene la diversidad de la población. El cruzado se considera como el principal operador de búsqueda en los algoritmos genéticos. Este cruzado específico es un operador propio que da a la eficiencia al algoritmo. Si este parámetro se establece en 1, sólo se utiliza este operador. La búsqueda es eficiente, pero en caso de problemas altamente no lineales el optimizador puede quedarse atascado en las soluciones locales óptimas. Los valores 0 y 1 no son recomendables. El valor predeterminado es de 0.5, aunque este valor debe ser aumentado en caso de problemas relativamente suaves y disminuido en el resto de casos.
- **Probabilidad de Selección:** Este valor indica la probabilidad de que las configuraciones de diseño no cambien durante la evolución. Con el fin de mantener una buena diversidad entre los puntos, este parámetro debe ser pequeño. Si el valor es 1 no se llevará a cabo ningún cálculo después de la población inicial. El valor por defecto es de 0,05 y su valor no se debe aumentar a valores que queden por encima de 0,2. Si la probabilidad de cruzado direccional más la probabilidad de selección es superior al valor de 1,0 se aplica la siguiente regla:

Probabilidad de Selección = 1 - (Probabilidad de Cruzado Direccional)



- **Probabilidad de mutación:** Este valor indica la probabilidad de que una configuración de diseño sea cambiada aleatoriamente. Si el valor es 1, el algoritmo se convierte en una búsqueda puramente aleatoria. Si la probabilidad de cruzado direccional más la probabilidad de selección además de la probabilidad de mutación es superior al valor de 1,0 se aplicará lo siguiente:

$$\text{Probabilidad de Mutación} = 1 - (\text{Probabilidad de Cruzado Direccional} + \text{Probabilidad de Selección})$$

La probabilidad del cruzado clásico se ajustará automáticamente a:

$$\text{Probabilidad de Cruzado Clásico} = 1 - (\text{Probabilidad de Cruzado Direccional} + \text{Probabilidad de selección} + \text{Probabilidad de mutación})$$

Si la probabilidad del cruzado clásico es alta, el algoritmo saltará más fácilmente a una región diferente del espacio de diseño, pero también será más lento en la convergencia.

- **Ratio de mutación de la cadena de ADN:** Este valor indica el porcentaje del ADN individual que es perturbado por el operador de mutación. El ADN de cada individuo es codificado en una cadena binaria, el ratio de mutación define el número de bits que mutan. Si la relación de mutación es 0.0 ninguno de los bits de ADN son modificados. Un valor de 1.0 cambiará todos los bits de ADN.

Ejemplo:

ADN longitud = 10

Ratio de mutación de la cadena de ADN = 0,5 (50% de la cadena de ADN se mutado)

ADN = [1001001101]

ADN nuevo = [1101110100]



x xxx x (x = bit cambiado)
Cinco bits de 10 elegidos aleatoriamente han cambiado.

- **Elitismo:** El elitismo puede ser activado o desactivado. Esta opción (si está activado) se asegurará de que las mejores soluciones (para cada objetivo) se conservan durante la evolución.

3.4.3 Tratamiento de las restricciones

Las restricciones pueden ser tratadas de dos maneras:

- Asegurando que una solución factible es siempre mejor que una solución no viable y teniendo en cuenta la intensidad de la violación de restricción. Esto se hace usando "fuzzy" limitaciones para las que se especifica una tolerancia de violación a nivel de definición del proyecto.
- Mediante la adición de un objetivo para el problema igual al número de restricciones violadas. Esta opción sólo tiene sentido para problemas sobre-restringidos en los que una solución factible no existe y hay una serie de limitaciones de importancia relativamente equivalente.

3.5 PROBLEMAS DE REFERENCIA

3.5.1 Justificación

Se pretende comprobar la robustez y precisión de algoritmo Moga-II, utilizado por modeFRONTIER. Y para ello, puesto que la optimización es en realidad un procedimiento estocástico, se requiere la realización de conjuntos de ejecuciones con unos cuantos ejemplos a fin de obtener unos resultados que sean lo más generales posible.

En primer lugar, se resolverá un problema sencillo de una viga en 1D de forma analítica. Y una vez comprobado el correcto funcionamiento de algoritmo se analizará un problema en 2D mediante el MEF, utilizando un programa

perteneciente al Departamento de Ingeniería Mecánica e implementado en Matlab, que utiliza mallados independientes de la geometría (Programa de GFEM).

Por último, se comprobarán los resultados obtenidos para los dos problemas de referencia anteriores con otro un poco más complejo de un puente sometido a un carga móvil, que también se calculará mediante el MEF.

3.5.2 Viga

En primer lugar, se ha resuelto un problema unidimensional sencillo consistente en una viga en voladizo empotrada en un extremo. En el extremo en voladizo de la viga se coloca una carga puntual P y se obtendrá la forma óptima de la viga teniendo en cuenta que las tensiones en la viga han de ser constantes.

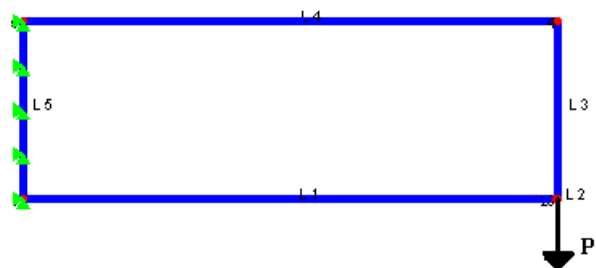


Fig. 1 Viga en voladizo con cargas aplicadas

En teoría, las tensiones normales en el extremo en voladizo de la viga serán nulas. No obstante, en dicho extremo ha de haber una sección mínima de tal forma que resista las tensiones de cortadura a las que se verá sometida la viga debidas a la carga puntual P' . Para ello, se ha considerado un momento M adicional que será constante e igual al producto de P (cuyo valor es igual al de P') por L_0 , con $L_0=0,1\text{m}$. Este momento suplirá de algún modo el efecto de las tensiones tangenciales en el extremo de la viga, ya que de esta manera se considera una tensión normal equivalente ficticia que hace que en el extremo de la viga haya una sección suficiente para resistir la cortadura.

En la siguiente figura se muestran de forma esquemática las cargas aplicadas a la viga. En la imagen se pueden ver en azul el momento ficticio y en rojo la carga puntual real aplicada.

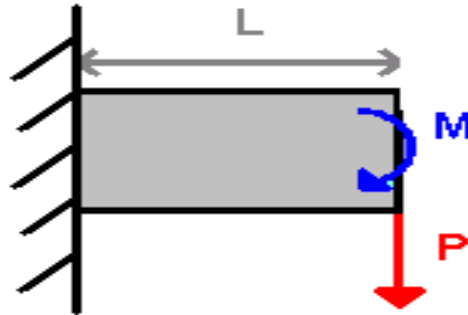


Fig. 2 Problema equivalente teniendo en cuenta las tensiones tangenciales en el extremo en voladizo de la viga

Las propiedades mecánicas del material y a la carga aplicada se detallan en la siguiente tabla:

| Carga Puntual P | Máxima tensión de von Mises admisible |
|-----------------|---------------------------------------|
| 500 N | $2,5 \times 10^7$ Pa |

Tabla 1 Máxima tensión admisible y carga aplicada

En el modelo empleado, se ha utilizado un plano de simetría, de forma que es suficiente con modelar la mitad de la viga. En este caso se ha definido únicamente la mitad superior de la viga, en la cual se considera que, tanto la longitud de la viga como su anchura son constantes y únicamente es variable su altura.



| Longitud | Anchura |
|----------|---------|
| 1 m | 0,3 m |

Tabla 2 Dimensiones de la viga

La descripción de cómo se ha parametrizado la geometría para su posterior optimización se especificará en el apartado 4 de este trabajo.

3.5.3 Cilindro

El segundo problema de referencia consiste en un cilindro sometido a presión interna.

En el modelo de elementos finitos se han empleado dos planos perpendiculares de simetría, de este modo, sólo es necesario modelar una cuarta parte de la sección. El problema de optimización consiste en encontrar la mejor forma manteniendo el contorno interno fijo y permitiendo variar el contorno externo.

La forma óptima de un conducto es aquella cuyos contornos interno y externo son arcos de circunferencia. Para este problema se conoce su solución analítica, es decir, el espesor del conducto que garantiza que no se produzca el fallo para una presión interna fijada. El conducto se ha modelado en un estado en deformación plana, es decir, se ha analizado una sección lejos de los extremos de un conducto largo de forma que no aparezcan efectos del borde del conducto relacionados a un estado de tensión plana.

Para evitar una posible influencia del error de discretización sobre la solución obtenida, se ha fijado un error relativo en norma energética del 1%, es decir, un nivel de error bajo. La posible influencia del error sobre el proceso de optimización se determina en un estudio posterior.

El nivel de tensiones en un conducto sometido a una presión interna respecto a un sistema de coordenadas cilíndricas se obtiene con las siguientes expresiones:

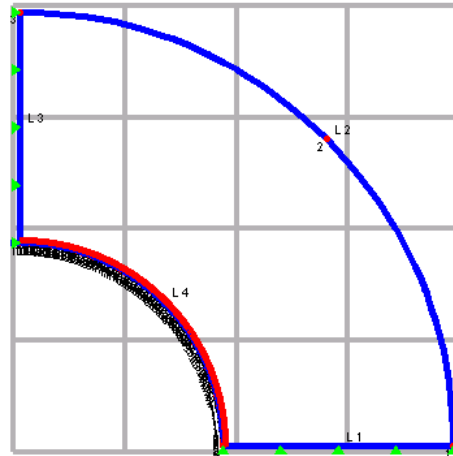


Fig. 3 Cilindro sometido a presión interna

$$\begin{aligned}
 k &= \frac{b}{a} \\
 \sigma_r &= \frac{P}{k^2 - 1} \left(1 - \frac{b^2}{r^2} \right) \\
 \sigma_t &= \frac{P}{k^2 - 1} \left(1 + \frac{b^2}{r^2} \right)
 \end{aligned}
 \tag{3.5}$$

k = Relación entre el radio externo (b) y el radio interno (a)

σ_r = Tensión en la dirección radial

σ_t = Tensión en la dirección tangencial

El nivel de tensiones con respecto a un sistema de coordenadas cartesianas con mismo origen se puede calcular como:

$$\begin{aligned}
 \sigma_{xx} &= \sigma_r \cos^2(\phi) + \sigma_t \sin^2(\phi) \\
 \sigma_{yy} &= \sigma_r \sin^2(\phi) + \sigma_t \cos^2(\phi) \\
 \tau_{xy} &= (\sigma_r - \sigma_t) \sin(\phi) \cos(\phi)
 \end{aligned}
 \tag{3.6}$$



Se observa que el mayor nivel de tensiones aparece en la zona interna del conducto, es decir, para un radio igual al interno e independiente del ángulo formado respecto la horizontal.

Las propiedades mecánicas utilizadas no pertenecen a ningún material en concreto. Tampoco se han especificado unidades en las diferentes propiedades mecánicas. El problema se ha resuelto adimensionalmente.

| Módulo de Young | Coefficiente de Poisson | Máxima tensión de von Mises admisible |
|------------------------|--------------------------------|--|
| $10,1 \times 10^6$ | 0,3 | $2,0 \times 10^6$ |

Tabla 3 Propiedades mecánicas del material del conducto

Los otros datos pertenecientes al problema son la presión interna aplicada y el radio interno:

| Presión interna | Radio Interno |
|------------------------|----------------------|
| $0,9 \times 10^6$ | 0,5 |

Tabla 4 Presión y radio interno del conducto

Al igual que en el problema anterior los detalles correspondientes a la parametrización de la geometría se abordarán en el apartado cuatro de esta tesis.



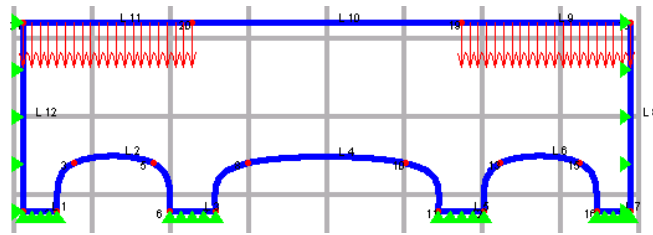
3.5.4 Puente

El ejemplo consiste en el cálculo de un puente con una carga móvil. El cuerpo del puente, cuyas dimensiones son 16m x 5m x 0,15m está soportado por cuatro columnas sólidas.

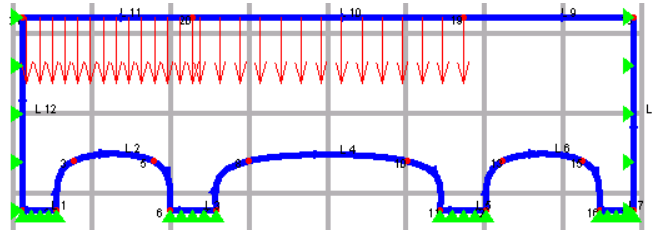
El puente se va a someter a 7 casos de carga distintos, pero debido a la simetría que presenta el problema únicamente será necesario calcular los 4 que se muestran en la Fig. 23. La carga a la que se verá sometido en todos los casos es una carga distribuida $P=1000\text{kN/m}^2$.

No obstante, se ha resuelto únicamente cinco de los 9 casos, debido a la simetría geométrica existente.

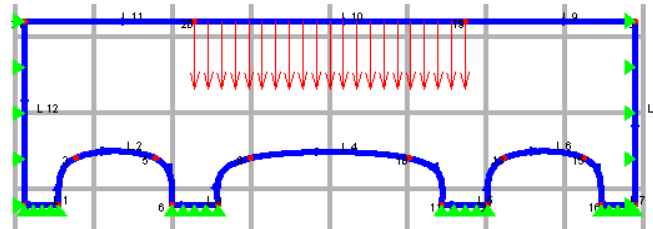
Se asume un módulo de Young $E=30\text{GPa}$ y un coeficiente de Poisson $\nu=0,2$, siendo la tensión máxima admisible de 10MPa y el desplazamiento máximo admisible de 1,5mm.



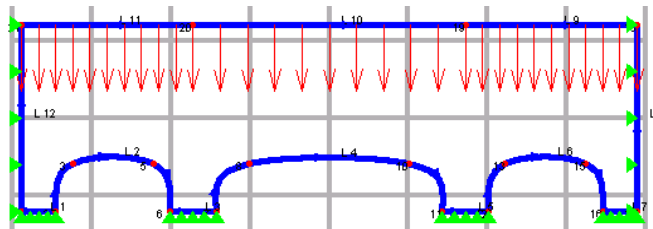
Caso de carga 1



Caso de carga 2



Caso de carga 3



Caso de carga 4

Fig. 4 Puente sometido diferentes casos de carga



Se considerará que la solución óptima de la geometría del puente es aquella formada por el contorno envolvente obtenido a partir de la solución óptima hallada para cada uno de los casos de carga.



4 PARÁMETROS DE CONTROL

4.1 INTRODUCCIÓN

El teorema *No Free Lunch* afirma que no existen ningún algoritmo ni ningunos parámetros que sean los más adecuados para todos los problemas de forma genérica. No obstante, una vez fijado el problema sí se pueden obtener parámetros de control del algoritmo evolutivo que sean mejores que otros. En teoría se podría pensar que dados unos parámetros de control que sean adecuados para un problema concreto sería deseable que también lo fueran para otros problemas del mismo tipo. Esto se puede justificar si se tiene en cuenta que los parámetros de control definen el porcentaje de los nuevos individuos que se basa en información precedente de los mejores encontrados hasta el momento y el que se obtiene de forma aleatoria. Lo que se pretende en este apartado es obtener un conjunto de parámetros de control que funcione razonablemente bien para la optimización de forma utilizando el algoritmo genético Moga-II del programa modeFRONTIER.

El citado algoritmo es estocástico por lo que, como se ha comentado antes, la metodología a aplicar consistirá en conjuntos de ejecuciones de un mismo problema de forma que los resultados obtenidos sean lo más generales posible.

Aunque la evaluación de los parámetros de control no en sí mismo uno de los objetivos de este trabajo, se ha realizado un estudio previo a comenzar a estudiar el efecto que tienen sobre la velocidad de convergencia del algoritmo la parametrización, la FO, y el error de discretización en aquellos casos en los que se utilice el MEF, ya que de este modo se ahorraría mucho coste computacional. Los resultados obtenidos con los diferentes algoritmos probados para diferentes parametrizaciones y FO se muestran a continuación en la siguiente tabla.



| MODELO | TIPO VARIABLE | Nº EJECUCIONES | Nº ITERACIONES | CONVERGIDO |
|---------|---------------|----------------|----------------|------------|
| Moga-II | Discreta | 8 | 1500 | Sí |
| MOSA | Discreta | 1 | 7500 | No |
| NSGA-II | Discreta | 1 | 30000 | No |
| MOPSO | Discreta | 1 | 30000 | No |
| B-BFGS | Continua | 1 | 90000 | No |

Tabla 5 Convergencia diferentes algoritmos

4.2 AJUSTE DE LOS PARÁMETROS

Los parámetros cuya influencia sobre la optimización se ha tratado de comprobar son los que se detallan a continuación:

- Número de individuos de la población inicial para un mismo número de generaciones.
- Probabilidad de cruzado direccional.
- Probabilidad de selección.
- Probabilidad de mutación.

De estos factores tanto la probabilidad de selección como el cruzado direccional son factores que tratan de preservar la información de los mejores individuos encontrados hasta el momento (elitismo). En cambio, la probabilidad de mutación sirve para ampliar la búsqueda de forma aleatoria.

Para determinar la influencia de cada uno de los parámetros en la optimización se han tomado dos valores diferentes de cada uno de los parámetros de control dentro de su rango de validez, y además se han aplicado a dos problemas diferentes a fin de obtener resultados más generales. Los problemas que se han utilizado, para realizar el estudio, han sido el del cilindro sometido a presión interna y el de la viga en voladizo con una carga puntual aplicada en un extremo. También cabe destacar que, para este último problema se ha comprobado también la influencia de los parámetros de control al utilizar dos funciones objetivo distintas simultáneamente.

En primer lugar se especificarán las combinaciones de parámetros con las cuales se han realizado las diferentes optimizaciones:

| Combinación | Cruzado λ_1 | Mutación λ_2 | Población λ_3 | Selección λ_4 |
|-------------|------------------------|-------------------------|--------------------------|--------------------------|
| 1 | 0,3 | 0,05 | 10 | 0,05 |
| 2 | 0,3 | 0,05 | 10 | 0,2 |
| 3 | 0,3 | 0,05 | 20 | 0,05 |
| 4 | 0,3 | 0,05 | 20 | 0,2 |
| 5 | 0,3 | 0,2 | 10 | 0,05 |
| 6 | 0,3 | 0,2 | 10 | 0,2 |
| 7 | 0,3 | 0,2 | 20 | 0,05 |
| 8 | 0,3 | 0,2 | 20 | 0,2 |
| 9 | 0,5 | 0,05 | 10 | 0,05 |
| 10 | 0,5 | 0,05 | 10 | 0,2 |
| 11 | 0,5 | 0,05 | 20 | 0,05 |
| 12 | 0,5 | 0,05 | 20 | 0,2 |
| 13 | 0,5 | 0,2 | 10 | 0,05 |
| 14 | 0,5 | 0,2 | 10 | 0,2 |
| 15 | 0,5 | 0,2 | 20 | 0,05 |
| 16 | 0,5 | 0,2 | 20 | 0,2 |

Tabla 6 Combinaciones de parámetros de optimización

Se han utilizado dos poblaciones diferentes, una con 10 individuos y otra con 20. Dichas poblaciones se explican en los apartados correspondientes a los problema de la viga (4.2.1 Viga4.2.1) y del cilindro (4.2.2). La semilla de



aleatoriedad del algoritmo Moga-II se ha tomado de forma que los resultados dependan del reloj de la CPU del ordenador, lo que asegura que se utiliza un algoritmo completamente estocástico. Se ha fijado un número máximo de generaciones de 250 en el problema de la viga y de 750 en el del cilindro, puesto que su velocidad de convergencia es menor al tratarse de un problema más complejo. Si algún resultado ha alcanzado este valor sin encontrar el resultado óptimo se ha señalado convenientemente en la tabla marcando las casillas correspondientes en color naranja.

Se han realizado diferentes ejecuciones con cada combinación de parámetros, a fin de obtener unos resultados promediados de la influencia de cada una de ellas sobre la velocidad de convergencia.

4.2.1 Viga

Se ha partido de una población de 20 individuos generada mediante algoritmo *Random*. Este algoritmo basa el diseño de experimentos en una secuencia aleatoria, es decir se rellena dicho espacio con una distribución uniforme, donde la secuencia de puntos queda determinada por el valor de la semilla de aleatoriedad. En este caso se ha tomado semilla de aleatoriedad 1, lo que significa que se ha generado de forma independiente de la hora y que se puede repetir la misma secuencia en cualquier momento. La población de 10 individuos se ha generado tomando los 10 primeros de la de 20.

| y1 | y2 | y3 | y4 | y5 |
|---------|---------|---------|---------|---------|
| 0,07387 | 0,04171 | 0,02060 | 0,03367 | 0,09698 |
| 0,00050 | 0,09698 | 0,09397 | 0,09497 | 0,09397 |
| 0,03970 | 0,03467 | 0,02965 | 0,05075 | 0,01156 |
| 0,07789 | 0,06583 | 0,01558 | 0,03769 | 0,01357 |
| 0,06985 | 0,08090 | 0,00050 | 0,05276 | 0,07487 |
| 0,01457 | 0,04874 | 0,05477 | 0,05779 | 0,02060 |
| 0,06281 | 0,01859 | 0,00151 | 0,01658 | 0,01759 |
| 0,05477 | 0,09799 | 0,02462 | 0,03970 | 0,02161 |
| 0,04372 | 0,02362 | 0,08894 | 0,00352 | 0,05980 |
| 0,06583 | 0,01156 | 0,06583 | 0,09899 | 0,02060 |
| 0,03769 | 0,04673 | 0,03367 | 0,04472 | 0,05075 |
| 0,10000 | 0,06382 | 0,09095 | 0,05075 | 0,04975 |
| 0,04271 | 0,03065 | 0,07186 | 0,09698 | 0,02060 |
| 0,01759 | 0,05477 | 0,05578 | 0,05879 | 0,07889 |
| 0,06985 | 0,02060 | 0,02563 | 0,07789 | 0,02261 |
| 0,09899 | 0,08090 | 0,08392 | 0,01658 | 0,06382 |
| 0,00050 | 0,06382 | 0,02060 | 0,08894 | 0,07085 |
| 0,07286 | 0,05075 | 0,09899 | 0,01558 | 0,07186 |
| 0,08291 | 0,01256 | 0,06181 | 0,04874 | 0,00955 |
| 0,06181 | 0,00251 | 0,05980 | 0,00754 | 0,00754 |

Tabla 7 Población inicial de la viga de 20 individuos y parametrización de 5 puntos con coordenada fija en x y variable en y.

Las variables de la población inicial están definidas la Fig. 19 del correspondiente apartado de parametrización.

En la primera tanda se utiliza como función objetivo el volumen de la viga y se trata de resolver el correspondiente problema de minimización y en la siguiente tabla se pueden ver los resultados obtenidos, que aparecen ordenados según las combinaciones de parámetros con las que se ha conseguido obtener una velocidad de convergencia más alta (promedio del número de iteraciones más bajo).

| Combinación | Iteración 1 | Iteración 2 | Iteración 3 | Promedio | Desviación |
|-------------|-------------|-------------|-------------|----------|------------|
| 3 | 1433 | 1064 | 1384 | 1294 | 200 |
| 9 | 1055 | 1357 | 1577 | 1330 | 262 |
| 16 | 962 | 1521 | 1594 | 1359 | 346 |
| 11 | 995 | 1761 | 1531 | 1429 | 393 |
| 4 | 1127 | 822 | 2632 | 1527 | 969 |
| 13 | 1699 | 1515 | 1451 | 1555 | 129 |
| 2 | 1183 | 2500 | 1182 | 1622 | 761 |
| 1 | 1615 | 1536 | 1924 | 1692 | 205 |
| 8 | 1365 | 1451 | 2294 | 1703 | 513 |
| 12 | 2320 | 842 | 2503 | 1888 | 911 |
| 5 | 1031 | 2500 | 2500 | 2010 | 848 |
| 6 | 1971 | 2500 | 2500 | 2324 | 305 |
| 10 | 5000 | 818 | 1200 | 2339 | 2312 |
| 14 | 2500 | 2284 | 2268 | 2351 | 130 |
| 7 | 2792 | 2523 | 5000 | 3438 | 1359 |
| 15 | 5000 | 5000 | 5000 | 5000 | 0 |

Tabla 8 Resultados de las 16 combinaciones de parámetros para el problema de referencia de la viga con el volumen como FO. Combinaciones coloreadas no ha convergido y la promediadas no son válidas.

En el gráfico que se muestra a continuación se ha representado el promediado de las tres iteraciones frente a su correspondiente combinación de parámetros.

Iteraciones Promediadas Viga

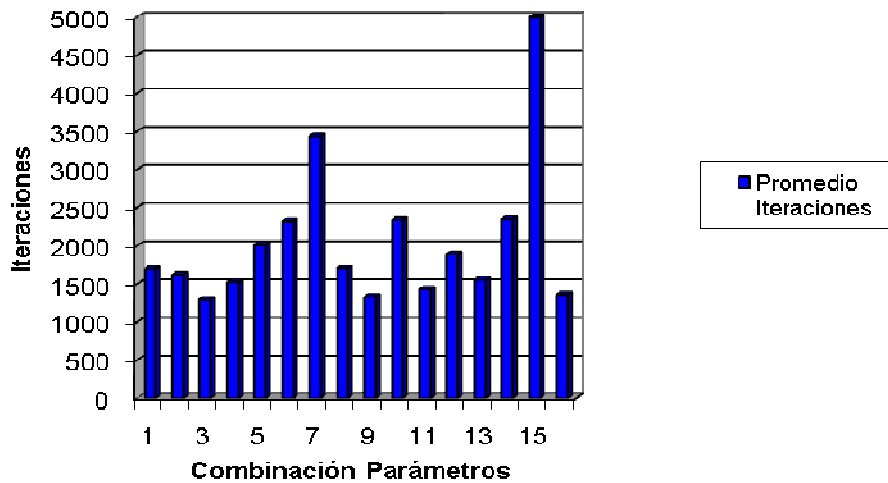


Fig. 5 Gráfica del número de iteraciones promediadas de la viga frente a su correspondiente combinación de parámetros con el volumen comoFO

En este gráfico se puede observar que la mayor parte de combinaciones de parámetros funcionan razonablemente bien y de forma similar en este problema, exceptuando aquellas (5, 6, 7, 10, 14 y 15) en las que el promedio de iteraciones supera los 2000 individuos porque no se ha producido la convergencia de todas las ejecuciones y que por tanto se considerarán como no válidas.

En cualquier caso, la combinación con la que se ha obtenido una mejor velocidad de convergencia ha sido la número 3, seguida muy de cerca por la 9. Ahora bien, si se tiene en cuenta la desviación de los resultados obtenidos la número 13 es algo más estable que las dos anteriores, aunque sea un poco más lenta en cuanto a velocidad de convergencia.

Para ver la influencia de los parámetros y como se interrelacionan entre sí se ha obtenido su correspondiente matriz de correlación. Teniendo en cuenta que únicamente se han tomado las 8 mejores combinaciones válidas de parámetros para calcularla. (En caso de tomarlas todas no aparecía correlación alguna entre ellos)

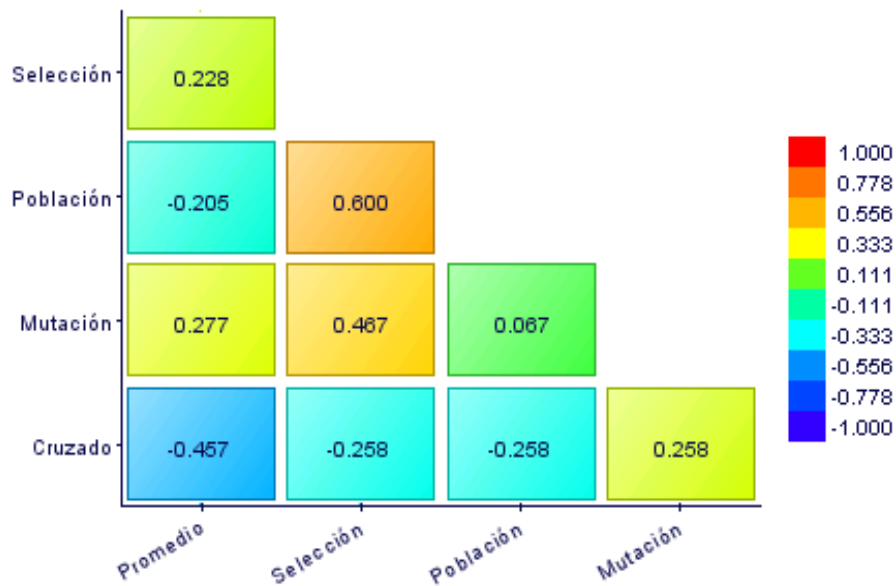


Fig. 6 Matriz de correlación de las 8 mejores combinaciones de parámetros para la viga con el volumen como FO

De esta matriz se puede deducir que los parámetros que están más relacionados entre sí son población (λ_3) y mutación (λ_2) con selección (λ_4). Y por otro lado, el cruzado (λ_1) con el promediado de las iteraciones (λ_5).

En segundo lugar se ha realizado una nueva tanda de ejecuciones del problema de la viga para todas las combinaciones de parámetros que aparecían en la Tabla 6, teniendo en cuenta ahora que la función objetivo utilizada es: $\min \int (\sigma_u - \sigma)^2 dV$. Los resultados obtenidos se pueden ver a continuación:

| Combinación | Iteración 1 | Iteración 2 | Iteración 3 | Promedio | Desviación |
|-------------|-------------|-------------|-------------|----------|------------|
| 9 | 817 | 663 | 803 | 761 | 85 |
| 1 | 700 | 1505 | 1622 | 1276 | 502 |
| 13 | 1558 | 1104 | 2070 | 1577 | 483 |
| 8 | 2488 | 886 | 1374 | 1583 | 821 |
| 12 | 2583 | 1430 | 1108 | 1707 | 776 |
| 10 | 2378 | 1247 | 1577 | 1734 | 582 |
| 7 | 1100 | 1750 | 2452 | 1767 | 676 |
| 11 | 3435 | 1510 | 871 | 1939 | 1335 |
| 14 | 1921 | 1481 | 2439 | 1947 | 480 |
| 5 | 1587 | 2500 | 1979 | 2022 | 458 |
| 16 | 2819 | 1848 | 1482 | 2050 | 691 |
| 4 | 1907 | 2043 | 2355 | 2102 | 230 |
| 6 | 2500 | 1420 | 2500 | 2140 | 624 |
| 2 | 1935 | 2168 | 2500 | 2201 | 284 |
| 3 | 1854 | 3201 | 1781 | 2279 | 800 |
| 15 | 4059 | 1950 | 2819 | 2943 | 1060 |

Tabla 9 Resultados obtenidos para las diferentes combinaciones de parámetros de control para el problema de la viga con la integral de tensiones como FO. Combinaciones coloreadas no ha convergido y la promediadas no son válidas.

En este caso, las combinaciones 2, 5 y 6 se pueden descartar como combinaciones no válidas puesto que existen ejecuciones que no han convergido.

Y en cuanto al resto, como se puede ver en el gráfico de barras de la Fig. 7, la combinación número 9 funciona mejor que el resto, tanto porque es con la que se consigue una menor velocidad de convergencia como por la estabilidad de los resultados obtenidos.

Iteraciones Promediadas Viga

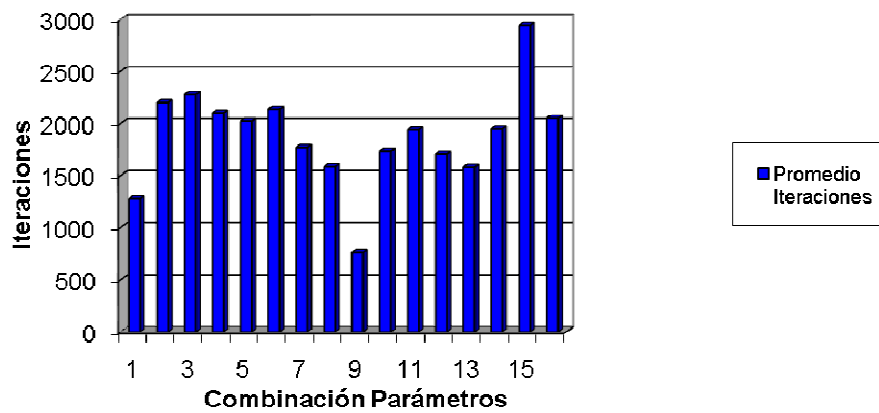


Fig. 7 Gráfica del número de iteraciones promediadas de la viga frente a su correspondiente combinación de parámetros con la integral de tensiones como FO

La gráfica de la Fig. 8 muestra la matriz de correlación, que se ha calculado del mismo modo que la anterior (Fig. 6).

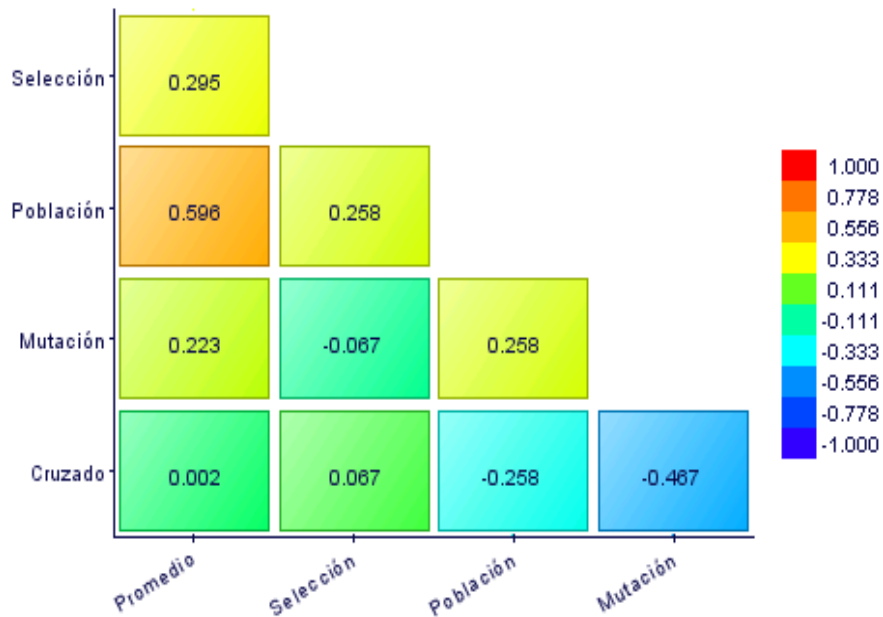


Fig. 8 Matriz de correlación de las 8 mejores combinaciones de parámetros para la viga con la integral de tensiones como FO

En este caso los parámetros que están más relacionados entre sí son la población (λ_3) y el promedio (λ_5), y por otra parte, la mutación (λ_2) y el cruzado (λ_1).

4.2.2 Cilindro

La población inicial de este problema se ha generado siguiendo la misma metodología que en el problema de la viga pero utilizando un algoritmo denominado Sobol.

Este algoritmo basa el diseño de experimentos en una secuencia pseudo-aleatoria, que funciona adecuadamente cuando existen entre 2 y 20 variables, que los distribuye uniformemente en el espacio de diseño.



| x1 | x2 | y3 |
|----------|----------|----------|
| 27,62242 | 27,02302 | 27,62242 |
| 16,41121 | 15,51151 | 16,41121 |
| 38,83363 | 38,53453 | 38,83363 |
| 22,01682 | 32,77878 | 22,01682 |
| 44,43924 | 9,75576 | 44,43924 |
| 10,80561 | 44,29029 | 10,80561 |
| 33,22803 | 21,26727 | 33,22803 |
| 19,19159 | 47,14515 | 47,21962 |
| 41,61401 | 24,12212 | 24,79720 |
| 7,98038 | 35,63363 | 36,00841 |
| 30,40280 | 12,61061 | 13,58599 |
| 13,58599 | 18,36637 | 30,40280 |
| 36,00841 | 41,38939 | 7,98038 |
| 24,79720 | 6,85485 | 41,61401 |
| 47,21962 | 29,87788 | 19,19159 |
| 26,18739 | 25,54955 | 12,19580 |
| 48,60981 | 48,57257 | 34,61822 |
| 14,97618 | 14,03804 | 23,40701 |
| 37,39860 | 37,06106 | 45,82943 |
| 9,37057 | 42,81682 | 17,80140 |

Tabla 10 Población inicial del cilindro de 20 individuos y parametrización de $x_2=y_2$

Las variables de la población inicial, al igual que en el caso anterior, están definidas en el apartado de parametrización en la Fig. 22.

En el caso del problema del cilindro se han realizado dos tandas de análisis de combinaciones de parámetros, una para FO. La única diferencia radica en que para estas tandas se han realizado análisis únicamente con aquellas combinaciones válidas que quedan entre las mejores de las analizadas y son comunes a ambas FO, y que en concreto son la 1, 8, 9, 11 y 13.



En primer lugar se ha resuelto el problema con el volumen como FO, obteniendo los siguientes resultados:

| Combinación | Iteración 1 | Iteración 2 | Iteración 3 | Media | Desviación |
|-------------|-------------|-------------|-------------|-------|------------|
| 13 | 1202 | 2478 | 4426 | 2702 | 1624 |
| 11 | 6961 | 1307 | 1516 | 3261 | 3206 |
| 9 | 1609 | 5364 | 6192 | 4388 | 2442 |
| 1 | 799 | 7500 | 7500 | 5266 | 3869 |
| 8 | 15000 | 12629 | 15000 | 14210 | 1369 |

Tabla 11 Resultados obtenidos para las 5 combinaciones de parámetros de control del problema del cilindro con el volumen como FO. Combinaciones coloreadas no ha convergido y la promediadas no son válidas.

La combinación que mejor funciona en cuanto a velocidad de convergencia y estabilidad es la 13. Y se puede ver también que, aunque la 11 presenta una mayor velocidad de convergencia que la 9 es menos estable.

Por otro lado, ninguna de las otras dos funcionan para este tipo de problema puesto que alcanzar la convergencia requeriría un coste computacional muy elevado e invertir mucho tiempo.

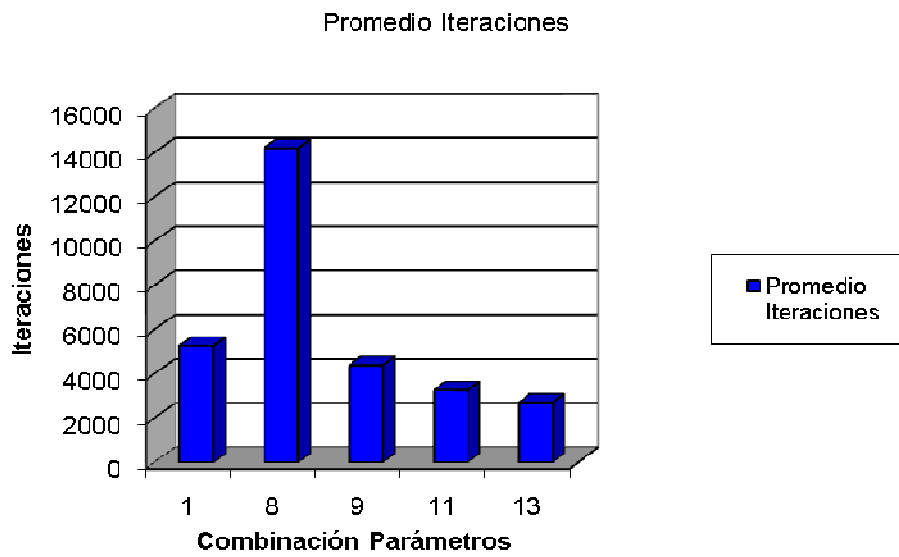


Fig. 9 Gráfica del número de iteraciones promediadas del cilindro frente a su correspondiente combinación de parámetros con el volumen como FO

En la matriz de correlaciones se puede observar que todos los parámetros están muy relacionados con el promedio del número de iteraciones (λ_5) y con la selección (λ_4).

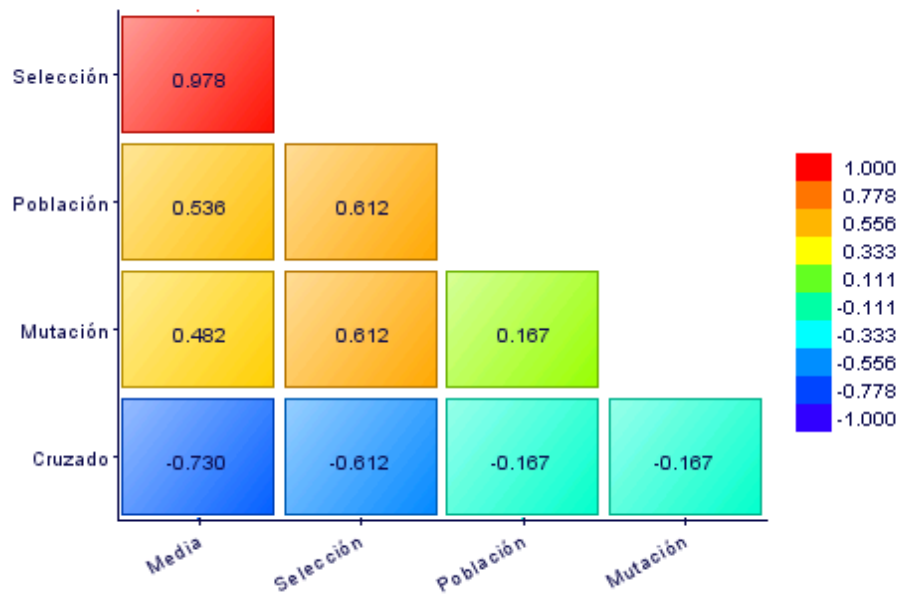


Fig. 10 Matriz de correlación de las combinaciones de parámetros del cilindro con el volumen como FO

En segundo lugar, se ha resuelto el problema del cilindro teniendo en cuenta que la función objetivo es $\min \int (\sigma_u - \sigma)^2 dV$. Los resultados son los que se muestran a continuación:

| Combinación | Iteración 1 | Iteración 2 | Iteración 3 | Media | Desviación |
|-------------|-------------|-------------|-------------|-------|------------|
| 11 | 1627 | 5499 | 2413 | 3180 | 2047 |
| 13 | 2970 | 6007 | 4191 | 4389 | 1528 |
| 9 | 6836 | 2014 | 7058 | 5303 | 2850 |
| 1 | 7486 | 4266 | 7500 | 6417 | 1863 |
| 8 | 15000 | 15000 | 4880 | 11627 | 5843 |

Tabla 12 Resultados obtenidos para las 5 combinaciones de parámetros de control del problema del cilindro con la integral de tensiones como FO. Combinaciones coloreadas no ha convergido y la promediadas no son válidas.

En este caso, los parámetros más adecuados al problema ha resultado ser los de la combinación 11, aunque es más estable la 13 a pesar de presentar una velocidad de convergencia más baja. En cambio, los de la 8 no son adecuados puesto que no ha convergido en dos de las ejecuciones realizadas.

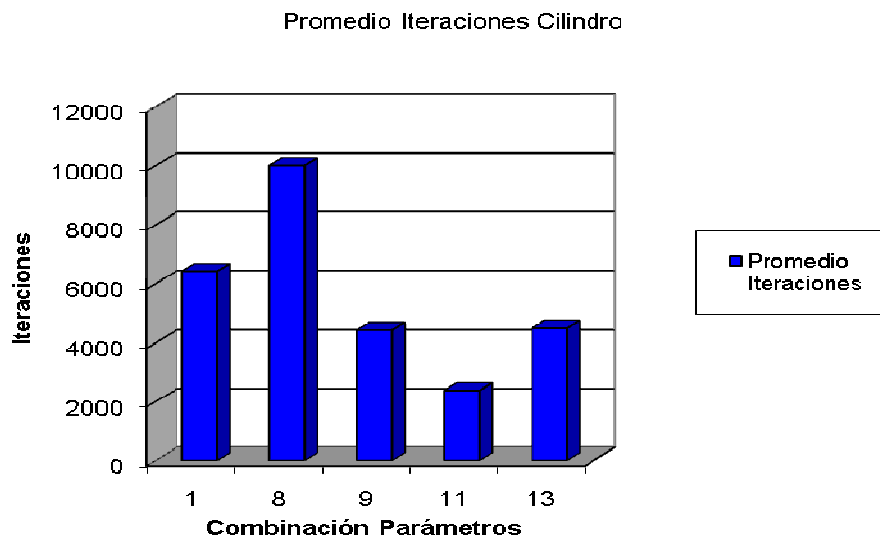


Fig. 11 Gráfica del número de iteraciones promediadas del cilindro frente a su correspondiente combinación de parámetros con la integral de tensiones como FO

Si se tiene en cuenta la correlación entre parámetros se ve claramente como la selección (λ_4) y el cruzado (λ_1) están fuertemente relacionados con la media (λ_5). Y de nuevo, todos los parámetros están bastante relacionados con la selección (λ_4).

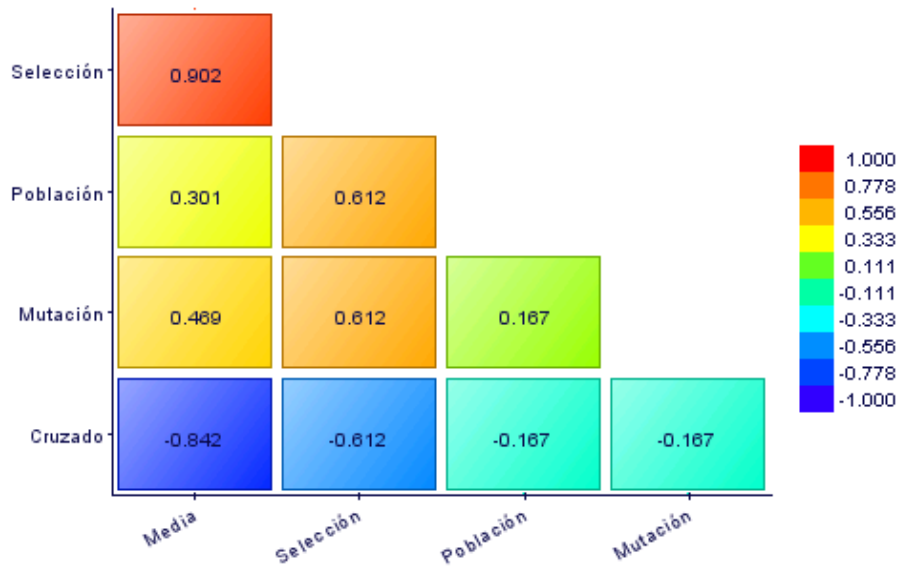


Fig. 12 Matriz de correlación de las combinaciones de parámetros del cilindro con la integral en tesiones como FO

También se realizará una comparación de los resultados obtenidos con cada combinación de parámetros con la que tiene definida por defecto el algoritmo MOGA-II de modeFRONTIER, y que se detallan en la siguiente tabla:

| Combinación | Cruzado | Mutación | Población | Selección |
|--------------|---------|----------|-----------|-----------|
| modeFRONTIER | 0,5 | 0,1 | 10 | 0,05 |

Tabla 13 Parámetros de optimización de referencia de modeFRONTIER

4.3 CONCLUSIONES

A la vista de los resultados anteriores, se puede concluir que los parámetros que utiliza modeFRONTIER por defecto son muy similares a los de la



combinación de parámetros número 9 y funcionan razonablemente, sobre todo en el problema más sencillo del cálculo de una viga.

En este problema, para dicha combinación de parámetros se obtiene una velocidad de convergencia elevada para una misma población inicial y con independencia de la función objetivo utilizada. Y al mismo tiempo la dispersión de los resultados obtenidos es muy baja, lo que garantiza una mayor estabilidad del tiempo de convergencia del problema.

En cambio, en el problema del cilindro las mejores velocidades de convergencia se han obtenido para las combinaciones 11 (FO integral de tensiones) y 13 (FO volumen). No obstante, la más estable en cuanto a dispersión de los resultados es la 13 en ambos casos.

Así pues, si analizamos en conjunto las tres mejores combinaciones obtenidas para este último problema se puede apreciar que cuando la población inicial es más elevada se requiere una menor capacidad de mutación (λ_2) de los genes y viceversa. Esto parece lógico puesto que λ_2 es el parámetro que permite disminuir o aumentar, respectivamente, la probabilidad con la cual una configuración de diseño cambia aleatoriamente, y cuanto menor sea la población inicial más cambios serán necesarios en la genética de los individuos.

Por otra parte, se puede constatar que cuando la FO a tratar es el volumen es más efectiva una mayor aleatoriedad en el diseño, ya que como se verá en las conclusiones del apartado 5.3 la evolución de la forma del cilindro es bastante más irregular que cuando la FO a minimizar es el volumen.

Sin embargo, el resto de parámetros son los que utiliza por defecto modeFRONTIER. Estos valores son razonables debido a que un valor de cruzado (λ_1) de 0,5 da eficiencia al algoritmo porque se trata de un buen compendio entre la obtención de individuos con mejores características y el mantenimiento de la diversidad. Y un valor de selección (λ_4) de 0,05 también consigue buenos resultados puesto que se encarga de mantener una buena diversidad entre puntos.

De todo esto, se puede deducir, que los parámetros que utiliza modeFRONTIER por defecto en general son válidos, pero que en función del problema (sobre todo si es complejo) a resolver se debería hacer un estudio previo para ver si es posible obtener alguna combinación de parámetros que permita mejorar la velocidad de convergencia, como es el caso del cilindro.



5 FUNCIÓN OBJETIVO

5.1 INTRODUCCIÓN

Como ya se ha comentado anteriormente la función objetivo se usa para clasificar diseños. Generalmente se trata de resolver problemas de minimización en los que puede haber una o más funciones objetivo. Este trabajo se centrará en problemas mono-objetivo aunque se han realizado pruebas con dos funciones objetivo simultáneamente como se verá en el apartado **¡Error! No se encuentra el origen de la referencia.**, pero se trata en cualquier caso de dos funciones complementarias que en tienen una solución común.

5.2 COMPARACIÓN DE RESULTADOS

Se han realizado optimizaciones con diferentes funciones objetivo para el problema de referencia de la viga para determinar su influencia en la velocidad de convergencia. Para cada tipo de función objetivo se han cinco optimizaciones utilizando diferentes poblaciones iniciales para obtener un resultado más general.

5.2.1 Viga

En primer lugar se ha fijado como función objetivo la minimización del volumen de la viga y posteriormente se han tomado diferentes integrales volumétricas en función de la tensión.

A continuación se va a plantear una de las integrales volumétricas, siendo el análogo el planteamiento y desarrollo de las demás funciones objetivo. Para el desarrollo de la integral se ha de tener en cuenta que las ecuaciones que definen el spline están planteadas de forma paramétrica, y son función de un parámetro t , cuyos valores oscilan entre 0 y 1 en cada uno de los tramos del spline que estarán regidos por diferentes ecuaciones.

$$\begin{aligned} \sigma(x, y) &= \sigma_{\max} \cdot \frac{y}{h} = \frac{6 \cdot P \cdot (L-x)}{4 \cdot b \cdot h^3} \cdot y \\ I &= 2 \cdot b \cdot \int_0^h \int_0^L \left[\sigma_u - \frac{6 \cdot P \cdot (L-x)}{4 \cdot b \cdot h(x)^3} \cdot y \right]^2 dx dy = \\ &= 2 \cdot b \cdot \int_0^h \int_0^L \left[\sigma_u^2 + \frac{36 \cdot P^2 \cdot (L-x)^2}{16 \cdot b^2 \cdot h^6} \cdot y^2 - \frac{6 \cdot P \cdot (L-x) \cdot \sigma_u}{2 \cdot b \cdot h^3} \cdot y \right] dx dy = \\ &= 2 \cdot b \cdot \int_0^h \int_0^L \left[\sigma_u^2 + \frac{36 \cdot P^2 \cdot (L-x(t))^2}{16 \cdot b^2 \cdot h(t)^6} \cdot y(t)^2 - \frac{6 \cdot P \cdot (L-x(t)) \cdot \sigma_u}{2 \cdot b \cdot h(t)^3} \cdot y(t) \right] \cdot x'(t) \cdot dt \cdot y'(t) \cdot dt = \\ &= \sigma_u^2 \cdot V + 2 \cdot \int_0^1 \frac{36 \cdot P^2 \cdot (L-x(t))^2}{16 \cdot b^2 \cdot h(t)^6} \cdot \left[\int_0^{h(t)} y^2 \cdot dy \right] \cdot x'(t) \cdot dt - \int_0^1 \frac{6 \cdot P \cdot (L-x(t)) \cdot \sigma_u}{2 \cdot b \cdot h(t)^3} \cdot \left[\int_0^{h(t)} y^2 \cdot dy \right] \cdot x'(t) \cdot dt \\ I &= \sigma_u^2 \cdot V + \frac{3}{2} \cdot \frac{P^2}{b} \cdot \int_0^1 \frac{[L-x(t)]^2}{h(t)^3} \cdot x'(t) \cdot dt - 3 \cdot P \cdot \sigma_u \cdot \int_0^1 \frac{[L-x(t)]}{h(t)} \cdot x'(t) \cdot dt \end{aligned}$$

Y posteriormente, se han evaluado las mismas funciones objetivo pero enfocando el problema desde un punto de vista distinto como si se tratase de una optimización multi-criterio. Es decir, se han tratado de minimizar simultáneamente el volumen junto con cada una de las integrales volumétricas anteriormente mencionadas, aunque en realidad se trata de un mismo objetivo, puesto que ambas funciones trabajan en una misma dirección.

Los resultados obtenidos se pueden observar en la siguiente tabla:

| Una Función Objetivo (Viga) | P1 | P2 | P3 | P4 | P5 | Media | Desviación |
|--|-----------|-----------|-----------|-----------|-----------|--------------|-------------------|
| $\min V$ | 1772 | 2250 | 2390 | 2057 | 2128 | 2098 | 207 |
| $\min \int \sigma^2 dV$ | 1978 | 2189 | 3079 | 3093 | 1442 | 2171 | 644 |
| $\min \frac{\int \sigma^2 dV}{V}$ | 1096 | 1631 | 1718 | 1718 | 1723 | 1529 | 243 |
| $\min \int (\sigma_u^2 - \sigma^2) dV$ | 803 | 2079 | 1519 | 1193 | 1635 | 1304 | 429 |
| $\min \int (\sigma_u - \sigma)^2 dV$ | 1455 | 1547 | 915 | 1321 | 617 | 1041 | 351 |
| Dos Funciones Objetivo (Viga) | P1 | P2 | P3 | P4 | P5 | Media | Desv. |
| $\min V$ $\min \int \sigma^2 dV$ | 2168 | 1373 | 1140 | 1252 | 2003 | 1486 | 417 |
| $\min V$ $\min \frac{\int \sigma^2 dV}{V}$ | 1276 | 2092 | 1359 | 1675 | 1303 | 1487 | 310 |
| $\min V$ $\min \int (\sigma_u^2 - \sigma^2) dV$ | 1380 | 1501 | 1036 | 1903 | 1604 | 1427 | 283 |
| $\min V$ $\min \int (\sigma_u - \sigma)^2 dV$ | 1684 | 1666 | 1932 | 964 | 1265 | 1413 | 344 |

Tabla 14 Resultados según las funciones objetivo utilizadas para el problema de referencia de la viga

A continuación se muestran diferentes secuencias de imágenes, donde se pueden apreciar algunas de las variaciones de forma que ha ido sufriendo la viga, a medida que iba avanzando la optimización según la función o funciones objetivo utilizadas.

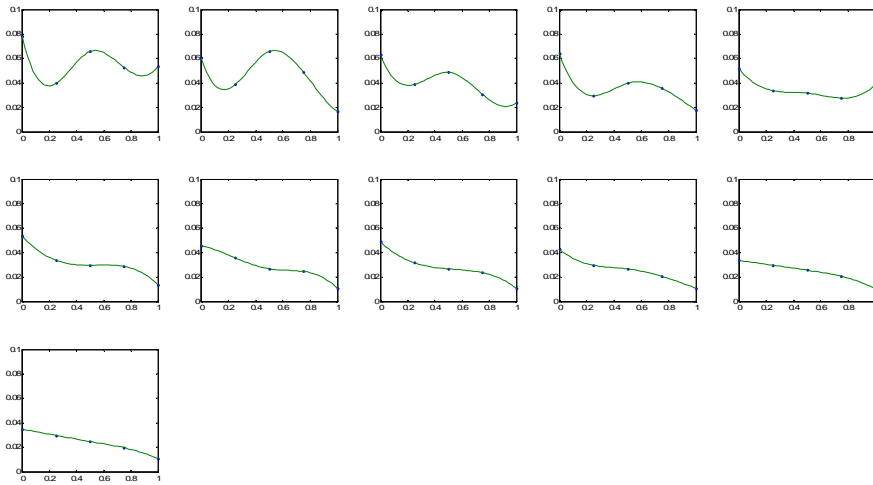


Fig. 13 Evolución de forma de algunos de los mejores individuos de la viga cuando la FO a minimizar es el volumen

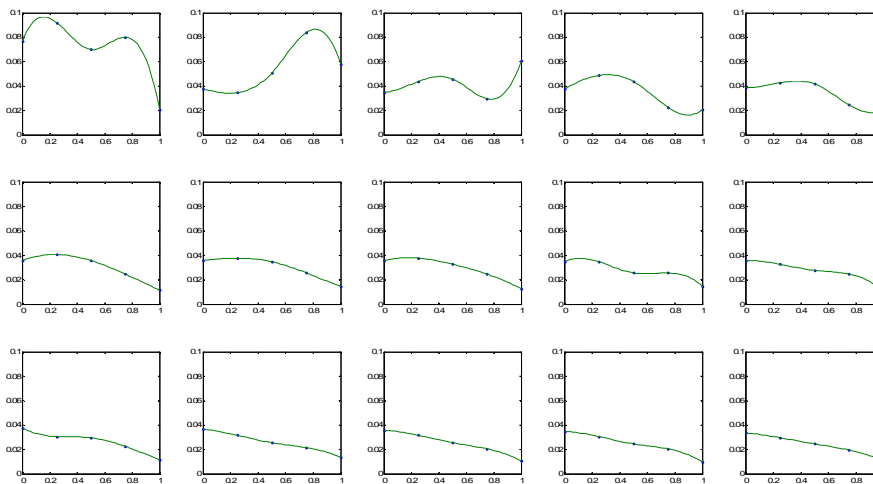


Fig. 14 Evolución de forma de algunos de los mejores individuos de la viga cuando la FO a minimizar es la integral

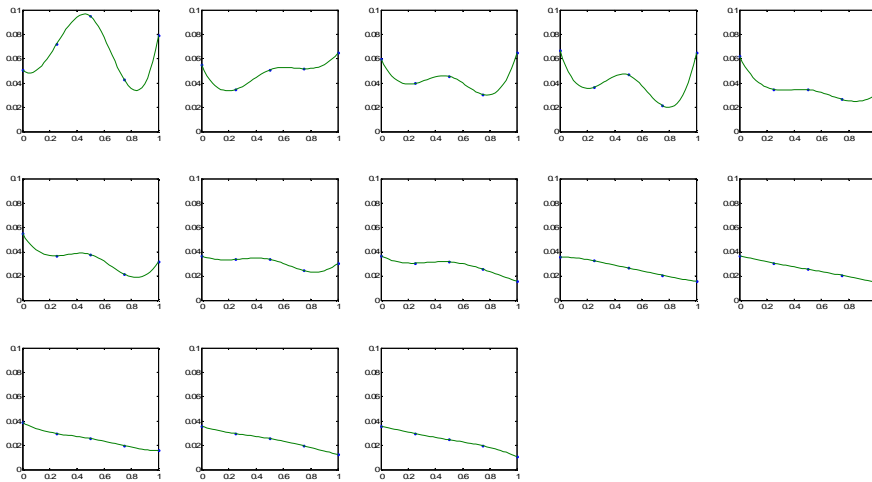


Fig. 15 Evolución de forma de algunos de los mejores individuos de la viga cuando las FO a minimizar son el volumen y la integral

5.2.2 Cilindro

A fin de obtener una mayor generalidad en los resultados, se ha realizado también una comparación de la velocidad de convergencia, utilizando dos funciones objetivo distintas, en el problema de cilindro sometido a presiones internas, siendo los resultados obtenidos:

| Una Función Objetivo (Cilindro) | P1 | P2 | P3 | P4 | P5 | Media | Desviación |
|--------------------------------------|------|------|------|------|------|-------|------------|
| $\min V$ | 2008 | 1412 | 2417 | 3186 | 1538 | 2112 | 721 |
| $\min \int (\sigma_u - \sigma)^2 dV$ | 3960 | 1426 | 4121 | 2092 | 2171 | 2754 | 1211 |

Tabla 15 Resultados según las funciones objetivo utilizadas para el problema de referencia del cilindro

La siguiente figura muestra la evolución de la forma del cilindro cada vez que se ha hallado una mejora al optimizar cuando la función objetivo es la minimización del volumen.

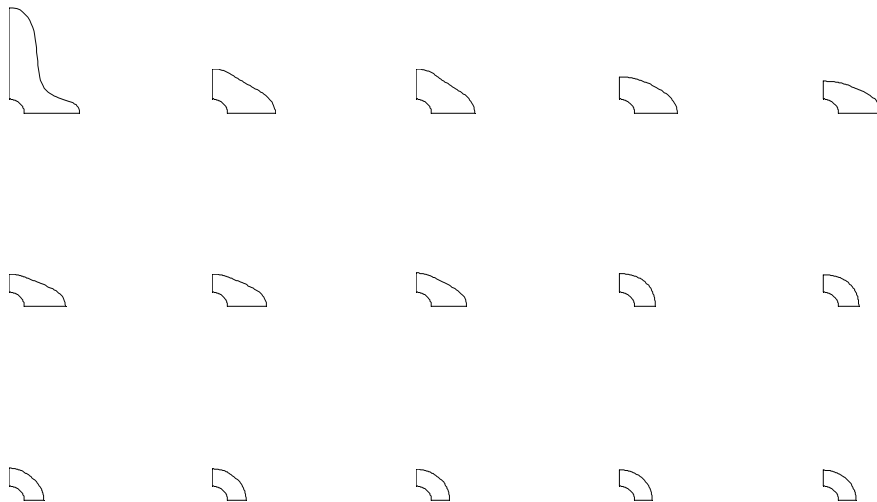


Fig. 16 Evolución de forma de algunos de los mejores individuos del cilindro cuando la FO a minimizar es el volumen

Y si la función objetivo es la minimización de la integral de las tensiones la evolución de la forma del cilindro sometido a presión interna será:

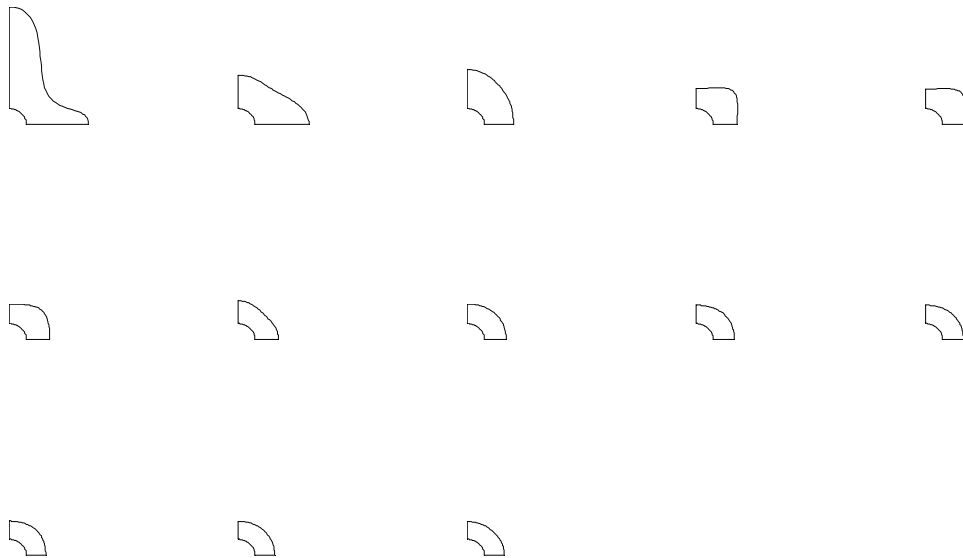


Fig. 17 Evolución de forma de algunos de los mejores individuos del cilindro cuando la FO a minimizar es la integral

5.3 PUENTE

Para comprobar los resultados obtenidos e intentar generalizar en la medida de lo posible los resultados obtenidos, se ha resuelto el problema del puente utilizando las dos funciones objetivo que se emplearon también en la resolución del problema de cilindro, es decir, se ha resuelto el problema tanto con el volumen como con la integral en tensiones. Los resultados obtenidos se muestran a continuación:

| Caso de carga 1 | | |
|--------------------------------------|-----------------------|-----------------------|
| Función Objetivo | Nº Iteraciones | Volumen Mínimo |
| $\min V$ | 4100 | 25,288 |
| $\min \int (\sigma_u - \sigma)^2 dV$ | 1060 | 25,863 |
| Caso de carga 2 | | |
| Función Objetivo | Nº Iteraciones | Volumen Mínimo |
| $\min V$ | 1972 | 36,675 |
| $\min \int (\sigma_u - \sigma)^2 dV$ | 12321 | 36,619 |
| Caso de carga 3 | | |
| Función Objetivo | Nº Iteraciones | Volumen Mínimo |
| $\min V$ | 4673 | 35,201 |
| $\min \int (\sigma_u - \sigma)^2 dV$ | 4323 | 35,207 |
| Caso de carga 4 | | |
| Función Objetivo | Nº Iteraciones | Volumen Mínimo |
| $\min V$ | 1670 | 35,978 |
| $\min \int (\sigma_u - \sigma)^2 dV$ | 1363 | 35,405 |

Tabla 16 Resultados obtenidos para cada caso de carga según la FO utilizada

De la observación de esta tabla se puede deducir que ambas funciones objetivo funcionan de forma bastante similar si exceptuamos los resultados obtenidos para los casos de carga 1y 2, donde parece que el proceso de optimización al ser aleatorio ha tardado más en encontrar valores de volumen parecidos.



5.4 CONCLUSIONES

De las diferentes funciones objetivo que se han probado, en un primer momento para el problema de la viga, se puede observar que la función que presenta una mayor velocidad de convergencia es $\min \int (\sigma_u - \sigma)^2 dV$ frente a la utilizada en gran parte de los problemas de optimización, $\min V$.

Por otra parte, se puede observar que, al resolver la optimización del cilindro la convergencia de la solución, en el caso de la integral, es más lenta pero las soluciones intermedias que va tomando son ya casi desde el principio mucho más estables, es decir, su forma es mucho más parecida a la forma óptima obtenida.

También es importante añadir que, a diferencia del problema de la viga, en el cilindro existen unas condiciones de contorno que debe adoptar el spline que define el contorno exterior de la pieza, por lo que le cuesta más encontrar soluciones factibles.

En los análisis realizados del puente, se han podido observar dos cosas: que a pesar de ser un problema más complejo que el del cilindro (su geometría está definida con un mayor número de variables) la velocidad de convergencia de la optimización es mayor y que la respuesta ante las distintas FO utilizadas es similar.





6 PARAMETRIZACIÓN

6.1 INTRODUCCIÓN

El problema de obtener una parametrización correcta de un análisis consiste en la capacidad de establecer una relación, lo más sencilla posible, entre el espacio de diseño y el espacio de coordenadas. Esto puede llegar a ser un problema bastante complejo si, además se tienen en cuenta las funciones objetivo o las restricciones de los problemas de optimización.

Dentro de la optimización de forma se podrían distinguir dos tipos de parametrización diferentes en función de los requisitos de cada uno.

En primer lugar, se considerará el caso en el cual lo que se pretende optimizar es el contorno exterior de la pieza. En este caso parte del contorno suele ser fijo, ya que, debido a los requisitos estructurales de la pieza en ese tramo del contorno se habrán de imponer condiciones de contorno, tales como cargas o restricciones en desplazamientos. Así pues, lo que se busca es parametrizar de algún modo el espacio de diseño de forma que se limite, de alguna manera, el recorrido de las variables de diseño para acelerar la velocidad de convergencia de la optimización.

En segundo lugar, se va a considerar el caso en el que lo que se pretende es aligerar una pieza introduciendo agujeros en su interior. Ahora bien, en este caso no queda claro el número de agujeros que se deben introducir, ni en que zona del espacio de diseño se deben situar, ni por qué número de puntos debería estar definido cada uno de ellos. Además de que, al variar la geometría interna de la pieza, se verá alterada la forma exterior óptima de la misma. Razones, todas ellas, por las que se puede concluir que se trata de un caso bastante más complejo que el anterior, y que requeriría un estudio profundo que queda fuera de los límites de este trabajo.



6.2 DEFINICIÓN POR PUNTOS. NÚMERO DE COORDENADAS

Para el estudio de parametrización que se pretende realizar en este apartado se han tomado dos problemas de referencia: el de un cilindro sometido a presión interna y el de una viga en voladizo con una carga puntual en el extremo.

6.2.1 Viga

En primer lugar, se ha tratado de optimizar la forma de la viga. Para ello se han definido algunos parámetros constantes como son la longitud y la anchura de la viga, dejando únicamente como variable la altura de la viga en cada sección. No obstante se ha teniendo en cuenta, como ya se comentó en el apartado 1.4, que en el extremo ha de existir una sección mínima para que la viga soporte las tensiones tangenciales debidas a cortadura.

Es importante destacar que, debido a que la viga presenta simetría respecto del eje x ya que se espera una solución antisimétrica asociada al problema de flexión, como se puede observar en la figura, únicamente se ha optimizado la forma de la mitad superior de la viga para agilizar los cálculos.

La viga se ha definido mediante tres líneas rectas y una cuarta definida mediante un spline.

El número de punto mediante los cuales se ha definido el spline en cada caso y de las variables empleadas quedan registradas en la siguiente tabla:



| Nº DE CASO | Nº DE PUNTOS DEL SPLINE | COORDENADAS VARIABLES |
|-------------------|------------------------------------|----------------------------------|
| 1 | 3 | x, y |
| 2 | 3 | y |
| 3 | 5 | x, y |
| 4 | 5 | y |
| 5 | 9 | y |

Tabla 17 N° de puntos y variables que definen el spline correspondiente al contorno de la viga

Se han representado también de modo esquemático las variables y los rangos de validez que definen el espacio de diseño para cada una de ellas. Las variables de los casos 2 y 4 no se han representado debido a que los rangos de validez de la coordenada y son los mismos que en los casos 1 y 3 respectivamente.

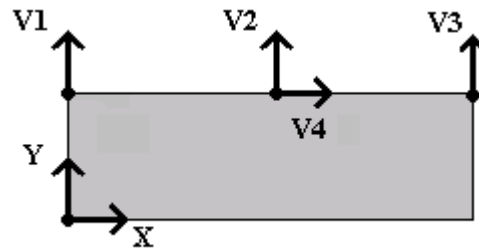


Fig. 18 Esquema de las variables del caso 1

| Variable de diseño | Rango de variable |
|--------------------|-------------------|
| V_1, V_2, V_3 | [0,0005- |
| V_4 | 0,1] |
| | [0,2- |
| | 0,8] |

Tabla 18 Variables del caso 1 y rango de diseño

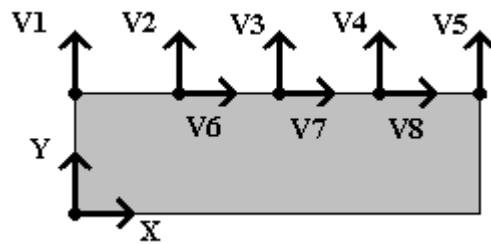


Fig. 19 Esquema de las variables del caso 3

| Variable de diseño | Rango de variable |
|---------------------------|-------------------|
| V_1, V_2, V_3, V_4, V_5 | [0,0005-0,8] |
| V_6 | [0,1-0,35] |
| V_7 | [0,4-0,6] |
| V_8 | [0,65-0,9] |

Tabla 19 Variables del caso 3 y rango de diseño

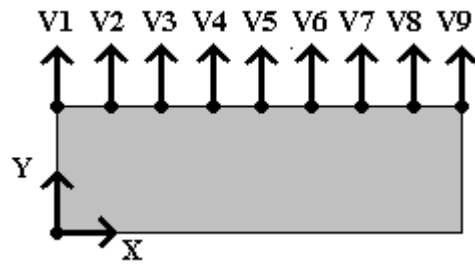
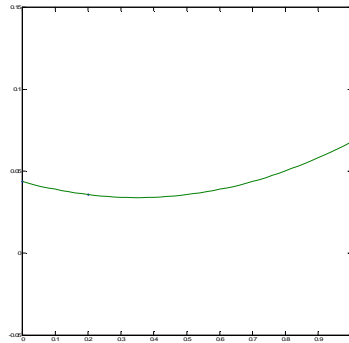


Fig. 20 Esquema de las variables del caso 5

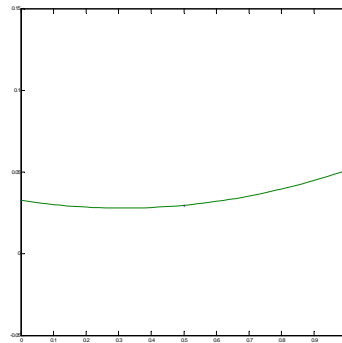
| Variable de diseño | Rango de variable |
|------------------------|-------------------|
| V_1, V_2, \dots, V_9 | [0,0005-0,8] |

Tabla 20 Variables del caso 5 y rango de diseño

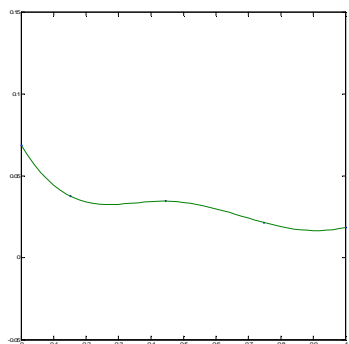
En las siguientes gráficas se puede observar la forma óptima de la viga que se ha obtenido en cada uno de los casos anteriores.



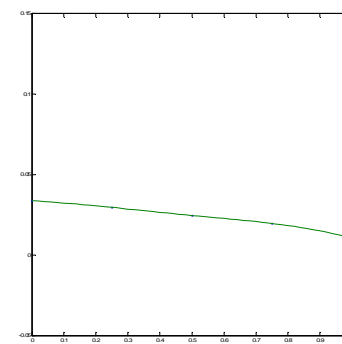
Caso 1



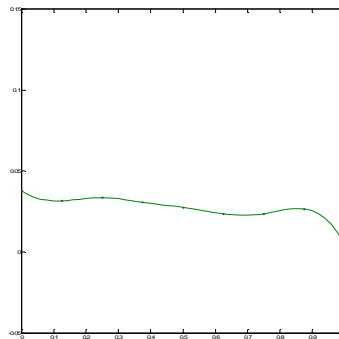
Caso 2



Caso 3



Caso 4



Caso 5

Fig. 21 Formas óptimas de la viga según su definición geométrica



De los resultados obtenidos, que se muestran en las gráficas anteriores, se puede deducir que tres es un número insuficiente de puntos para definir la geometría porque el spline, en este caso, quedará definido como una curva cuadrática. Esto hace que resulte imposible representar con exactitud la forma óptima de la viga (la obtenida en el caso en el cual se ha parametrizado la viga con cinco puntos, manteniendo las coordenadas en x constantes). Como con tres puntos no se puede obtener en ningún caso el volumen mínimo real, el obtenido tras la optimización queda claramente por encima del volumen mínimo obtenido al aumentar al número de puntos de definición del spline.

Por otro lado, se ha comprobado que nueve puntos es un número excesivo ya que hace que el contorno de la viga, al verse obligado a pasar por semejante número de puntos adquiera una forma bastante irregular. Además de que el volumen mínimo obtenido, tras haber analizado el doble de generaciones, queda por encima del mínimo obtenido para el caso 4.

Por tanto, el mejor número de puntos para definir el spline entre los estudiados y que realmente se ajuste a la solución es cinco. En este caso el volumen mínimo obtenido es de $1.45125 \cdot 10^{-3} m^3$.

Si nos fijamos en el número de coordenadas variables, se han realizado dos tipos de optimizaciones. En primer lugar, se ha permitido variar tanto a la coordenada x como a la y en cada punto en las primeras optimizaciones realizadas, tanto para el caso de un spline definido por tres puntos, como para el de cinco. Una vez comprobada la velocidad de convergencia de este tipo de parametrización, se ha tratado de averiguar si al fijar las coordenadas en x , dejando únicamente como variables las coordenadas en y se producía alguna mejora en la velocidad de convergencia. Las coordenadas en x se han definido de forma que los puntos estén equiespaciados y cubran toda la longitud de la viga. De los resultados obtenidos, se puede concluir que la velocidad de convergencia mejora considerablemente ya que, en el caso de cinco puntos y x constantes la convergencia de la solución se había producido antes de analizar 4500 individuos (límite de individuos a analizar), mientras que siendo las x variables en 9000 iteraciones aún no se había producido la convergencia de la solución.



6.2.2 Cilindro

En este apartado se ha optimizado el cilindro, teniendo en cuenta que, la geometría del conducto se crea a partir de cinco puntos, dos de ellos se encuentran fijos y unidos mediante un arco de circunferencia, formando la parte interna del conducto. La parte externa se define mediante un spline, que se crea con los otros tres puntos. Estos tres puntos pueden variar de posición. También se ha fijado la pendiente del spline en los extremos de manera que el principio y final de los splines resulten perpendiculares al contorno de simetría, lo que evita la aparición de soluciones singulares en el caso elástico.

Este problema va a estar caracterizado por la utilización de restricciones entre variables, en particular se han incorporado 2 restricciones geométricas de forma que se rechazan algunas configuraciones generadas por el algoritmo de optimización que se conoce de antemano que serán soluciones no factibles. Y en cuanto a la parametrización se han empleado dos tipos distintos.

Por una parte, en las primeras optimizaciones al igual que ocurría en la viga se ha permitido a las coordenadas 2 y 3 (ver Fig.) que varíen de forma independiente mientras que, más tarde se impone una tercera restricción, en la macro que conecta el modeFRONTIER con el programa de cálculo de elementos finitos que utiliza mallados independientes de la geometría y que consistirá en obligar a que las coordenadas que definen el punto intermedio del contorno exterior tengan el mismo valor. Puesto que, si no se impusiera de este modo esta restricción es extremadamente difícil encontrar geometrías válidas ya que encontrar dos coordenadas iguales generadas aleatoriamente es prácticamente imposible.

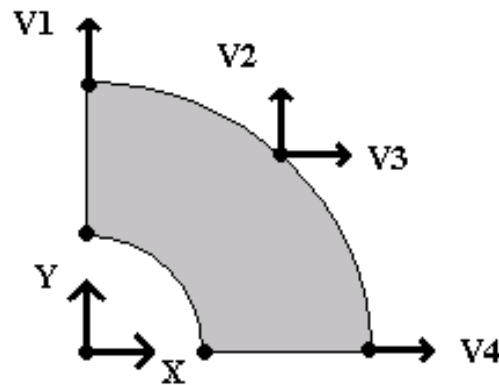


Fig. 22 Esquema variables de diseño del cilindro

| Variable de diseño | Rango de variable | Restricción |
|--------------------|-------------------|-------------------|
| V_1 | [5,2-50,0] | |
| V_2 | [4,0-50,0] | $V_2 < V_1 - 0,5$ |
| V_3 | [4,0-50,0] | |
| V_4 | [5,2-50,0] | $V_4 > V_3 + 0,5$ |

Tabla 21 Datos de las variables de diseño del conducto sometido a presión interna

Con los datos del problema se determina que el radio externo óptimo es 10,670330824461 e implica que para la solución óptima las variables toman los valores mostrados en la siguiente tabla:



| Variable de diseño | Solución analítica |
|------------------------|--------------------|
| V_1 | 10,6703 |
| $\sqrt{V_2^2 + V_3^2}$ | 10,6703 |
| V_4 | 10,6703 |

Tabla 22 Solución analítica del conducto

Debido a que la parte externa se crea con un spline, existe la posibilidad de que el problema posea un número importante de óptimos locales o globales, por lo tanto existen muchas configuraciones con un mismo valor de área y en las que no se supera la tensión máxima admisible de von Mises. Lo que se debe cumplir en cualquier caso es que $\sqrt{V_2^2 + V_3^2} \approx 10,670330824461$, de forma que la hipotenusa formada por estas dos variables de diseño coincida con el radio externo óptimo de la solución analítica. En este problema, la función objetivo es el área del cuarto de sección del conducto. El área de la solución analítica es $A_{opt} = 69,7873077150$.

6.3 PUENTE

El contorno del puente consta de una parte móvil, que estará constituida por los tres arcos, y una fija, que estará formada por el resto de líneas que lo conforman.

En un primer momento cada uno de los vanos se parametrizó mediante un spline. Y a su vez cada spline se definió mediante 5 puntos, siendo fijos los dos puntos extremos de cada uno. Para la definición de las variables correspondientes a las coordenadas de los puntos móviles se consideraron las x fijas siguiendo un espaciado de Tschebyscheff y considerando únicamente como variables sus respectivas coordenadas en y .

Es importante destacar que los resultados en este caso (a diferencia de lo que ocurría en el caso de la viga) no fueron buenos, debido a la incapacidad del spline de reproducir de forma adecuada el contorno de los vanos del puente. En la siguiente figura se muestra el análisis del individuo óptimo hallado en el análisis, en el cual se puede ver como existe un pico sin tensión en la zona central de los vanos que no debería existir, puesto que representa un volumen adicional de la pieza que no aporta nada a la hora de soportar la carga.

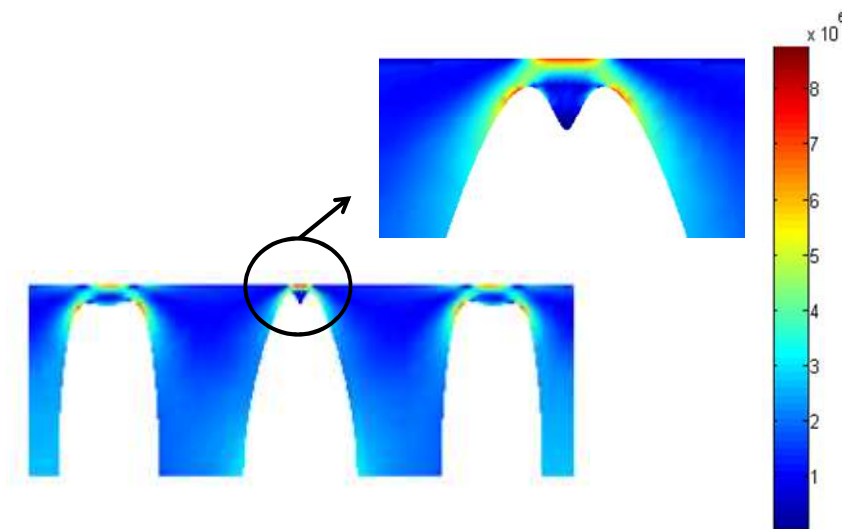


Fig. 23 Tensiones de Von Mises y forma del individuo óptimo obtenido con parametrización del spline de 5 puntos

Posteriormente se pasó a parametrizar el spline mediante un número de puntos inferior, 4 en este caso, siendo fijos nuevamente los puntos de los extremos. En la Fig. 23 aparecen indicadas las variables utilizadas, que por otra parte presentan simetría respecto de eje central del puente.

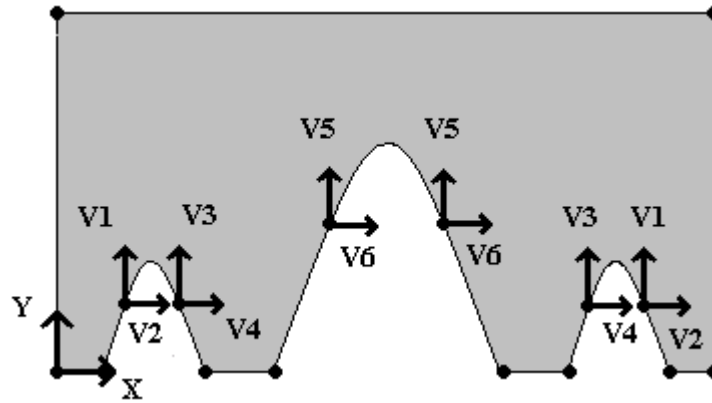


Fig. 24 Esquema de las variables de diseño del puente

En este caso, se ha dejado variar tanto las coordenadas en x como en y de los puntos porque al reducir los puntos de definición del spline si se mantenían las coordenadas en x fijas se pudo comprobar que únicamente se podía representar un grupo de curvas concreto, todas ellas con la misma forma, limitando así de algún modo la solución óptima que se podía obtener.

| Variable de diseño | Rango de variable |
|--------------------|-------------------|
| V_1, V_3, V_5 | [0,5-4,5] |
| V_2 | [1,0-2,7] |
| V_4 | [2,9-3,7] |
| V_5 | [5,2-7,9] |

Tabla 23 Variables del puente y rango de diseño

En la siguiente figura se puede apreciar la forma óptima del puente obtenida al hallar la envolvente inferior de los diferentes casos de carga al utilizar la parametrización anterior.

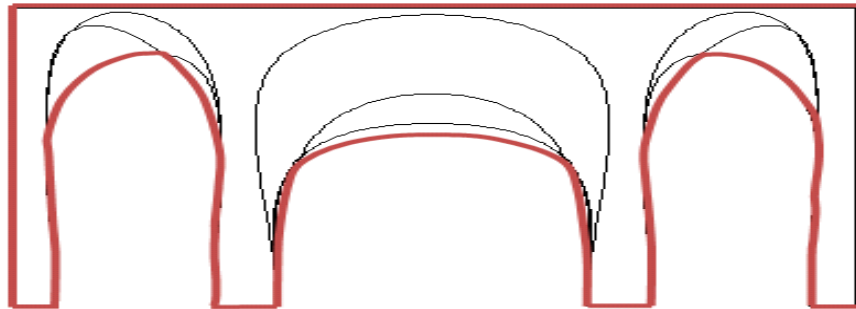


Fig. 25 Forma óptima del puente (en rojo) obtenida a partir de la superposición de las soluciones obtenidas para los distintos casos de carga

En este caso si se observan en la Fig. 25 la forma óptima obtenida para cada caso de carga y la correspondiente distribución de tensiones de von Mises se puede ver como no aparece el problema anterior que parecía al definir el spline mediante 5 puntos.

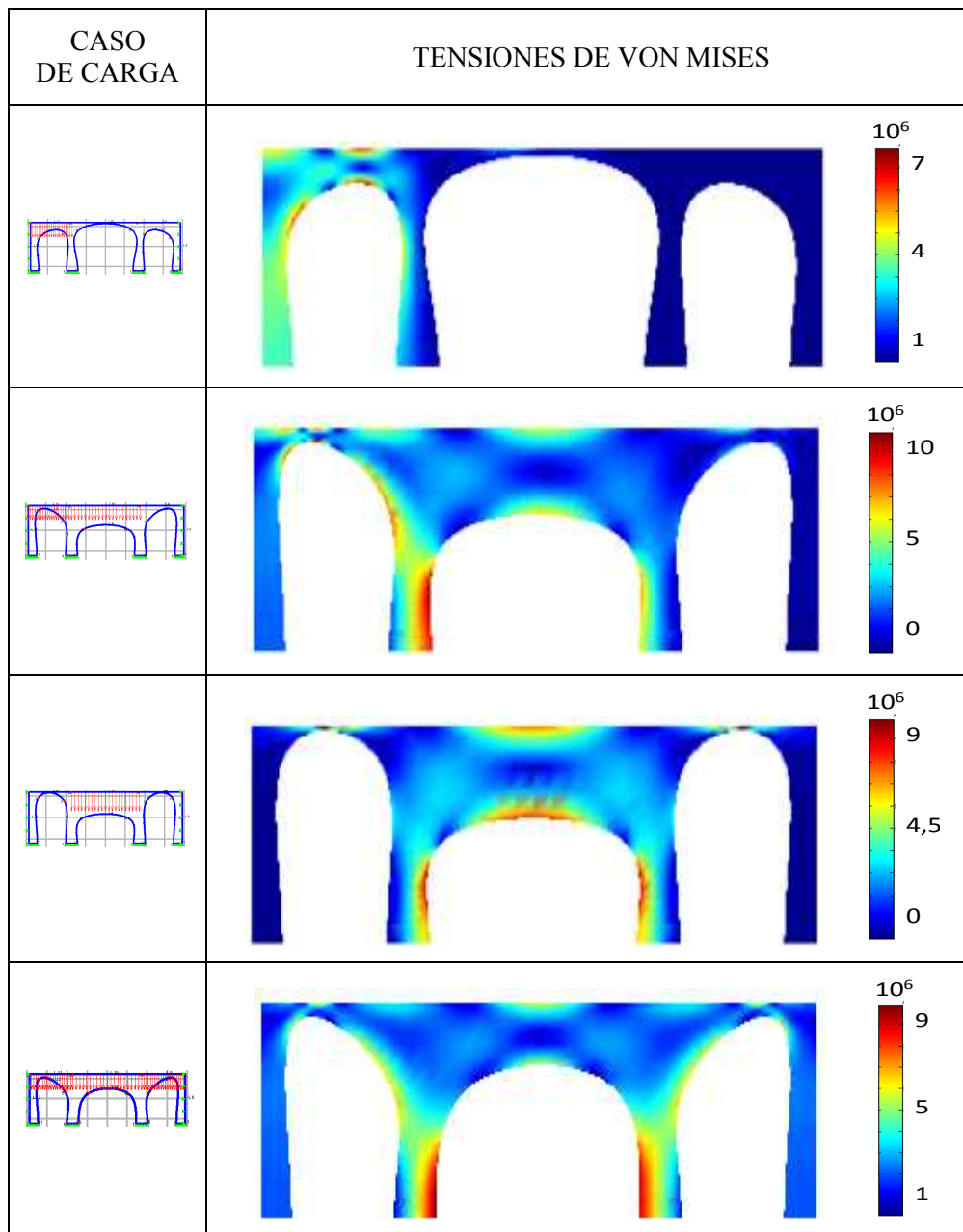


Fig. 26 Forma óptima hallada para una parametrización de tres puntos y tensiones de Von Mises



Aunque, posiblemente la parametrización del spline mediante 4 puntos no sea la más adecuada puesto que no se obtiene una tensión elevada en los correspondientes bordes de los arcos, se puede observar que el resultado obtenido es mucho más razonable que en el caso anterior.

6.4 ACTUALIZACIÓN SECUENCIAL

En este apartado se ha realizado una optimización secuencial del cilindro, considerando una restricción que hará que las variables 2 y 3 sean iguales.

Antes de comenzar a explicar el procedimiento que se ha seguido es importante tener en cuenta los diferentes tipos de tratamiento de la restricción que se pueden utilizar:

- Asegurando que una solución factible es siempre mejor que una solución no viable y teniendo en cuenta la intensidad de la violación de restricción. Esto se hace usando "fuzzy" limitaciones para las que se especifica una tolerancia de violación a nivel de definición del proyecto. Cuando la tolerancia utilizada es 0, el método recibe el nombre de método barrera, puesto que se rechazan todas las configuraciones no válidas.
- Mediante la adición de un objetivo para el problema igual al número de restricciones violadas. Esta opción sólo tiene sentido para problemas sobre-restringidos en los que una solución factible no existe y hay una serie de limitaciones de importancia relativamente parecida.

En este apartado se ha utilizado el primer método de tratamiento de restricciones utilizando una penalización del 3% de la tensión máxima admisible, es decir, se permitirá tener en cuenta los individuos que cuya tensión máxima quede comprendida entre 2000000 y los 2060000 Pa.

A continuación se procederá a explicar como se ha realizado la optimización secuencial. El esquema del procedimiento seguido es:



1. Se deja solamente que una de las variables actúe como tal, considerando el resto como si fuesen constantes y dándoles un valor inicial aproximado.
2. Se fija un número de iteraciones no demasiado elevado y se resuelve dicha optimización.
3. Una vez resuelta se fija el valor obtenido y se toma la siguiente coordenada como variable y así sucesivamente.
4. Cuando el mejor valor admisible obtenido se haya quedado muy próximo al anterior, pero el mejor valor de los penalizados quede a una distancia superior a dos unidades, se tomará este último como nueva constante a fin de aumentar la velocidad de convergencia.
5. Repetir al proceso hasta que converja la solución.

Este procedimiento es más rápido puesto que se consigue una convergencia de la solución en aproximadamente unas 1500-2000 iteraciones totales frente a la obtenida utilizando un procedimiento convencional. El único inconveniente es que se trata de un procedimiento que se ha de realizar manualmente, ya que durante el mismo se han de ir tomando decisiones y modificando los parámetros a optimizar para guiarlo en función de lo que decida el usuario en cada momento.

6.5 CONCLUSIONES

La definición paramétrica de la geometría es fundamental, puesto que, la forma óptima de la pieza que se puede obtener depende de ella en gran medida. Según el número de puntos utilizado para definir el contorno de la pieza, como se ha demostrado en el caso de la viga en el apartado 4.2 o el puente de la 6.3, se puede llegar a representar o no la forma real óptima.

Por otro lado, según se impongan las restricciones y la forma en que se limite por lo tanto, el espacio de diseño, también influirá de forma importante en la velocidad de convergencia de la solución, como se puede observar en el caso del cilindro ya que, cuando se impone la condición de que las coordenadas x e y del punto 2 la velocidad de convergencia aumenta.



En la mayoría de los casos esta optimización ha convergido antes de alcanzar los 7500 individuos, mientras que si se permite a las dos coordenadas que puedan ser diferentes y ambas generadas aleatoriamente la marca de los 7500 individuos es rebasada de modo regular, puesto que al alcanzar esta cifra aún no ha convergido la solución.

La optimización secuencial es un buen método para acelerar la convergencia de la solución. Ahora bien, presenta el inconveniente de que las actualizaciones con modeFRONTIER, como ya se ha comentado antes, se han de realizar de forma manual y de que no resulta fácil guiar la optimización salvo que se tenga una idea previa y/o aproximada de la solución.



7 CONTROL DE VELOCIDAD DE CONVERGENCIA

7.1 INTRODUCCIÓN

Se ha demostrado que el error de discretización influye significativamente sobre la mejor solución encontrada cuando el proceso de optimización se encuentra en una etapa avanzada, sin embargo, durante la etapa inicial se obtienen unos valores de la función objetivo parecidos indistintamente del nivel de error fijado. Una forma de aumentar la eficiencia del algoritmo es combinar un alto nivel de error en la etapa inicial con un nivel bajo de error en etapas finales, de forma que se reduzca el número de análisis, ya que se reduce el número de remallados y el número de resoluciones del mismo problema mediante el MEF.

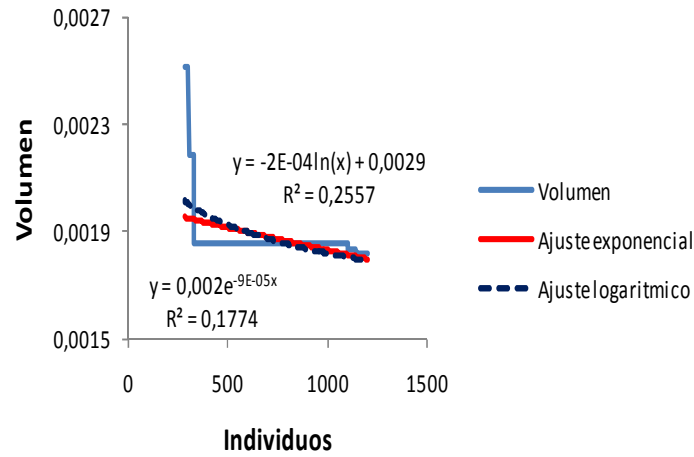
Es aconsejable no realizar cambios bruscos del error de discretización ya que podrían admitirse finalmente soluciones de muy poca sección, que han resultado factibles por haber sido evaluadas con un alto nivel de error pero que no serían factibles si se especificase la realización de su análisis con mayor precisión.

7.2 AJUSTE DE LA CURVA DE CONVERGENCIA

Se ha tratado de encontrar curvas que se ajusten lo mejor posible a la curva de convergencia, para posteriormente poder definir una estrategia que guíe el cálculo del error de discretización durante la convergencia, de forma que este último vaya disminuyendo a medida que avanza la optimización con el consecuente ahorro computacional que eso supone.

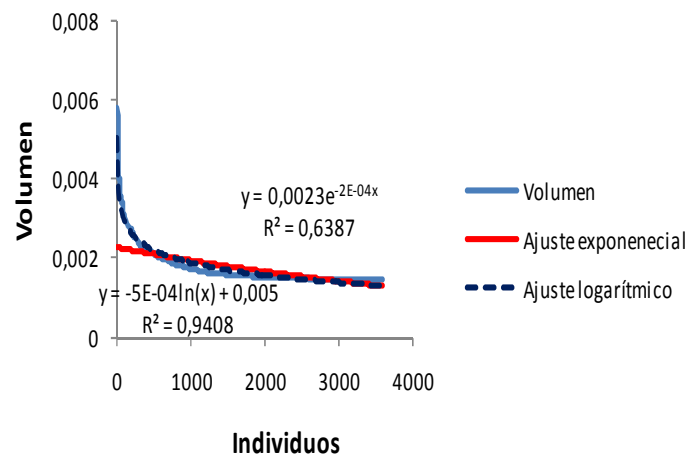
Como una primera aproximación al problema se ha tratado de ajustar a la curva de convergencia mediante el método de mínimos cuadrados curvas de tipo exponencial y logarítmica.

Ajuste convergencia 1



(a)

Ajuste convergencia 2



(b)

Fig. 27 Evolución de la curva de convergencia de un análisis de la viga y ajuste de una curva logarítmica y una exponencial



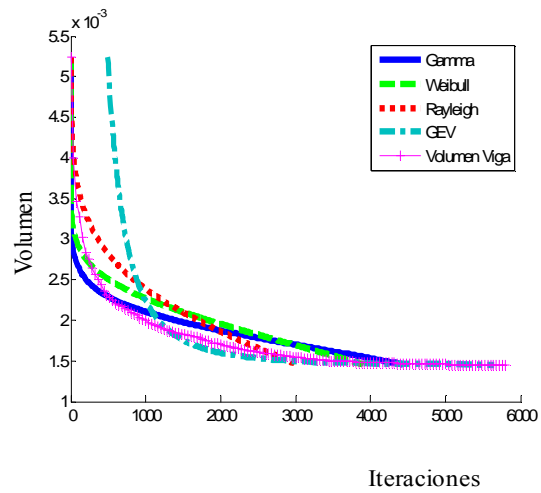
A la vista de las gráficas de las curvas de convergencia, que se presentan con sus respectivos ajustes en la Fig. 26, se deduce que la aproximación de la curva logarítmica es mejor que la de la exponencial. Sin embargo, el ajuste de la curva logarítmica, aunque es bueno en algunos casos (Fig. 26b), cuando la curva de convergencia desciende de forma muy brusca, el ajuste de la logarítmica no es capaz de imitar dicho comportamiento por lo que el ajuste no es válido (Fig.26a), ni siquiera para valores altos de la iteración lejos del escalón.

Por tanto, se ha tratado de buscar otro tipo de curvas que consigan una mejor representación de este tipo de comportamientos de las curvas de convergencia como se explicará a continuación en los dos siguientes subapartados (7.2.1 y 7.2.2).

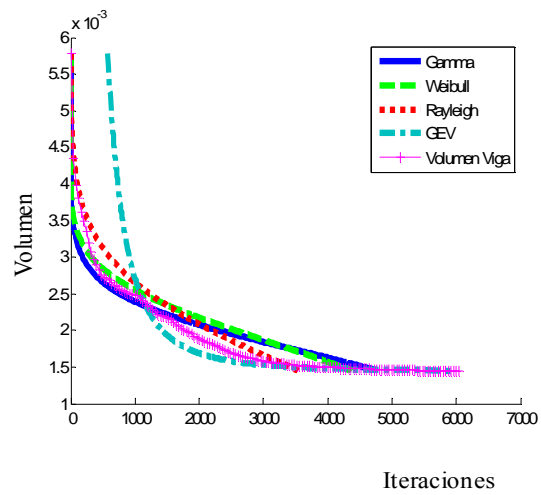
7.2.1 Ajustes de la curva de convergencia para la viga y el cilindro

Para realizar un ajuste se han tomado varias curvas de convergencia generadas a partir de diversos análisis realizados con los problemas de referencia. De lo cual, se pudo observar que la forma de la curva en ambos casos era muy similar, por tanto, se tratará de buscar una curva cuya forma se ajuste bien en ambos casos y que se pueda extrapolar también a otros problemas.

Puesto que, el proceso de optimización en el fondo es un proceso estocástico su forma resulta similar a las curvas estadísticas de distribución de probabilidades. Tras intentar ajustar diferentes curvas de este tipo a la curva de convergencia tales como la Distribución Gamma, la de Weibull, la de Rayleigh y la del Valor Extremo Generalizado (GEV), se ha llegado a la conclusión de que la más se aproxima a ella es esta última. Aunque en el primer tramo el error que se comete en la aproximación es más alto, en la zona final donde realmente es conveniente obtener menor error ya que nos estamos acercando realmente a la solución del problema, se aproxima mucho a la curva real de volumen, tanto de la viga como del cilindro, mientras que el resto de curvas quedan claramente por encima

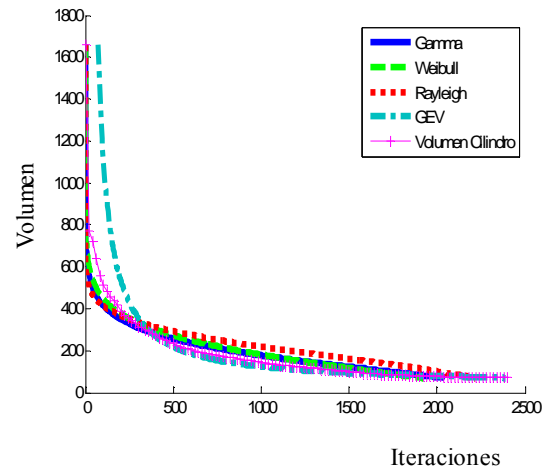


Grupo 1 de curvas de ajuste de convergencia de la viga

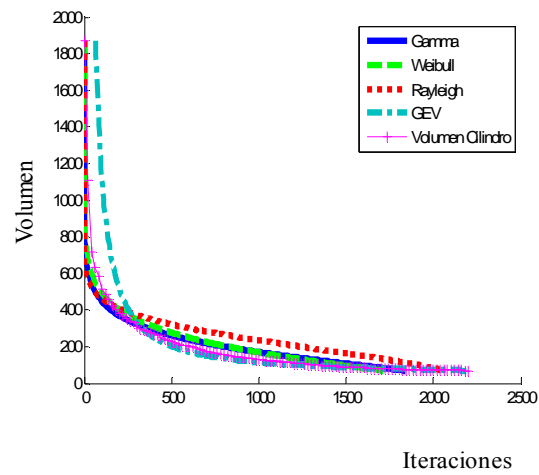


Grupo 2 de curvas de ajuste de convergencia de la viga

Fig. 28 Curvas de convergencia obtenidas para dos ajustes de del problema de la viga y ajustes realizados con las curvas estadísticas de probabilidad



Grupo 1 de curvas de ajuste de convergencia del cilindro



Grupo 2 de curvas de ajuste de convergencia del cilindro

Fig. 29 Curvas de convergencia obtenidas para dos ajustes de del problema del cilindro y ajustes realizados con las curvas estadísticas de probabilidad

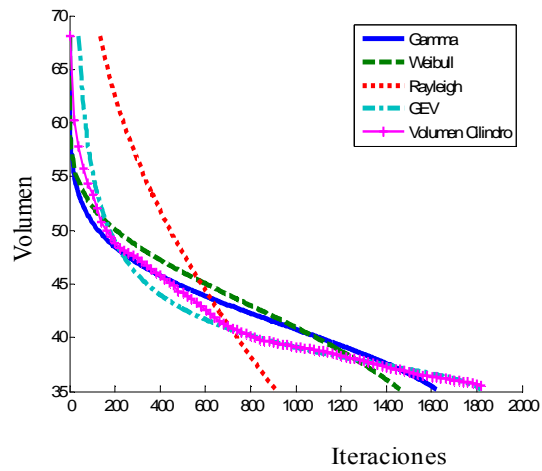


7.2.2 Ajustes de la curva de convergencia para el puente

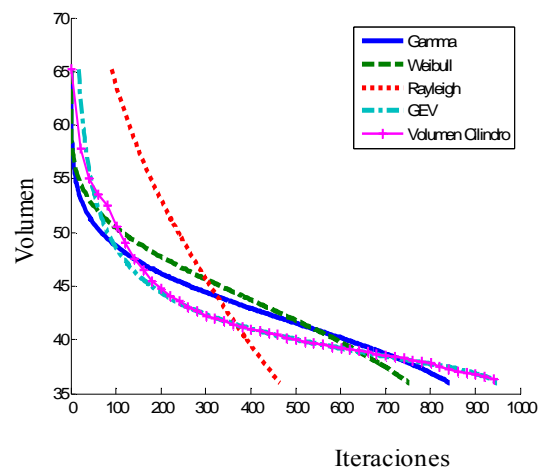
Se ha realizado el mismo ajuste de la curva de convergencia que en el subapartado 7.2.1 para las curvas obtenidas en los análisis de los casos de carga 3 y 4 del problema del puente cuando la FO utilizada es el volumen.

En la siguiente figura se puede comprobar que, de todas las distribuciones utilizadas, la que mejor se ajusta a la curva de convergencia es la curva obtenida mediante el GEV.

Si comparamos los resultados obtenidos para el puente con los obtenidos para los problemas de referencia se puede determinar que la aproximación es mucho mejor en este caso, ya que el ajuste es bastante bueno incluso en el primer tramo de la curva de convergencia donde la pendiente de la curva es más pronunciada y era más difícil conseguir un buen ajuste.



Grupo de curvas de ajuste de convergencia del puente
(Caso de carga 3)



Grupo de curvas de ajuste de convergencia del puente
(Caso de carga 4)

Fig. 30 Curvas de convergencia obtenidas para dos ajustes del problema del puente y ajustes realizados con las curvas estadísticas de probabilidad

7.3 ESTRATEGIA DE ESTIMACIÓN DE LA CURVA DE CONVERGENCIA

Para estimar la curva, que más se aproxima en cada momento a los valores que se han ido obteniendo en tiempo real durante el proceso de optimización y poder guiar la estrategia de error de discretización exigido en el análisis de elementos finitos, se ha utilizado una función que estima los parámetros que definen la curva de GEV que mejor se ajusta a la curva de convergencia en cada momento conforme va avanzando el proceso de optimización.

La función de densidad de probabilidad para el Valor Extremo Generalizado con parámetro de posición μ , parámetro de escalado σ , y parámetro de forma $k \neq 0$ es:

$$y = f(x|k, \mu, \sigma) = \left(\frac{1}{\sigma}\right) \cdot \exp\left[-\left(1+k \cdot \frac{(x-\mu)}{\sigma}\right)^{-\frac{1}{k}}\right] \cdot \left(1+k \cdot \frac{(x-\mu)}{\sigma}\right)^{-\frac{1}{k}}, \text{ donde}$$
$$1+k \cdot \frac{(x-\mu)}{\sigma} > 0$$

7.4 ESTRATEGIAS DE RESTRICCIÓN DE ERROR DE DISCRETIZACIÓN EN FUNCIÓN DE LA CURVA DE CONVERGENCIA DE OPTIMIZACIÓN

Para realizar la estimación del error, que se está cometiendo al aproximar la curva de convergencia a una curva de GEV, se calculará el error relativo en función del volumen mínimo obtenido hasta el momento por el programa de optimización y del estimado a partir de la aproximación a la curva de convergencia.

$$Error_Relativo = abs\left(\frac{V_{\min_optimización} - V_{\min_estimado}}{V_{\min_optimización}}\right)$$

Para obtener el volumen mínimo estimado en cada momento según la curva de convergencia aproximada se tomará en primer lugar la función de probabilidad acumulada:

$$P(V) = 1 - e^{-\frac{1}{\sqrt[k]{1+k \cdot z}}}, \text{ donde } z = \frac{V - \mu}{\sigma}$$

Si a continuación se toma el límite cuando $V \rightarrow V_{min}$ se tiene que, $P(V_{min}) = 1$, luego:

$$e^{-\frac{1}{\sqrt[k]{1+k \cdot z}}} = 0 \Rightarrow \frac{1}{\sqrt[k]{1+k \cdot z}} \rightarrow \infty \Rightarrow \sqrt[k]{1+k \cdot z} \rightarrow 0 \Rightarrow k \cdot z = -1$$
$$k \cdot \frac{V_{min} - \mu}{\sigma} = -1 \Rightarrow V_{min} = \frac{-\sigma}{k} + \mu$$

Por tanto, con este volumen mínimo ya se puede estimar el error relativo durante el proceso de optimización.

Además, se han probado diversas estrategias para disminuir el error de discretización a medida que se van estimando volúmenes, durante el proceso de optimización, y se va reduciendo también el error relativo a medida que la solución va convergiendo al volumen óptimo.

También cabe destacar previamente que, el primer ajuste de la curva de convergencia no se empieza a estimar hasta que se ha completado el cálculo de la primera generación (compuesta de 10 individuos). Dicha generación se calcula para cualquiera de las estrategias con un error del 20%, que se irá modificando en función de los resultados que se vayan obteniendo para cada estrategia.

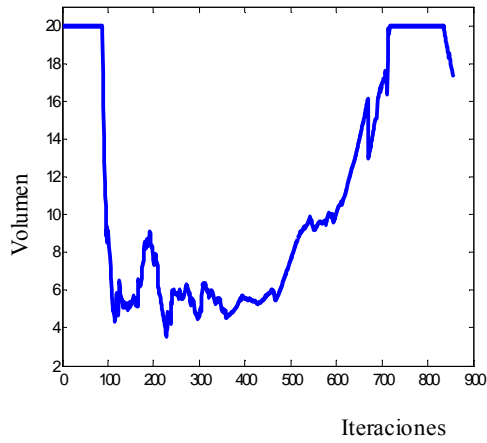
7.4.1 Estrategias utilizadas para el problema del cilindro

Para cada estrategia, de las que se explican a continuación, se han realizado diversos análisis del problema del cilindro (con diferentes poblaciones iniciales) a fin de comprobar la robustez de cada una de las estrategias.



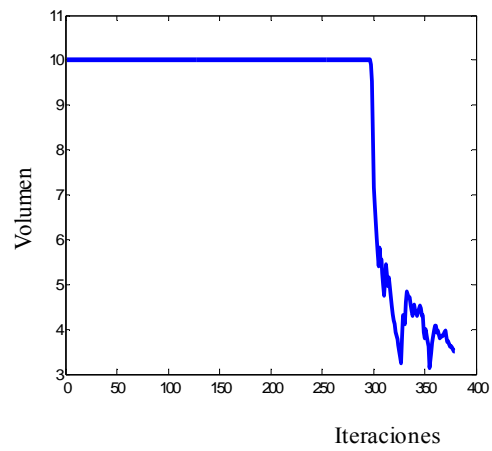
- **Estrategia 1:** Se considera un error de discretización del 20% mientras el error relativo de la curva ajustada de velocidad de convergencia sea superior al 20%, a continuación se disminuye el error de discretización a la par que lo hace el relativo hasta que este último alcanza un valor igual o inferior al 3%, punto a partir del cual se tomará un error de discretización del 1%.
- **Estrategia 2:** Se considera un error de discretización del 10% hasta que se alcanza el 10% de error relativo, disminuyéndolo de aquí en adelante a un valor constante del 1%.
- **Estrategia 3:** Mientras el error relativo sea mayor o igual al 100% se considera un error de discretización del 10%, y cuando el error relativo se halle entre el 100% y el 50% se toma un error de discretización del 5%, mientras que, por debajo del 50% se tomará uno del 1%.

A continuación se muestran de forma gráfica los resultados obtenidos para cada una de las estrategias:



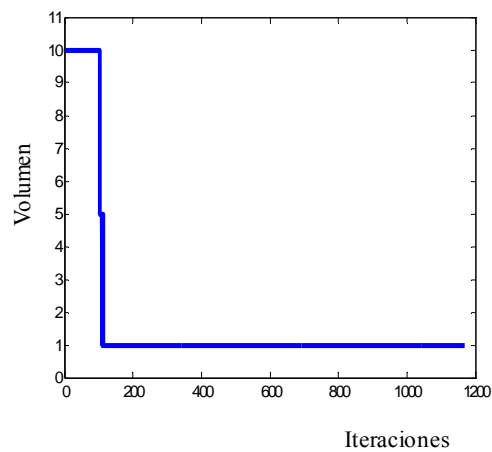
Evolución del error de discretización en la estrategia 1

(a)



Evolución del error de discretización en la estrategia 2

(b)



Evolución del error de discretización en la estrategia 3

(c)

Fig. 31 Evolución del error y el volumen según las diferentes estrategias de ajuste del error

En la siguiente tabla se recogen algunos de los resultados numéricos correspondientes a las gráficas anteriores:

| | Error de discretización impuesto según la estrategia | Volumen mínimo obtenido en el análisis con error de discretización exigido | Volumen mínimo estimado con la curva ajustada |
|---------------------|---|---|--|
| Estrategia 1 | 17,43 % | 65,58 | 54,14 |
| Estrategia 2 | 7,17 % | 66,01 | 61,28 |
| Estrategia 3 | 1,00 % | 68,71 | 61,49 |

Tabla 24 Resultados numéricos obtenidos según la estrategia de estimación del error utilizada

Para que la estrategia definida funcione correctamente se tiene que exigir un nivel de error de discretización bajo cuando el algoritmo encuentre el valor de volumen óptimo. Si la estrategia exige un nivel error de discretización alto para el individuo óptimo, éste se calcula mal y el volumen mínimo en este caso se subestima. En el caso contrario, si se exige un nivel de error de discretización muy bajo mucho antes de encontrar el individuo óptimo, se emplea un esfuerzo computacional demasiado elevado innecesariamente.

Como se puede observar en la figura 25 (a), si se utiliza la estrategia 1, el error de discretización impuesto por dicha estrategia crece cuando el número de iteraciones analizadas se hace elevado. Como consecuencia se realizan análisis en etapas avanzadas del proceso de optimización con niveles de error de discretización elevados y se obtiene un volumen mínimo demasiado bajo. Uno de los motivos que produce el efecto descrito es que para número de iteraciones bajo, el ajuste de la curva de velocidad de convergencia es malo (Figura 24) y la estimación del error de convergencia en optimización también.

Con la estrategia 2 se mejora algo la situación puesto que al final del primer tramo de error relativo del 10 % la curva de velocidad de convergencia



estimada se aproxima más a la real y el error de discretización impuesto es más realista. No obstante, parece que se empieza a disminuir el error demasiado tarde y no se impone un error de discretización suficientemente bajo cuando se encuentra el valor mínimo, lo que redundará en una diferencia entre el óptimo encontrado y el teórico aún significativa.

Con la estrategia 3 se empieza a reducir el error de discretización más rápidamente y por ello se subestima menos el volumen óptimo obtenido, a costa de un coste computacional más elevado.

7.4.2 Estrategia de optimización para el puente

En el problema del puente se ha seguido una estrategia muy similar a la número 3 de las utilizadas para el cilindro. La única diferencia radica en que al resolver este problema no resulta posible alcanzar errores de discretización del 1% con los niveles de refinamiento disponibles en el programa de elementos finitos utilizado. Por lo tanto, para valores menores del 50% de error relativo se tomará también un error de discretización del 5%.

Por otra parte, una vez realizados varios análisis del puente y a la vista de los resultados obtenidos en las gráficas de la Fig. 24 se puede observar que el ajuste obtenido es muy bueno, por lo que la estrategia utilizada en este caso no sería la más adecuada, ya que con la exclusión de los 10 primeros individuos que se han calculado con un error del 20%, el resto se ha calculado con un error de discretización del 5%. Como consecuencia apenas se ha conseguido disminuir el coste computacional del análisis.

7.5 CONCLUSIONES

Se han estudiado varias alternativas para el ajuste de la curva de convergencia a un modelo teórico. Los ajustes basados en distribuciones estadísticas han proporcionado mejores resultados, posiblemente porque el proceso real es estocástico pero donde los parámetros de control del algoritmo evolutivo establecen la probabilidad de mejora de la solución. Finalmente, se ha optado por utilizar una distribución del tipo Valor Extremo Generalizado (GEV), que es de



tipo asimétrico y se aproxima mejor a las curvas de convergencia de los ejemplos estudiados.

A partir de dicho ajuste se ha definido una función error de convergencia del proceso de optimización y se ha deducido como puede ser obtenido a partir de la estimación del volumen mínimo para cada iteración del proceso de optimización.

En cuanto a las estrategias de reducción del error de discretización exigido para cada iteración del proceso de optimización cuando el calculador se basa en el método de los elementos finitos, se han probado tres estrategias para el caso del cilindro. La mejor de ellas es la tercera que reduce más rápidamente el error de discretización. Sin embargo, el coste computacional asociado a esta estrategia es el más elevado, aunque menor que el que sería necesario imponiendo un nivel del error constante del 1 %.

En cambio, para el problema del puente esta estrategia no resulta adecuada puesto que el ajuste que se ha obtenido es mucho mejor. Por lo que, posiblemente una estrategia en la que se defina una estrategia en la que el error de discretización dependa del error relativo estimado a partir del ajuste, resultaría más efectiva y se reduciría de forma importante el coste computacional.

Será necesario un estudio más profundo para establecer conclusiones de más alcance.



8 RESULTADOS OBTENIDOS, CONCLUSIONES Y TRABAJOS FUTUROS

En este apartado se recogen las principales conclusiones extraídas de los apartados anteriores y se enumeran posibles desarrollos futuros que continúen la línea de investigación en la que se enmarca la tesina aquí presentada.

8.1 CONCLUSIONES

Es interesante realizar una recapitulación de cuales han sido los factores más importantes que se ha encontrado en el proceso seguido.

- Resulta imposible encontrar unos parámetros de control que funcionen bien para todo tipo de problema (Teoremas No Free Lunch), por lo que se deberá buscar los más adecuados en cada caso, teniendo en cuenta que dependiendo de cuál sea la función objetivo dichos parámetros pueden variar incluso dentro de un mismo problema. A pesar de ello para los problemas estudiados de optimización de forma sí se pueden seleccionar unos parámetros de control que funcionen razonablemente en varios problemas y que son muy próximos a los que tiene por defecto el programa modeFRONTIER para el algoritmo Moga-II utilizado.
- La función objetivo determina en gran medida la evolución de la optimización ya que, en función de la que se ha escogido, la velocidad de convergencia puede verse incrementada y/o verse estabilizada. En concreto, con las funciones objetivo que sean utilizado, se puede afirmar que con la función volumen se obtienen una solución razonable en la mayoría de problemas, mientras que con integral propuesta que involucra la tensión funciona muy bien pero no en todos los problemas. La utilización de ambas FO conjuntamente aunque no proporciona ninguna mejora de la velocidad de convergencia sí permite tener una estrategia muy estable y robusta (la dispersión en el número de iteraciones para obtener el óptimo se suele reducir).



- Es importante que la definición paramétrica de la geometría sea la adecuada para poder conseguir una correcta representación del contorno de la geometría y que permita generar una variedad de geometrías lo suficientemente amplia como para representar una buena aproximación de la solución óptima.
- La denominada actualización secuencial de variables permite aumentar de manera importante la velocidad de convergencia pero los parámetros de control del algoritmo tienen que estar especialmente seleccionados para el problema. En caso contrario la velocidad de convergencia puede ser incluso más baja que con las estrategias usuales. Además, modeFRONTIER no permite implementar este tipo de parametrización, por lo que tendría que realizarse de forma manual y el usuario tendría que estar constantemente pendiente del proceso, lo cual es una clara desventaja.
- Se ha tratado de ajustar una curva a la velocidad de convergencia pero a pesar de ser interesantes los resultados obtenidos, aún no es lo suficientemente precisa. Para los casos en que se utiliza como calculador el método de elementos finitos, parece que una estrategia de restricción del error de discretización que permita disminuirlo de forma razonable a medida que la optimización avanza, permitirá reducir el coste computacional aunque para ello habrá que mejorar previamente el ajuste de la velocidad de convergencia.

8.2 DESARROLLOS FUTUROS

Las áreas de trabajo que se proponen como continuación de la línea de investigación en la que se enmarca esta Tesina son las siguientes:

- El estudio se ha realizado con un programa de optimización de propósito general (modeFRONTIER) con lo cual algunos de los estudios realizados no se han podido implantar con facilidad. En ese sentido para continuar con el estudio sería necesario seguir desarrollando los algoritmos genéticos propios disponibles en el departamento.
- Comprobación de la robustez de la utilización conjunta de las dos funciones objetivo con un mayor número de problemas de



optimización de forma con el fin de obtener resultados más concluyentes.

- Respecto al estudio de la parametrización de una pieza hay que tener en cuenta tres aspectos fundamentales: deben permitir suficiente variación de la geometría que permita la obtención del óptimo buscado, que el número de parámetros se debe mantener lo más bajo posible para aumentar la velocidad de convergencia y es diferente describir el contorno de una pieza al de un agujero. Aunque se han obtenido algunas ideas sobre este tema con los estudios realizados se requieren estudios más profundos para definir una estrategia correcta de parametrización.
- Ampliación del estudio de ajuste de la curva de convergencia del problema de optimización, por ejemplo mediante la utilización de una combinación ponderada de varias distribuciones estadísticas. La obtención de un ajuste mejor de la curva de velocidad de convergencia, permitirá el desarrollo de nuevas estrategias de restricción de error de discretización en función de la curva de convergencia de optimización.



9 ANEXO I: PLANTEAMIENTO DE UN PROBLEMA EN MODEFRONTIER

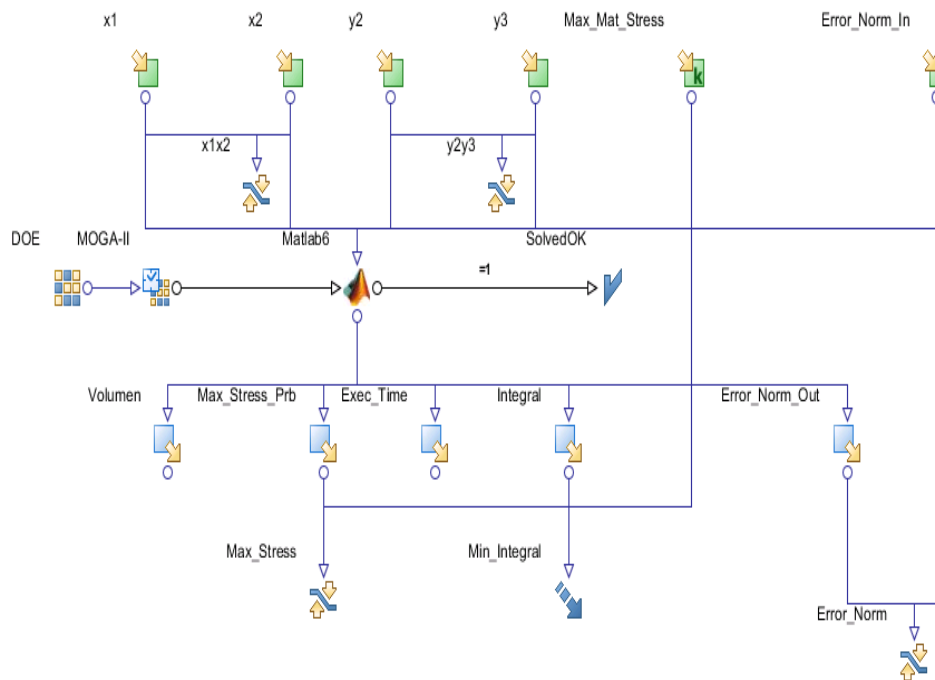


Fig. 32 Esquema del planteamiento del problema del cilindro con modeFRONTIER

A continuación se detallará como se ha realizado el planteamiento de uno de los problemas de referencia con el programa modeFRONTIER, en concreto del problema de un cilindro sometido a presión interna.

En primer lugar se han definido las variables de entrada, que en este caso son: x_1 , x_2 , y_2 e y_3 (en verde en la imagen). Para dichas variables es necesario definir algunos parámetros cómo son:

- **Nombre de la variable**



- **Tipo de variable:** Continua o discreta. Si se trata de variables discretas, como es el caso, se debe definir además un “paso” que determinará los valores del espacio de diseño que puede tomar la variable y que deberá adecuarse al espacio de diseño de la variable en cada problema.
- **Rango de variación:** Intervalo del que puede tomar valores la variable, que se definirá mediante unos límites inferior y superior determinados por el usuario.

En algunos problemas, como en este caso, también se han de definir algunos valores constantes, que variarán de un problema a otro y, que deben ser definidos también como parámetros de entrada por el usuario. En este problema dichos valores son el de tensión máxima admisible y el de error de discretización, que aparecen marcadas con la letra K (también en verde en la imagen).

Todas estas variables irán conectadas al programa que actuará como calculador, Matlab en este problema, y cuyo símbolo se puede apreciar en el centro de la imagen. Será necesario indicar la versión de Matlab con la que se va a trabajar y la ruta del archivo que actuará como unión entre el programa de cálculo de GFEM y el modeFRONTIER.

También se deberá conectar al calculador los llamados “nodos lógicos” que definirán los parámetros de la optimización propiamente dichos. Además se tienen que definir las variables de salida y otro conector lógico, que será el que nos indique si durante la ejecución del programa de cálculo no se ha producido ningún fallo.

Las variables de salida (en azul en la imagen) en este caso son: el volumen, la tensión máxima obtenida, el tiempo de ejecución, la integral en tensiones y error de discretización alcanzado tras el cálculo.

Es importante destacar que pueden existir restricciones (dos flechas con distinto sentido en la figura), tanto para las variables de entrada como para las de salida, aunque serán de distinta índole.

En primer lugar, para las variables de entrada las restricciones serán de tipo geométrico (una variable menor que otra p.e.) mientras que, para las de salida el objetivo es que no se superen los valores definidos como valores constantes de entrada, en este caso la tensión y el error máximos admisibles. Dichas restricciones se pueden introducir como fórmulas matemáticas.



De entre todas las variables de salida se debe fijar una como valor objetivo, en este problema será la integral en tensiones, y que en este caso se ha de minimizar (en la figura aparece indicado con una flecha hacia abajo).

En cuanto a los conectores lógicos, se debe declara una variable que indique que el programa de Matlab se ha ejecutado correctamente en Propiedades del Espacio de Trabajo de Matlab y en las de Final Lógico (\surd azul).

Por último, se han de indicar los parámetros propios de la optimización, es decir, mediante que algoritmo se van a obtener los individuos de la primera generación de forma que se repartan por todo el espacio de diseño, y cuantos individuos se han de generar. En este caso algoritmo utilizado es Random, que genera los individuos de forma aleatoria en función de una semilla de aleatoriedad, que puede hacer que su generación sea dependiente de la hora actual del reloj del computador cuando vale 0. Todo esto se detallará en el DOE mientras que, el algoritmo que regirá la optimización será el MOGA-II, como se puede observar en la figura. Los parámetros que gobiernan dicho algoritmo se encuentran explicados detalladamente en el apartado 3.4.2 del presente trabajo de investigación.





10 ANEXO II: INTERFAZ MODEFRONTIER-GFEM

La aplicación que permite la optimización de un componente se encuentra dividida en dos partes, al igual que los métodos de optimización. Por una parte, se encuentra el nivel superior, donde se lleva a cabo la generación y selección de soluciones individuales (programa de optimización). Por otra parte, se encuentra el nivel inferior, donde se evalúan las soluciones (programa de GFEM). Las dos partes del programa asociadas al nivel superior e inferior se han denominado *módulo de optimización* y *módulo de evaluación* respectivamente.

La comunicación entre ambos módulos se realiza a través de un script de Matlab, que será el encargado de transmitir toda la información necesaria del *módulo de optimización* a una macro (también generada en Matlab) que se encargará de gestionar la información recibida (datos sobre la geometría la malla y el material) en cada iteración y de llamar al programa de EF en modo *batch* para resolver el problema correspondiente.

El *gestor de evaluación* permanece en espera hasta que el programa de GFEM finaliza y le devuelve el control. Una vez ejecutada la macro, la interfaz lee un archivo de salida de fallos, llamado `Errores.txt` y la notificación de fallo del comando de llamada de Matlab, posteriormente sobrescribe el archivo de salida de fallos y le comunica al *módulo de optimización* si se ha detectado algún fallo durante toda la etapa de evaluación.

A continuación a modo de ejemplo se muestra el código empleado en el problema del cilindro.

No obstante, en caso necesario en la referencia [19] se puede encontrar más información sobre cómo se han de generar los ficheros.



SCRIPT MATLAB

```
Parameters(1)=x1;  
Parameters(2)=x2;  
Parameters(3)=x2;  
Parameters(4)=y3;  
  
TargedErrorNorm=Error_Norm_In;  
MaxMaterialStress=Max_Mat_Stress;  
  
Problema='GeometriaValid_100points';  
Problem_path='E:\Eva\Pipe2D';  
RutaSolver='C:\Documents and  
Settings\evsnor\Desktop\GFEM_v_3.00';  
PathVolume='E:\Eva\VolumenC';  
PathEVC='E:\Eva\EVC';  
  
if Individual>9 && exist('E:\Eva\EVC');  
    fid=fopen(PathEVC);  
    [Dat,count]=fscanf(fid,'%f',[4 inf]);  
    fclose(fid);  
    if isempty(Dat)==0  
        Longitud=length(Dat(1,:));  
        TargedErrorNorm=Dat(1,Longitud);  
    end  
end  
  
Data=Pipe_modeF(Parameters,TargedErrorNorm,  
MaxMaterialStress,Problema,Problem_path,RutaSolver,  
Individual);  
  
Volume=Data.Volume;  
Max_Stress_Prpb=Data.Stress;  
Error_Norm_Out=Data.ErrorNorm;  
Exec_Time=Data.Time;  
SolvedOK=Data.SolvedOK;  
Integral=Data.Integral;
```




SCRIPT MATLAB

```
if (Max_Stress_Prbc<=MaxMaterialStress) && (Max_Stress_Prbc~==1)
    fid=fopen(PathVolume,'a');
    fprintf(fid,'%f \n',Volume);
    fclose(fid);

    if Individual>=9;
        [Vmin,Vmax,Error]=AjusteC(PathVolume);
    end
end
```



MACRO MATLAB

```
function
Data=Pipe_modeF(Parameters,TargedErrorNorm,MaxMaterialStress,
Problema,Problem_path,Ruta,Individual)

global Geo Param FEMGui Elm BatchInfo Iter iMesh CalcMesh
GlobalIntersData GlobalMatKData MatK Prob

TimeIni=tic;

ModeBatch=1;

%=====
% Next part of the program must be modified for each example
%=====

%-----
% Mode Parameter
%-----

Param.ModeBatch=ModeBatch; % Param.ModeBatch=1 Allows to run
the program without GUI

%-----
% Geometry generation
%-----

% Load all problem variables only in first iteration
addpath(Problem_path)
load(Problema)

cd ../..
cd(Ruta)
```



MACRO MATLAB (*Continuación*)

```
% Initialization of the corresponding Matlab paths. It
mustn't be modified
%-----
Ini
Constants
%-----
%Modification of geometrical points

Geo.Pt.XYZ(1,1)=Parameters(1);
Geo.Pt.XYZ(1,2)=Parameters(2);
Geo.Pt.XYZ(2,2)=Parameters(3);
Geo.Pt.XYZ(2,3)=Parameters(4);

%Max Stress definition
Geo.Material.MaxStress=MaxMaterialStress;

%-----
%      Mesh
%-----
BatchInfo.HadaptiveMethod=1;
BatchInfo.AnalysisType='h_adaptive_mesh'; %h_adaptive_mesh;
TopOpt
BatchInfo.EType=2;
BatchInfo.IniMesh=2;
BatchInfo.IntsMesh=2;
BatchInfo.MaxLevelMesh=9;
BatchInfo.SquareMesh=1;
BatchInfo.UniformRefinement=0;
BatchInfo.RecovType=300;
BatchInfo.LoadCase2Solve=1; %Load cases that I need to
solve.
BatchInfo.TargErr=TargedErrorNorm; %Objetive Relative error
in Percentage
BatchInfo.ErrReduct=50;%30 %Max allowed error reduction (ex:
25=> reduce to 75% of current error)
```



MACRO MATLAB (*Continuación*)

```
BatchInfo.MxRefInc=3;           %Maximum refinement increment
BatchInfo.MxRefLevel=8;        %Maximum refinement level
BatchInfo.MxNIter=10;%5        %Maximum number of
iterations
BatchInfo.GeometricalRefinementArea=0;
BatchInfo.GeometricalRefinementCurvature=1;
BatchInfo.ExactPressSolution=0;
BatchInfo.AnalysisMode2D=PlaneStrain; %PlaneStrain=0;
PlaneStress=1;

BatchInfo.LoadCase2Solve=1;

%-----
% Definition of global variables for speed
%-----
Nel=0;
for iL=0:BatchInfo.MaxLevelMesh
    Nel=Nel+2^(2*iL);
end
GlobalIntersData.Data=[];
GlobalIntersData.DataF=[];
GlobalIntersData.Elements=sparse(Nel,length(Geo.Cv));
GlobalIntersData.ElementsF=sparse(Nel,length(Geo.Cv));
GlobalIntersData.NumStored=0;

GlobalMatKData.Elements=sparse(Nel,1);
GlobalMatKData.Data=[];

%-----
% Contour Parameter
%-----
Param.Contour=1;
%Param.Contour=1 Allows to calculate only at elements
belonging to contour.
%Param.Contour=0 Otherwise
```



MACRO MATLAB (*Continuación*)

```
%-----  
% Path and File  
%-----  
Geo.Path=Problem_path;  
Geo.File=Problema;  
  
%-----  
% GFEM PROGRAM  
%-----  
Constants; % Here are included the codes corresponding to  
each magnitude  
  
Errores.Fallo=0;  
Errores.Results=0;  
Errores.InvalidStress=0;  
  
ReCalcAll % This function creates the new geometry  
% BatchInfo.MeshCoords=MeshLimAuto; % Is is calculated the  
mesh limits automaticaly  
BatchInfo.MeshCoords=[-1 51;-1 51];  
if BatchInfo.SquareMesh  
    dummymin=min(BatchInfo.MeshCoords(:,1));  
    dummymax=max(BatchInfo.MeshCoords(:,2));  
    BatchInfo.MeshCoords=[dummymin dummymax;dummymin  
dummymax];  
end  
  
Source=2; % The program will be executed without Gui AND  
FROM OPTIMIZER
```



MACRO MATLAB (*Continuación*)

```
try
    MainGE(Source)

    Data.Volume=Iter(iMesh).Volume;

    if Param.Contour
        ActiveEls=CalcMesh(iMesh).OnContour';
    else
        ActiveEls=CalcMesh(iMesh).Material';
    end

    %Check if the shape is valid in stresses
    iLoadCase=BatchInfo.LoadCase2Solve;
    Calc(iMesh,iLoadCase)

    InvalidElemsStress=zeros(max(ActiveEls),1);
    for iElem=ActiveEls

    Index=Iter(iMesh).ErrorAnalysis(iLoadCase).ContourStresses.
    Index(iElem);

    Length=Iter(iMesh).ErrorAnalysis(iLoadCase).ContourStresses.
    Length_N(Index);

    ElmStress=reshape(Iter(iMesh).ErrorAnalysis(iLoadCase).
    ContourStresses.Recovered_N(1:Length,Index),4,[]);
    A=reshape(Iter(iMesh).ErrorAnalysis(iLoadCase).
    ContourStresses.Recovered_N(1:Length,Index),4,[])
        InvalidElemsStress(iElem)=max(ElmStress(4,:));
    end

    %Max Stress
    Data.Stress=max(InvalidElemsStress);
```



MACRO MATLAB (*Continuación*)

```
Integral=0;
    ActiveEls=CalcMesh(iMesh).Material';
    for iElem=ActiveEls

StructElem=CalcMesh(iMesh).ActiveElements(iElem);
        KNum=Elm.KNum(StructElem);
        KObj=MatK(KNum);

        if ~KObj.IntegrType

Index=Iter(iMesh).ErrorAnalysis(iLoadCase).InternalStresses.
Index(iElem);

ElmStress=reshape(Iter(iMesh).ErrorAnalysis(iLoadCase).
InternalStresses.Recovered_GP(:,Index),4,[]);
        VMStress=ElmStress(4,:);
        Diferencia=(MaxMaterialStress-VMStress).^2;

        ScaleFactor=2^(2*(Elm.Level(StructElem)-
KObj.OrigLev));

Integral=Integral+(Diferencia*KObj.AreaGauss/ScaleFactor);

        else

Index=Iter(iMesh).ErrorAnalysis(iLoadCase).ContourStresses.
Index(iElem);

Length=Iter(iMesh).ErrorAnalysis(iLoadCase).ContourStresses.
Length_GP(Index);
```



MACRO MATLAB (*Continuación*)

```
ElmStress=reshape(Iter(iMesh).ErrorAnalysis(iLoadCase).  
ContourStresses.Recovered_GP(1:Length, Index), 4, []);  
VMStress=ElmStress(4, :);  
Diferencia=(MaxMaterialStress-VMStress).^2;  
  
ScaleFactor=2^(2*(Elm.Level(StructElem)-  
KObj.OrigLev));  
  
Integral=Integral+(Diferencia*KObj.AreaGauss/ScaleFactor);  
  
end  
  
end  
Data.Integral=Integral*Prob.Thick  
  
% Extract the error estimated in Energy Norm  
Data.ErrorNorm=Iter(iMesh).ErrorAnalysis(iLoadCase).  
RelatError;  
  
FEMGui.PostOK=1;  
  
Data.SolvedOK=1;  
Data.Time=toc(TimeIni);  
  
catch ME  
  
Data.SolvedOK=0;  
Data.Volume=-1;  
Data.Stress=-1;  
Data.ErrorNorm=-1;  
Data.Integral=-1;  
  
errmsg=ME.message;
```




MACRO MATLAB (*Continuación*)

```
if (strcmp(errmsg,'Subscript indices must either be real
positive integers or logicals.'))...
    || (strcmp(errmsg,'Attempted to access
Elm.NSides(-1); index must be a positive integer or
logical.'))
    Errores.Fallo=1;

elseif (strcmp(errmsg,'Index exceeds matrix
dimensions'))
    Errores.Fallo=2;
elseif (strncmp(errmsg,'Too many curves intersect
element',29))
    Errores.Fallo=3;
elseif strcmp(errmsg,['...
    'WARNING: Mesh cannot be refined anymore. ',...
    'Objetive error NOT reached!!'])
    Errores.Fallo=5;
else
    Errores.Fallo=4;
end

Failure.Date=clock';
Failure.Individual=Individual;
Failure.Coord=Parameters;
Failure.errmsg=ME.message;
Failure.fallo=Errores.Fallo;
Failure.file=ME.stack(1).file;
Failure.line=ME.stack(1).line;
PathERRORES = [Problem_path '\output\ERRORES.TXT' ];
fid = fopen(PathERRORES,'a');
fprintf(fid,' %f %f %f %f %f %f ',Failure.Date);
fprintf(fid,'%f ',Failure.Individual);
fprintf(fid,'%f %f %f %f ',Failure.Coord);
fprintf(fid,'%s ',Failure.errmsg);
fprintf(fid,'%f ',Failure.fallo);
fprintf(fid,'%s ',Failure.file);
fprintf(fid,'%f \n',Failure.line);
fclose(fid);
```



MACRO MATLAB (*Continuación*)

```
Data.Time=toc(TimeIni);  
  
    return  
end
```



11 BIBLIOGRAFÍA

- [1] JorgHeitkotter and David Beasley. *The Hitch-Hiker's Guide to Evolutionary Programming*. <http://www.etsimo.uniovi.es/ftp/pub/EC/FAQ/www> 1993-1998. FAQ for comp.ai.genetic.
- [2] Kucerova, Leps and Zeman. *Applying genetic algorithms to several problems of engineering practice*. In internacional Congress on Evolutionary Methods for Design, Optimisation and Control with Applications to Industrial problems. CIMNE, Barcelona, Spain, 2003.
- [3] Eckart Zitzler, Marco Laumans, and Stefan Bleuber. *A tutorial on evolutionary multiobjective optimization*. In *Workshop on Multiple Objective Metaheuristics (MOMH)*. Springer Verlag Berlin, 2003. <ftp://tfp.tik.ee.ethz/pub/people/zitzler/ZLB2003a.pdf>.
- [4] Berard, Désidéri, Habbal, Janka, and Oulladji. *Experiments with hybridized genetic algorithms in aerodynamics*. In *Internacional Congress on Evolutionary Methods for Design, Optimisation and Control with Applications to Industrial Problems*. CIMNE, Barcelona, Spain, 2003.
- [5] William H.Press et al. *Numerical Recipes in C, The art of Scientific Computing* Cambridge University Press, 1997.
- [6] Khaled Rashee, Swarop Vattam, and xiao Ni. *Comparison of methods for using reduced models to speed up design optimization*, 2002.
- [7] Khaled Rasheed, Xiao Ni and Swarop Vattam. *Comparisson of methods for using reduced models for design optimization*. In proceedings of the Congress on Evolutionary Computation (CEC 2002),2002.
- [8] Gabriel Bugada, Juan José Ródenas, Eugenio Oñate. *An integration of a low cost adaptative remeshing strategy in the solution of structural shape optimization problems using evolutionary methods*. CAS 4108. Computers and Structures 2007.



- [9] Gabriel Bugada, Juan José Ródenas, José Albelda and Eugenio Oñate. *Influence of finite element discretization error over the convergente of structural shape optimization algorithms*. 2007 IEEE Congress on Evolutionary Computation. Singapore 2007
- [10] S.S.Rao. *The Finite Element Method in engineering*. ISBN 0-08-033419-9. Pergamon Press plc 1989.
- [11] Rainer Storn and Kenneth Price. *Diferential evolution – a simple and efficient adaptive scheme for global optimization over continuous space*. Technical Report TR-95-012, International Computer Science Institute, Berkeley, CA, USA, March 1995.
- [12] D. Nha Chu, Y.M. Xie, A. Gira, G.P. Steven. *Evolutionary structural optimization for problems with stiffness constraints*. Finite Elements in Analysis and Design 21 (1996) 239-251.
- [13] Peter W. Christensen, Anders Klarbing *An Introduction to Structural Optimization* ISBN 978-1-40208665-6 Springer Science + Business Media B.V. 2009
- [14] H. Kim, W.H. Mason, L.T. Watson and B. Grossman *A Study of the Statistical Description of Errors from Structural Optimization* Symposium on Multidisciplinary Analysis and Optimization 6-8 September 2000 / Long Beach, CA
- [15] Jouni Lampinen *Choosing a shape representation method for optimization of 2D shapes by genetic algorithms* Proceedings of the 3NWGA, Helsinki, Finland 20- 22 August 1997
- [16] Wolfgang A. Wall *, Moritz A. Frenzel, Christian Cyron *Isogeometric structural shape optimization* Chair for Computational Mechanics, Technische Universität München, Boltzmannstraße 15, 85747 Garching (b. München), Germany
- [17] Eckart Zitzler, Marco Laumanns, and Stefan Bleuler *A Tutorial on Evolutionary Multiobjective Optimization*
- [18] G. Allaire, F. de Gournay, F. Jouve, A.M. Toader *Shape Optimization by the level-set method*



-
- [19] Eva M. Sánchez Orgaz *P.F.C.: Optimización de componentes estructurales mediante algoritmo genético y software de elementos finitos para elasticidad lineal 2D con mallados independientes de la geometría* (2009)
- [20] Santos López Real *Tesina de Máster: Optimización de componentes mecánicos mediante algoritmo genético y software comercial de elementos finitos* (2008)