



Optimización Multiobjetivo de Estructuras,
Utilizando Algoritmos de Estimación de Distribución e Información
de Vecindades

TESIS

que para obtener el grado de:

*Maestro en Ciencias con Especialidad en Computación y
Matemáticas Industriales*

Presenta:

Sergio Ivvan Valdez Peña

Directores de Tesis:

**Salvador Botello Rionda y,
Arturo Hernández Aguirre**

Centro de Investigación en Matemáticas A.C.
Departamento de Ciencias de la Computación
A.P. 402, Guanajuato, Gto. 36000, México
e-mail: ivvan@cimat.mx
Agosto del 2005

Contenido

| | | |
|----------|---|-----------|
| 1 | Introducción | 1 |
| 1.1 | Introducción | 1 |
| 2 | Optimización de formas | 4 |
| 2.1 | Motivación | 4 |
| 2.2 | Objetivos generales de la optimización de formas | 5 |
| 2.3 | Metodología | 5 |
| 2.4 | Alcance del trabajo actual | 6 |
| 3 | Algoritmos de estimación de distribución | 8 |
| 3.1 | Algoritmos de Estimación de Distribución | 8 |
| 4 | Definición del problema | 10 |
| 4.1 | Definición del problema | 10 |
| 4.2 | Discretización y representación | 11 |
| 4.3 | Primera función objetivo: Mínimo peso | 11 |
| 4.4 | Segunda función objetivo: Mínimo desplazamiento | 12 |
| 4.5 | Primera restricción: Máximo esfuerzo Von Misses Permissible | 13 |
| 4.6 | Segunda Restricción: Número de Objetos | 14 |
| 4.7 | Tercera Restricción: Agujeros Pequeños | 14 |
| 5 | Algoritmo Propuesto | 15 |
| 5.1 | Descripción del algoritmo | 15 |
| 5.2 | Inicialización de la Probabilidad | 15 |
| 5.3 | Generación de la población | 16 |
| 5.4 | Selección y manejo de restricciones | 17 |
| 5.5 | Actualización de la probabilidad | 17 |
| 5.6 | Regularización de la probabilidad | 17 |
| 5.7 | Cálculo de la varianza | 18 |
| 6 | Mejoras del Algoritmo | 24 |
| 6.1 | Modificación en al selección | 24 |
| 6.2 | Varianza mínima para la regulariazción | 24 |
| 6.3 | Uso de archivo fijo | 24 |
| 6.4 | Paralelización del cómputo | 25 |
| 7 | Métricas | 28 |
| 7.1 | Métricas de desempeño | 28 |
| 7.1.1 | Métrica de convergencia | 28 |
| 7.1.2 | Métrica de diversidad | 29 |

| | | |
|----------|--|-----------|
| 8 | Experimentos | 31 |
| 8.1 | Experimento 1: Placa simplemente apoyada, 36 elementos | 31 |
| 8.1.1 | Experimento 1. Comparación con el Conjunto Verdadero de Pareto | 31 |
| 8.1.2 | Experimento 1. Métrica de convergencia | 32 |
| 8.1.3 | Experimento 1. Métrica de diversidad | 32 |
| 8.1.4 | Experimento 1: Conclusiones | 32 |
| 8.2 | Experimento 2: Placa simplemente apoyada, 144 elementos. | 34 |
| 8.2.1 | Experimento 2. Frente de referencia | 34 |
| 8.2.2 | Experimento 2. Métrica de convergencia | 34 |
| 8.2.3 | Experimento 2. Métrica de diversidad | 35 |
| 8.2.4 | Experimento 2. Conclusiones | 35 |
| 8.3 | Experimento 3: Diseño de un marco para bicicleta. | 36 |
| 8.3.1 | Experimento 3. Métrica de convergencia | 37 |
| 8.3.2 | Experimento 3. Conclusiones | 37 |
| 9 | Comentarios y conclusiones | 49 |

Lista de Figuras

| | | |
|------|---|----|
| 3.1 | Estructura básica de un algoritmo de estimación de distribución. | 9 |
| 4.1 | Espacio de búsqueda, estado de cargas, y estructura con mínimo peso. | 10 |
| 4.2 | Discretización y representación de una estructura. | 11 |
| 4.3 | Estructura con el mínimo peso en el frente. | 12 |
| 4.4 | Simulación del desplazamiento de una estructura en el punto de carga (línea roja) | 12 |
| 4.5 | Estado de esfuerzos en una estructura encontrados por el método del elemento finito | 13 |
| 4.6 | Configuración con 4 objetos | 14 |
| 4.7 | Configuración que presenta 3 agujeros pequeños | 14 |
| 5.1 | Descripción del algoritmo principal (MASO) | 16 |
| 5.2 | Descripción esquemática del algoritmo. | 19 |
| 5.3 | Pseudo-código del Algoritmo Multiobjetivo para Optimización de Formas | 20 |
| 5.4 | Pseudo-código para la inicialización de la probabilidad | 21 |
| 5.5 | Pseudo-código para la generación de la población | 22 |
| 5.6 | Esquema de actualización de la probabilidad | 22 |
| 5.7 | Probabilidad: (a) Antes de la regularización, (b) Después de la regularización. | 23 |
| 5.8 | Pseudo-código del algoritmo de cálculo de la varianza | 23 |
| 6.1 | Esquema de paralelización del algoritmo. | 26 |
| 6.2 | Estado de cargas para diseño del puente | 26 |
| 6.3 | Solución de mínimo peso, evaluación paralelizada, 632 elementos. | 27 |
| 6.4 | Solución de mínimo peso, evaluación paralelizada, 1184 elementos. | 27 |
| 7.1 | Valores sugeridos para $m()$ | 30 |
| 8.1 | Dimensiones y estado de carga para el experimento 1 | 32 |
| 8.2 | Propiedades del material del experimento 1 | 32 |
| 8.3 | Frente Verdadero con diamantes, y frentes de las 30 corridas con cruces. | 33 |
| 8.4 | Estructuras resultantes de una corrida típica. | 34 |
| 8.5 | Valores de $C(P^{(N)})$ (convergencia) para 30 corridas del experimento 1. | 35 |
| 8.6 | Comportamiento de la convergencia para el experimento 1. | 36 |
| 8.7 | Métrica de diversidad para el experimento 1. | 37 |
| 8.8 | Placa simplemente apoyada, discretizada en 144 elementos para el experimento 2. | 38 |
| 8.9 | Propiedades del material para el experimento 2. | 38 |
| 8.10 | Frente de referencia y no dominados en cada corrida, para experimento 2. | 39 |
| 8.11 | Estructuras para una corrida típica, experimento 2 | 40 |
| 8.12 | Valores de convergencia en 30 corridas, del experimento 2. | 41 |
| 8.13 | Comportamiento de la convergencia en una corrida típica, del experimento 2. | 42 |

| | | |
|------|---|----|
| 8.14 | Valores de la métrica de diversidad para 30 corridas del experimento 2. | 43 |
| 8.15 | Definición del problema para el experimento 3. | 43 |
| 8.16 | Propiedades del material para el experimento 3. | 44 |
| 8.17 | Frente de referencia (gris) y frentes de algunas corridas (rojo) del experimento 3. | 44 |
| 8.18 | Algunas estructuras de una corrida típica del experimento 3. | 45 |
| 8.19 | Métrica de convergencia de 20 corridas para el experimento 3. | 46 |
| 8.20 | Gráfica de convergencia en una corrida típica del experimento 3. | 47 |
| 8.21 | Gráfica de convergencia para 15000 generaciones del experimento 3. | 48 |

Resumen

En este trabajo se presentan un conjunto de técnicas para la optimización multiobjetivo de estructuras (diseño óptimo automatizado), basadas en algoritmos de estimación de distribución. Esta nueva propuesta de solución utiliza el criterio de dominancia de Pareto para el manejo de las restricciones y los objetivos. Los objetivos a optimizar son: el peso mínimo de la estructura y el desplazamiento en nodos especificados. Las restricciones de diseño de las estructuras son: a) no exceder un esfuerzo Von Mises máximo, b) no presentar elementos desconectados por los lados y c) no presentar agujeros pequeños (del tamaño de un elemento).

El algoritmo propuesto, utiliza información de las vecindades para mejorar la exploración hacia las soluciones más probables, y evitar mínimos locales. Se utiliza el método del elemento finito para evaluar las funciones objetivo y restricción (desplazamiento y esfuerzo).

Capítulo 1

Introducción

En el presente capítulo se describe de manera general el alcance y complejidad del problema de optimización de formas, se pone en antecedentes al lector de algunos de los trabajos realizados en este campo y de sus virtudes y deficiencias; y de forma breve se expone la técnica utilizada para mejorar las propuestas anteriores, solucionando algunos de los problemas que se presentan comúnmente en optimización de formas con algoritmos evolutivos.

1.1 Introducción

Por la complejidad computacional en el problema de optimización de formas se pueden distinguir tres clases, estas son: dimensiones, forma y topológica (ver Richards [16], y Sandgren[17]).

- *La optimización de dimensiones* se refiere a la determinación de dimensiones geométricas específicas para una clase de diseño preseleccionado, tal como el espesor de una placa, el tamaño de un armazón o el radio de un elemento con esfuerzo circular. Esta clase de problemas han estado bajo investigación durante décadas.
- *La optimización de formas* introduce variables de diseño adicionales las cuales permiten el movimiento de la frontera. Debido al incremento en la dificultad relativa con la optimización de tamaño, los cambios geométricos han sido limitados históricamente; sin embargo, esta ha ganado importancia en las industrias aeronáutica y automovilística, como en otras, produciendo mejoras en las turbinas, formas de las laminas del avión y piezas conectoras. La optimización de dimensiones es un subconjunto de la optimización de formas.
- *La optimización topológica* involucra modificaciones tanto topológicas como de forma y dimensiones. Las modificaciones topológicas tratan con ensambles de componentes. Los componentes en el ensamble pueden ser modificados y se pueden agregar componentes, quitar o mover componentes en el ensamble en el intento de generar un diseño mejorado. Relativamente poco trabajo se ha hecho en esta área, no obstante la importancia del concepto.

En el trabajo actual tratamos con el problema de optimización de formas en 2 dimensiones; aunque parece factible abordar el problema de 3 dimensiones utilizando la misma metodología que se ha utilizado en 2. Se ha utilizado una técnica relativamente nueva en el campo de los algoritmos evolutivos: los algoritmos de estimación de distribución (EDA's, ver Baluja [1], Muhlenbein[13], Pelikan [15] [14], Li [9]), los cuales intentan encontrar una distribución cuyas muestras generadas sean las mejores soluciones al problema.

Un algoritmo genético simple propuesto por Golberg realiza una selección proporcional a la aptitud de los soluciones, de acuerdo a la siguiente ecuación:

$$p^*(x, t) = p(x, t) \frac{f(x)}{f(t)} \quad (1.1)$$

En la ecuación 1.1 $x = (x_1, x_2, \dots, x_n)$ denota un vector de variables discretas (genotipo), $p(x, t)$ es la distribución de x en la generación t y $f(t) = \sum p(x, t)f(x)$ es el promedio de la aptitud de la población. Para simplicidad se pueden asumir variables binarias $x_i \in \{0, 1\}$. En principio se podrían generar puntos nuevos de acuerdo a:

$$p(x, t + 1) \approx p^*(x, t) \quad (1.2)$$

En general una probabilidad discreta está definida por 2^{n-1} parametros. Por lo tanto una implementación de esta manera sería demasiado costosa. Por lo tanto, una alternativa sería intentar aproximar la ecuación 1.2 sin un esfuerzo computacional exponencial. Para un caso simple se pueden utilizar probabilidades marginales, en el siguiente paso se podría intentar utilizar información de las dependencias más fuertes y construir un modelo gráfico de las probabilidades, cuyas muestras generen soluciones potenciales al problema de optimización. El algoritmo de estimación de distribución conceptual, propuesto por Muhlenbein en [12], es el siguiente:

- **PASO 0:** Asignar $t \leftarrow 1$. Generar $N \gg 0$ puntos aleatoriamente.
- **PASO 1:** Seleccionar $M \leq N$ puntos, siguiendo un método de selección. Estimar la distribución $p^*(x, t)$ del conjunto seleccionado.
- **PASO 2:** Generar N nuevos puntos de acuerdo a la distribución $p(x, t + 1) = p^*(x, t)$. Asignar $t \leftarrow t + 1$.
- **PASO 3:** Si el criterio de terminación no se ha alcanzado, ir al PASO 1.

La siguiente proposición muestra la convergencia del EDA (Estimation of distribution algorithm, ver Muhlenbein [12]) al óptimo.

Proposición Sea el máximo $f(x_m) = \max f(x)$ único. Si el método de selección cumple:

$$1 - p^*(x_m, t) \leq c(1 - p(x_m, t)) \quad 0 \leq c < 1 \quad (1.3)$$

Entonces el EDA converge al máximo:

$$\lim_{t \rightarrow \infty} p(x_m, t) = 1 \quad (1.4)$$

Basado en lo anterior el algoritmo propuesto evita el cálculo de un modelo gráfico, estimando probabilidades marginales y explotando información relativa al problema físico (vecindades) y al método de solución (elemento finito). Sin que tenga que desecharse por completo para trabajos futuros, se debe tomar en cuenta que si la dimensionalidad del problema es de cientos o tal vez miles de variables binarias, el cálculo de un modelo gráfico de probabilidad resultaría sumamente costoso.

En intentos anteriores (Kane [8], Deb [4], Chapman[2]) se han logrado (relativamente) buenos resultados utilizando algoritmos genéticos con representación binaria, para la optimización de formas. Estos intentos tanto monoobjetivo como multiobjetivo presentan varios problemas que evitan que la mejor forma para un estado de cargas dado sea encontrada. Algunos de los problemas generales mas representativos en estos intentos son:

- Estancamiento en mínimos locales debido a una exploración deficiente.
- Las formas resultantes no tienen todas las características deseables, debido a que el modelo de optimización no guía adecuadamente la búsqueda, es decir, muchas veces con pedir que se minimice el peso no es suficiente. Para encontrar la mejor estructura, los objetivos o restricciones que tienen que ver con esfuerzos o desplazamientos ayudan a dirigir mejor la exploración; también las funciones objetivo y restricción que involucran la forma misma, como el número de objetos o agujeros ayudan al algoritmo a mejorar su poder de decisión por las “mejores estructuras”, desde el punto de vista de ingeniería.

- Existe una dependencia de la malla que limita el tipo de formas que se pueden generar, este problema se hace más grave para mallas burdas.

El trabajo actual trata con 2 de los 3 problemas mencionados anteriormente, se utiliza una estrategia, que al detectar la convergencia prematura automáticamente incrementa la exploración modificando los vectores de probabilidad, lo que lleva a un incremento de la diversidad de la población . Además, se utiliza un modelo que intenta capturar mayor número de las características deseables de la estructura desde el punto de vista de ingeniería. El último problema relativo a la dependencia de la malla por el momento no se ha tratado, pero se espera en un futuro trabajar con estrategias que nos permitan minimizar esta dependencia, aunque se tienen resultados preliminares con mallas relativamente finas que ayudan a disminuir la dependencia de las soluciones con la malla .

Capítulo 2

Optimización de formas

En el presente capítulo se tratan las motivaciones e importancia de la investigación en optimización de formas, sobre las metodologías para resolver el problema, y las características deseables en las soluciones. El definir el tipo de soluciones deseadas, acotará el trabajo actual y guiará el trabajo futuro en el tema.

2.1 Motivación

La optimización de productos y procesos es parte inherente del trabajo industrial, generalmente se intenta disminuir el tiempo de fabricación, eliminar operaciones, u operar una célula o línea de producción con menos personal; este tipo de mejoras no necesariamente significan de forma directa una disminución del costo final del producto o un incremento en las ganancias. La mejora en el tiempo de fabricación no significa menos equipo, sino tal vez un menor tiempo de operación del mismo, tampoco necesariamente disminuye los gastos de espacio, iluminación o mantenimiento; lo mismo sucede al eliminar operaciones, tal vez se tenga un menor costo de la herramienta en esa operación, pero posiblemente sea un aumento en el costo de alguna otra operación del proceso, a su vez un operador o técnico menos en una celula de transformación no significa un ahorro en recursos humanos. Además de todos los factores mencionados, el posible ahorro en los procesos generalmente se diluye en el sistema de costos de manera que en el balance final la empresa no percibe esas mejoras en dinero. Por otro lado, cada centavo que se deja de gastar en materia prima impacta directamente en las utilidades y/o en el precio del producto, aquí radica la importancia económica de la optimización de formas, el mínimo material necesario para cumplir con las condiciones de servicio debe ser una característica intrínseca a una forma óptima.

La optimización de formas mantendrá o mejorará el desempeño del sistema o componente bajo las condiciones de servicio, las cuales generalmente tienen que ver con fuerzas aplicadas, condiciones térmicas, movimientos, corrosión o desgaste y envejecimiento, esto obviamente impacta en la calidad del producto, la competitividad de la industria y la preferencia del mercado.

En el diseño de un componente o sistema existen también otras características físicas que serán deseables para alguna parte interesada o el cliente usuario final, como restricciones de espacio, de fabricación o algún requerimiento aerodinámico, suavidad de los contornos (formas suaves de las líneas de contornos), etc. El cumplimiento de los requerimientos de una “forma óptima” o la mejora de sus características, de acuerdo a lo anterior, impacta de forma benéfica en la sociedad, la cual es finalmente quien mejora sus condiciones de vida, al acceder a productos más seguros y de mejor desempeño y que se adecuan mejor a sus necesidades con un costo menor.

Finalmente, cualquier desarrollo científico o tecnológico en alguna área, puede ser un punto de partida para desarrollos posteriores; y puede importar y/o exportar conocimientos de otras áreas.

2.2 Objetivos generales de la optimización de formas

El proceso de diseño actual se considera un proceso iterativo, dentro de este se pueden distinguir tres fases, no necesariamente secuenciales, ni realizadas de manera separada, estas son:

- Diseño conceptual.
- Análisis del diseño.
- Diseño de detalle.

En el diseño conceptual se realizan propuestas para resolver un problema o necesidad, esta es la parte creativa del proceso. El análisis del diseño se encarga de estudiar la factibilidad de las propuestas. Finalmente, el diseño de detalle adecua las propuestas para su fabricación, y se encarga de definir de manera explícita las características del producto en planos, formulas, etcetera; utilizando estándares o normas.

La optimización de formas es inherente al proceso de diseño, aunque actualmente sea llevada en mayor proporción por un elemento humano. Dadas ciertas condiciones de servicio, se desea encontrar la forma que mejor desempeño tenga bajo estas condiciones, con el menor costo. Las condiciones de servicio están definidas, por el tipo de interacción con el ambiente, como: fuerzas aplicadas, condiciones térmicas, movimientos, corrosión o desgaste y el envejecimiento. Las fuerzas aplicadas generalmente se consideran por medio de los esfuerzos o desplazamientos que sufre el material, y se diseña para que el componente o sistema no supere un límite de esfuerzo o un desplazamiento dado, asegurando la resistencia del material y el correcto funcionamiento del sistema. Las condiciones térmicas se consideran por medio del cálculo de los esfuerzos térmicos y los cambios que pudieran sufrir las propiedades del material. Los movimientos se refieren a las vibraciones y amortiguamientos deseados; así como a las fuerzas inducidas por estos. La corrosión y el desgaste se refiere a modificaciones en las propiedades o dimensiones del material por la fricción con el medio, otros componentes, o el contacto con sustancias y/o materiales que ataquen química o mecánicamente al componente. Aunque la palabra envejecimiento se utiliza también para cierto tipo de corrosión, en este caso es utilizada para describir cambios en las propiedades del material por el simple paso del tiempo, sin que el componente esté sometido a ningún tipo de contacto con alguna sustancia en especial, en materiales como el acero se realizan tratamientos de endurecimiento y mejora de la resistencia por procesos de envejecimiento, que es básicamente una sedimentación lenta en el material.

Pueden existir además otros requerimientos que sean específicos para un problema de diseño en particular, los cuales pudieran ser modelados, de forma tal que sean considerados en un proceso automatizado de optimización. Ejemplos de estos son: características de los contornos, propagación de la falla, restricciones de fabricación, etc.

Resumiendo los objetivos generales de la optimización de formas son:

- La minimización del costo del producto.
- La maximización del desempeño bajo las condiciones de servicio.
- El cumplimiento y mejora de las características definidas explícitamente.

Se puede observar que estos objetivos pueden estar en conflicto, lo que provocará que las soluciones tengan un compromiso en el cumplimiento de los mismos.

2.3 Metodología

En [16] Richards expone que una metodología de optimización de formas debe ser capaz de:

- Determinar un óptimo global.
- Exhibir un desempeño estable.

- Manejar problemas bi y tridimensionales.
- Desarrollarse sin la ayuda de información sensible.
- Competir en eficiencia con el estado del arte en las técnicas de optimización de formas.
- Debe ser independiente de la técnica de análisis utilizada para determinar los esfuerzos.

Algunas observaciones relativas a algunos puntos son las siguientes:

- Determinar un óptimo global. Lo más deseable sería que la metodología nos entregase como producto final una forma “lista para usarse”; desafortunadamente, es difícil definir un solo óptimo global para un conjunto de funciones que tienen un compromiso entre ellas, Para definir cuales son las mejores soluciones generalmente se utiliza el criterio de *dominancia de Pareto*, el cual no define un óptimo global, sino un conjunto donde todos los elementos son soluciones óptimas. Trabajar en una metodología que entregue un óptimo global, equivale entonces a definir en primera instancia un criterio de optimalidad que determine un solo óptimo global para m funciones, que tienen un compromiso entre si.
- Exhibir un desempeño estable. Para mostrar este desempeño nos podemos auxiliar de métricas que muestren el comportamiento y estabilidad del algoritmo, que ayuden a mostrar que bajo condiciones semejantes, se encuentran resultados semejantes.
- Desarrollarse sin información sensible. Este punto se refiere a los parámetros necesarios para obtener “buenas soluciones”, algunos de estos parámetros en muchos casos involucran: los coeficientes de una función que considera los múltiples objetivos a optimizar, el sembrar algunos agujeros en la estructura, o algunas partes de un material débil A, siendo otras partes de un material fuerte B, y definiendo el problema como modificar la frontera de A (ver [7]). La metodología debe ser capaz de encontrar soluciones óptimas, sin mostrar una dependencia “significativa” a los parámetros o condiciones iniciales proporcionadas por el usuario.
- Competir en eficiencia con el estado del arte en las técnicas de optimización de formas. Obviamente el costo computacional es importante en la solución de cualquier tipo de problemas, incrementándose en los que pudieran ofrecer soluciones industriales, como es el caso de la optimización de formas, donde por cada día que el producto se encuentra fuera del mercado, es un día de ventas perdido e incluso puede significar la obsolescencia del mismo; sin embargo, el estado de la tecnología actual y futuro próximo, vuelve (y/o esta volviendo) factible el resolver problemas que hasta hace algunos años resultaba prohibitivo resolver por el tiempo de cómputo, y el caso de optimización de formas es uno de estos problemas. Este estado de la tecnología indica que habría que considerar también la “eficacia” de la metodología, ya que no existe en el estado del arte actual una metodología que cumpla con los objetivos generales que se plantean en la sección 2.2, ni con los que plantea Richards en [16].

2.4 Alcance del trabajo actual

Después de mostrar los objetivos generales del problema de optimización de formas en la sección 2.2, y de describir las características deseables de una metodología en la sección 2.3, especificaremos el alcance actual del trabajo realizado, y hacia donde se puede expandir este alcance en un futuro cercano. El trabajo actual trata con:

- El problema de optimización de formas en dos dimensiones.
- Estados de multicargas.
- Optimiza el peso y el desplazamiento de algunos puntos en la estructura.

- Utiliza restricciones de esfuerzo y de la forma misma.
- Utiliza el criterio de Pareto lo que arroja un conjunto de soluciones.

De manera general, el trabajo realizado encuentra mejores soluciones que otros sobre el tema (ver [2, 4, 8, 3]), muestra un comportamiento estable (ver [5]), y se considera que puede ser expandido hacia un alcance más amplio en un futuro cercano.

La expansión de este alcance y donde se pueden encontrar áreas de mejora a corto plazo, podría implicar:

- Utilizar la metodología en tres dimensiones, de echo el algoritmo tal y como se presenta podría ser utilizado para tres dimensiones sin ningún cambio significativo, más que la evaluación de las soluciones.
- Se podría incluir el efecto de cargas térmicas y de vibraciones o frecuencias de la estructura.
- Se podría utilizar algún otro tipo de elemento, aunque seguiría siendo sensible al método de análisis que es el método del elemento finito.
- Se puede disminuir el espacio de soluciones tratando de encontrar solo un conjunto pequeño de soluciones “representativas” (ver la sección 6.3 y [19]).

Junto con los puntos anteriores se debe de considerar la mejora en costo computacional, la solución de problemas de mayor dimensionalidad (mayor número de elementos) (ver la sección 6.4) y la optimización de sistemas que tengan varios componentes ensamblados.

Capítulo 3

Algoritmos de estimación de distribución

En el presente capítulo se introducen los *algoritmos de estimación de distribución (EDA's)*, se pone en antecedentes al lector de los trabajos que sirvieron de base para el presente y de las diferencias del trabajo actual con los anteriores, mostrando que la metodología mostrada es adecuada (y se adecuó) para resolver el problema de optimización de formas.

3.1 Algoritmos de Estimación de Distribución

La computación evolutiva son heurísticas motivadas por sistemas biológicos adaptables. Entre ellas se encuentran: las estrategias evolutivas (EE's), la programación evolutiva (EP), los algoritmos genéticos (AG) y la programación genética (GP). Estas técnicas han demostrado su capacidad para solucionar problemas de optimización, y cuentan con un amplio campo de aplicación en problemas de diversa índole. Durante la última década el estudio y utilización en esta área se ha incrementado, proporcionando más y mejores técnicas de optimización. Cabe mencionar que este tipo de técnicas no son “recetas de cocina” listas para usarse; sino un conjunto de mecanismos que pueden ser adaptados fácilmente para resolver el problema en cuestión.

Entre las ventajas que representa el uso de la computación evolutiva para la solución de problemas de optimización es que generalmente se supone el desconocimiento de las características de la función(es) a optimizar, y los requerimientos de características como un gradiente definido, suavidad, continuidad, etcetera, no son necesarios. A diferencia de los algoritmos de optimización clásica, las técnicas bioinspiradas tienen una alta capacidad de evitar los mínimos locales. Por último, cuentan también con una alta flexibilidad para la representación de las soluciones. Por otro lado, como desventajas se puede mencionar, el costo computacional, que generalmente es superior a los algoritmos de optimización clásica, no se puede asegurar que se encontrará un óptimo(s), y el uso de algoritmos particulares (aún para aquellos que utilizan variables de control autoadaptables) requieren un ajuste de parámetros, para asegurar cierto grado de éxito.

El presente trabajo trata con un nuevo paradigma dentro de la computación evolutiva: los algoritmos de estimación de distribución (EDA's). La idea principal es encontrar una distribución de probabilidad, la cual genera muestras cuyos valores son los óptimos o puntos muy cercanos a los óptimos. Es un algoritmo poblacional, que no utiliza los operadores de cruce y mutación, en su lugar la nueva población se muestrea de una distribución de probabilidad. En el caso más simple, las variables son binarias y se consideran independientes, con lo cual se pueden estimar las probabilidades marginales de cada variable. En otro caso, se puede estimar la estructura o modelo gráfico y las probabilidades conjuntas, o una distribución multivariada en general. Algunas referencias

para este trabajo se encuentran en [13, 1]. La figura 3.1 muestra la forma básica de los algoritmos de estimación de distribución.

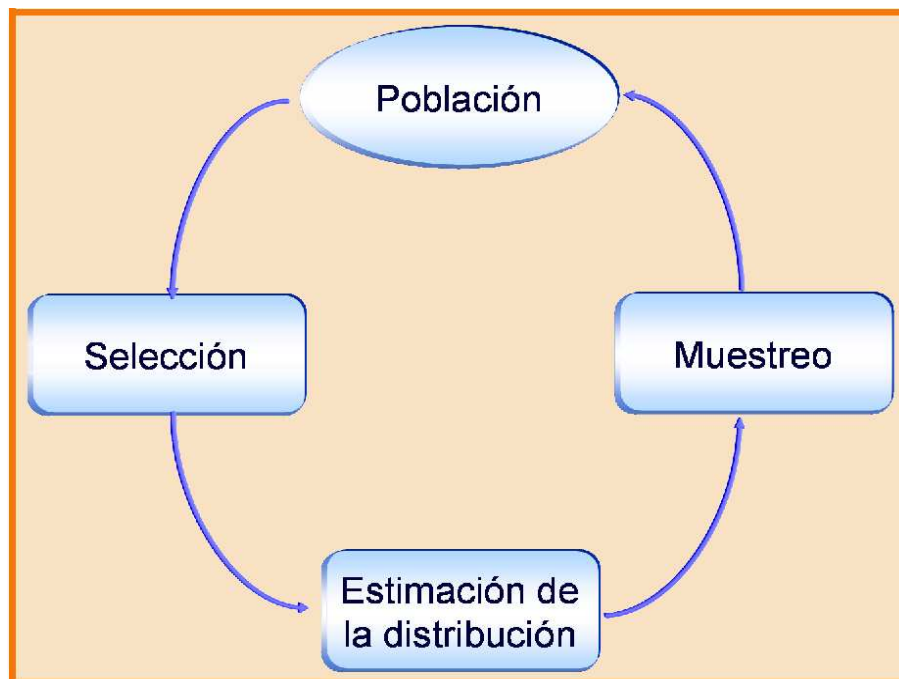


Figura 3.1: Estructura básica de un algoritmo de estimación de distribución.

En el trabajo realizado se supone independencia de las variables para calcular y muestrear la distribución de probabilidad; sin embargo, las posibles dependencias son tomadas en cuenta con el proceso de regularización de la probabilidad descrito en la sección 5.6. Este proceso mejora la exploración, al modificar las probabilidades en una forma congruente espacialmente, facilitando la búsqueda local alrededor de las mejores soluciones encontradas hasta ese momento.

Capítulo 4

Definición del problema

En este capítulo se explica cual es el problema de optimización a resolver. Se exponen las ecuaciones que modelan las funciones objetivo y restricciones, así como las variables y lo que representan en el problema físico. La forma en que se defina la representación de la información para construir las soluciones, y en que se modelen las características físicas deseables de las soluciones, impactará de manera importante en la capacidad del algoritmo de encontrar soluciones adecuadas.

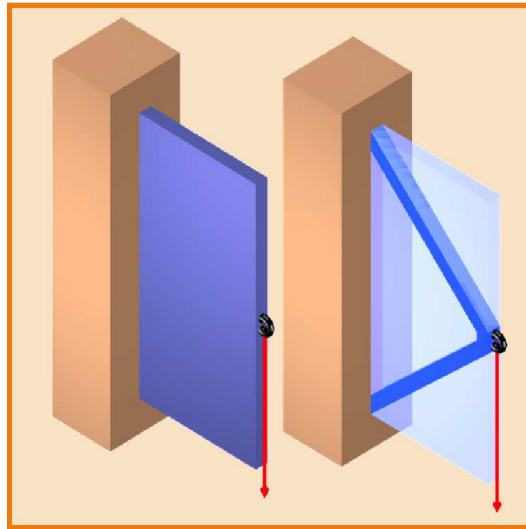


Figura 4.1: Espacio de búsqueda, estado de cargas, y estructura con mínimo peso.

4.1 Definición del problema

El problema multiobjetivo es encontrar un conjunto de estructuras con el mínimo peso y el menor desplazamiento en ciertos puntos dados (los puntos de carga), que cumplan con las siguientes restricciones de diseño: no exceder el máximo esfuerzo Von Mises permitido, no permitir mas de un objeto, ni agujeros pequeños. En la sección siguiente se definirá de manera más amplia el significado de estas funciones objetivo y restricciones. Para el diseño óptimo automatizado de las estructuras o la optimización de formas, el usuario entrega el estado de

cargas y el espacio de búsqueda inicial. La Figura 4.1, muestra en la izquierda el espacio de búsqueda inicial proporcionado por el usuario, y el estado de cargas al que esta sujeto; el lado derecho muestra una de las soluciones sobre el frente de Pareto, que representa la estructura de menor peso, con el mínimo desplazamiento posible para este peso en el punto de aplicación de la carga.

4.2 Discretización y representación

El espacio de búsqueda inicial se discretiza en elementos finitos (triangulares en este caso). Esto produce una malla que puede ser estructurada o no estructurada. En este trabajo hemos probado con ambas, pero en general no se supone ningún conocimiento de la forma, así que una malla no estructurada parece más acorde con la generalidad de los problemas. Cada elemento es representado por una variable binaria (ver Figura 4.2a), cuando esta variable es igual a 0, representa un hueco o un espacio sin material del tamaño de un elemento; de otro modo (igual a 1), representa un valor de espesor dado. De esta manera, cada configuración de una estructura puede ser representada por un vector binario de la misma dimensión que el número de elementos. La Figura 4.2 muestra la discretización del espacio y cómo se mapea de la representación binaria a una forma de alguna estructura.

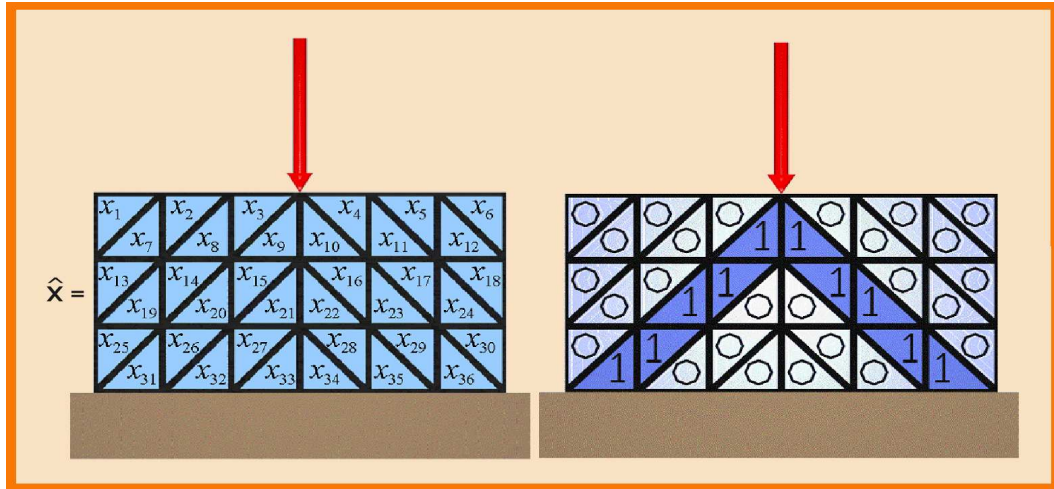


Figura 4.2: Discretización y representación de una estructura.

4.3 Primera función objetivo: Mínimo peso

El primer objetivo es el peso mínimo y esta dado por la Ecuación 4.1, que es la suma de los pesos w_i de cada uno de los elementos que tienen material o están representados por 1 en el vector binario \hat{x} .

Minimizar :

$$W(x) = \sum_{i=0}^n w_i x_i \quad (4.1)$$

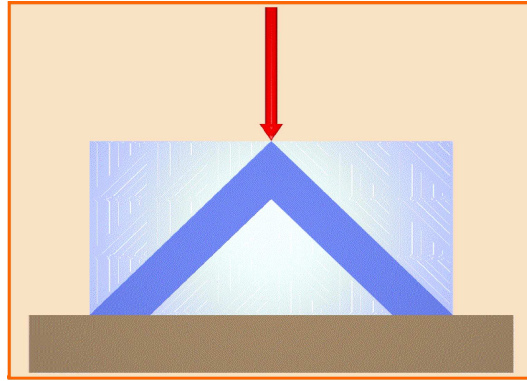


Figura 4.3: Estructura con el mínimo peso en el frente.

4.4 Segunda función objetivo: Mínimo desplazamiento

El segundo objetivo es el mínimo desplazamiento en algunos puntos (nodos), en los ejemplos mostrados se han utilizado los nodos con carga como los puntos donde se desea minimizar el desplazamiento. La Ecuación 4.2 representa la segunda función objetivo, que es la sumatoria de los desplazamientos en los puntos definidos por el usuario. En todos los ejemplos estos puntos fueron los de aplicación de las cargas.

Minimizar :

$$G(\delta) = \sum_{j=0}^m |\delta_j| \quad (4.2)$$

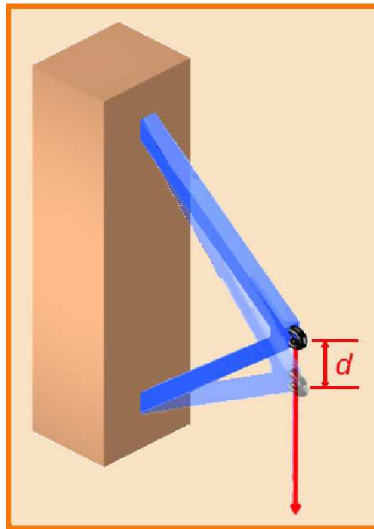


Figura 4.4: Simulación del desplazamiento de una estructura en el punto de carga (línea roja)

4.5 Primera restricción: Máximo esfuerzo Von Misses Permisible

La primera función restricción esta dada por el esfuerzo máximo permisible Von Misses (ver Malvern [10]), el cual representa la resistencia del material al estado de cargas, y es un criterio estándar de diseño. El esfuerzo Von Misses (o de máxima energía de distorsión) es calculado en cada elemento por el método del elemento finito (ver Zienkiewicz [18]). Si este esfuerzo sobrepasa el máximo permisible para el material, aumenta el valor de la restricción, si ningún elemento tiene un valor mayor que este, entonces la restricción vale cero. La restricción se expresa por la Ecuación 4.3.

$$H(x, \rho(\sigma), \gamma(\sigma)) = \left(\sum_{i=0}^n \rho_i(\sigma) \right) \left(\sum_{i=0}^n x_i \gamma_i(\sigma) \right) \quad (4.3)$$

Donde :

$$\rho_i(\sigma) = \begin{cases} 0 & \text{Si } (\sigma_M - \sigma_i) \geq 0 \\ 1 & \text{de otra forma} \end{cases} \quad (4.4)$$

y :

$$\gamma_i(\sigma) = \begin{cases} 0 & \text{Si } (\sigma_M - \sigma_i) \geq 0 \\ (\sigma_i - \sigma_M) & \text{de otra forma} \end{cases} \quad (4.5)$$

En la Ecuación 4.3, la suma de $\rho_i(\sigma)$ cuenta el número de elementos que están esforzados más alla del esfuerzo máximo permisible, x_i es el valor del bit en la i^{esima} posición del arreglo binario; y $\gamma_i(\sigma)$ es el exceso de esfuerzo en el i^{esimo} elemento cuando el esfuerzo en el elemento es mayor que el máximo permisible. Note que las Ecuaciones 4.4 y 4.5 dependen implícitamente de la configuración representada por el arreglo binario \hat{x} , debido a que los valores de estas funciones se derivan de la solución por el método del elemento finito, el cual es completamente dependiente de la configuración de la estructura \hat{x} .

La Figura 4.5 muestra una simulación del estado de esfuerzos en una estructura, utilizando el método del elemento finito.

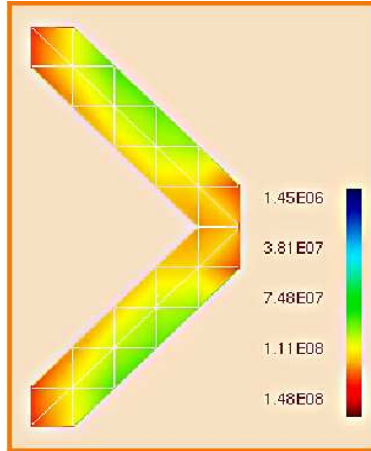


Figura 4.5: Estado de esfuerzos en una estructura encontrados por el método del elemento finito

4.6 Segunda Restricción: Número de Objetos

La segunda restricción penaliza el número de piezas de las que está formada la estructura, el modelo supone que las mejores estructuras son las que están hechas de una sola pieza. La Figura 4.6 muestra una configuración de una estructura formada por 4 objetos, podemos observar que estos objetos son un conjunto de elementos conectados entre sí por los lados. El método del elemento finito puede evaluar las estructuras que están compuestas por varios objetos mientras estos objetos estén conectados al menos por un vértice, desde el punto de vista de ingeniería los objetos conectados solamente por un vértice no son factibles.

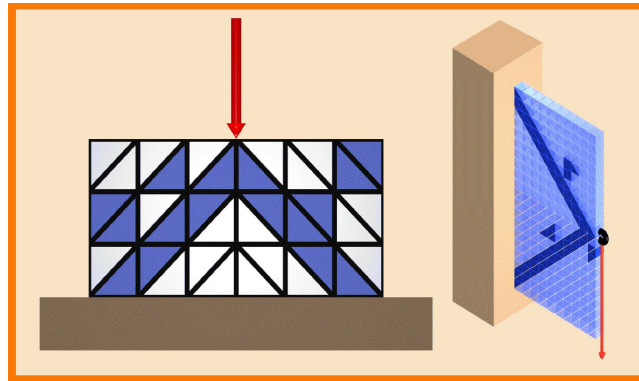


Figura 4.6: Configuración con 4 objetos

4.7 Tercera Restricción: Agujeros Pequeños

La tercera restricción cuenta el número de agujeros pequeños. Un agujero pequeño se define como un agujero del tamaño de un elemento, y que está rodeado (en sus tres lados para elementos triangulares) por elementos que sí tienen material. La Figura 4.7 muestra una configuración con 3 agujeros pequeños.

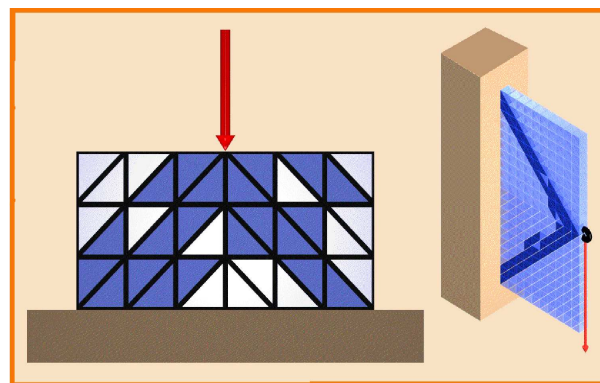


Figura 4.7: Configuración que presenta 3 agujeros pequeños

Capítulo 5

Algoritmo Propuesto

En este capítulo se describe una propuesta de algoritmo para la optimización de formas, el algoritmo se basa en la estimación de probabilidades marginales de las variables a optimizar, ignorando las dependencias entre variables para la estimación de las probabilidades; sin embargo, estas dependencias (o las dependencias más probables) son tomadas en cuenta mediante una regularización de la distribución de probabilidad, que se lleva a cabo cuando se estanca la búsqueda utilizando información de las vecindades.

5.1 Descripción del algoritmo

La Figura 5.1 muestra los principales procedimientos del algoritmo. Primero, se inicializan los vectores de probabilidad \hat{p} utilizando la rutina presentada en la subsección 5.2. Después, cada vector de probabilidad p_i genera un subconjunto de la población. Finalmente, todos los individuos factibles no dominados y algunos de los no factibles (*Conjunto de trabajo de Pareto*), se toman para actualizar las distribución de probabilidad. Cuando la distribución de probabilidad pierde su capacidad de exploración, se realiza un procedimiento de regularizado, tomando información de las vecindades de cada elemento (ver la sección 5.6). Cuando se ha realizado la regularización de la probabilidad, se vacía el *conjunto de trabajo de Pareto*. En cada generación se actualiza el *conjunto conocido de Pareto* para guardar las mejores soluciones encontradas durante todas las generaciones; mientras el *conjunto de trabajo de Pareto* guarda individuos tanto factibles como no factibles, el *conjunto conocido de Pareto* guarda solamente los individuos no dominados factibles, otra diferencia entre el *conjunto de trabajo de Pareto* y el *conjunto conocido de Pareto*, es que el primero guarda no dominados únicamente con respecto a las generaciones desde la última vez que se vació este conjunto, mientras que el segundo guarda los no dominados con respecto a todas las generaciones realizadas. A esta primera propuesta del algoritmo le llamaremos MASO-I (Multiobjective Algorithm for Shape Optimization) para diferenciarlo del que llevará las mejoras propuesta en el capítulo 6. La figura 5.2 muestra de forma esquemática el algoritmo con una pequeña explicación de cada uno de los pasos del mismo. El pseudo-código del algoritmo principal se muestra en la Figura 5.3.

5.2 Inicialización de la Probabilidad

De acuerdo con el algoritmo mostrado en la Figura 5.4, la distribución inicial de probabilidad \hat{p} muy posiblemente generará individuos factibles con diferentes configuraciones. Por otro lado, los vectores de probabilidad pueden ser inicializados con el valor de 0.5 en cada posición (la máxima varianza en la población inicial) o incluso valores aleatorios; sin embargo, la rutina propuesta ayuda a la convergencia y al costo de evaluación, dando tan buenas soluciones como la inicialización de máxima varianza, los valores aleatorios al contrario pueden sesgar la búsqueda y retardar la convergencia del algoritmo sin una ventaja evidente.

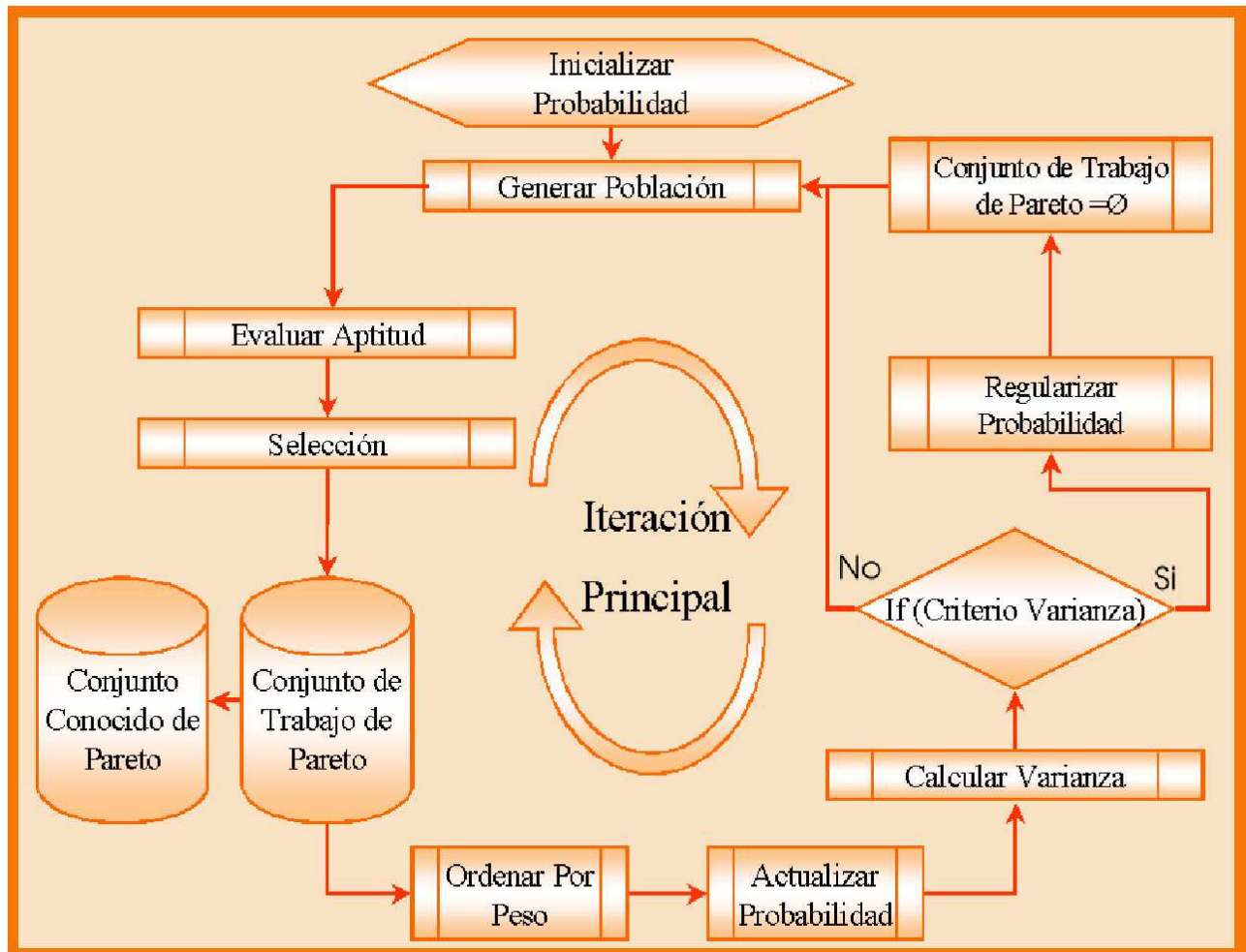


Figura 5.1: Descripción del algoritmo principal (MASO)

Básicamente, la rutina para la inicialización de la probabilidad busca por una configuración factible de bajo peso dentro del espacio de búsqueda. Cuando este individuo es encontrado, la rutina comienza inicializando un vector de probabilidad que muy posiblemente generará esta estructura, y parte de ahí para inicializar los demás con probabilidades que muy posiblemente generarán estructuras más pesadas que la primera, pero permaneciendo dentro de la zona factible.

5.3 Generación de la población

El algoritmo utiliza k vectores de probabilidad \hat{p} , cada vector genera un subconjunto de la población, entonces, cada bit en \hat{x} es generado por un experimento Bernoulli con probabilidad de éxito p_i (ver [1]). Todos los vectores generados \hat{x} son la población actual, de donde saldrán los “mejores” individuos que actualizarán las distribuciones de probabilidad. Los individuos son generados de acuerdo al algoritmo en la Figura 5.5.

5.4 Selección y manejo de restricciones

La selección de los mejores individuos se realiza utilizando el criterio de dominancia de Pareto, tratando las restricciones como tres objetivos más; una vez que han sido seleccionados de esta manera, se separa a los factibles de los no factibles. Los primeros de manera directa para al *conjunto de trabajo de Pareto*, mientras que los segundos serán filtrados nuevamente utilizando el criterio de Pareto, pero ahora solo sobre las restricciones. Este tipo de selección mejora la exploración y extrae información “importante” de los individuos no factibles; pero, al mismo tiempo dirige la población hacia la zona factible.

5.5 Actualización de la probabilidad

La actualización de la probabilidad se realiza utilizando los individuos seleccionados conforme se indica en la sección 5.4. Estos individuos deben de ser ordenados por el primer objetivo (peso); entonces, cada vector de probabilidad será modificado por un porcentaje (proporcional al número de vectores de probabilidad) de los mejores individuos.

$$m = \begin{cases} 1 & \text{Si } \lceil \frac{C_t}{K} \rceil < 1 \\ \lceil \frac{C_t}{K} \rceil & \text{de otra forma} \end{cases} \quad (5.1)$$

Donde m es la cantidad de individuos que actualizan cada vector, C_t es la cantidad de individuos en el *conjunto de trabajo de Pareto*, y K es la cantidad de vectores de probabilidad. En la figura 5.6 se muestra esquemáticamente como cada subconjunto de individuos actualizan un vector de probabilidad.

Se calculan los vectores de frecuencias \hat{F}_l para cada subconjunto de vectores de probabilidad, como lo indica la ecuación 5.2, donde existen tantos vectores \hat{F}_l como número de vectores de probabilidad.

$$F_{l,j} = \sum_{i \in l} x_{i,j}^l \quad (5.2)$$

Donde $x_{i,j}^l$ representa el valor del bit j , en el vector i que pertenece al conjunto l (el conjunto de individuos que actualizan el vector de probabilidad l).

Entonces en forma vectorial al ecuación de actualización de la probabilidad se puede escribir como:

$$\hat{P}_l^{t+1} = \lambda \hat{P}_l^t + (1 - \lambda) \frac{1}{m} \hat{F}_l \quad (5.3)$$

Donde $(1 - \lambda)$ es la *tasa de aprendizaje* (*learning rate*, ver [1]) la cual indica que tanto se preserva del conocimiento de la distribución anterior y cuanto se “aprende” de la nueva.

5.6 Regularización de la probabilidad

Debido al reforzamiento implícito del aprendizaje en las probabilidades de la Ecuación 5.3, las distribuciones de probabilidad tienden a perder su capacidad de exploración a través de las generaciones; entonces, los vectores de probabilidad son modificados utilizando información de las vecindades, donde se supone una alta correlación positiva entre las variables x_i , esta correlación supuesta, generalmente es cierta debido a las implicaciones físicas que representa: es difícil que un elemento con material este junto a varios que no lo tienen, o viceversa. Esta correlación no será verdad en las fronteras, donde el que un elemento tenga material, no implica que su vecino también lo tenga. Por otro lado, al modificar la probabilidad no se toman decisiones deterministas; sino que únicamente se le da a la distribución la capacidad de explorar el espacio en direcciones con altas posibilidades de éxito, que generalmente implican expandir o estrechar la estructura, mover las fronteras, o rellenar

huecos. La Ecuación 5.4, muestra cómo la distribución de probabilidad se puede modificar para mejorar la exploración. Básicamente, el funcional $U(f)$ se minimiza para encontrar una nueva distribución de probabilidad, la cual preserva información de los mejores individuos encontrados, o la mejor distribución encontrada hasta ese momento; pero, trata de explorar en las vecindades, como se muestra en la Figura 5.7.

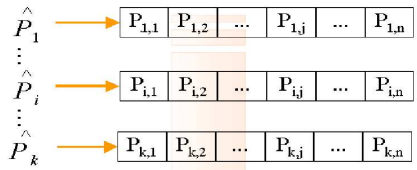
$$\min U(f) = \sum_j^{NoElem.} (f_j - g_j)^2 + \mu \sum_j^{NoElem.} \sum_{l \in \text{vecinos}} (f_j - f_l)^2 \quad (5.4)$$

En la Ecuación 5.4, g es la distribución conocida o la distribución actual (cualquier vector de probabilidad \hat{p}) y f es la nueva distribución (el nuevo vector de probabilidad con mejor capacidad de exploración, a través de las vecindades). En la segunda sumatoria, f_j es la probabilidad en la $j^{ésima}$ posición del vector de probabilidad, y f_l son los elementos vecinos físicamente (normalmente tres para elementos triangulares, cuatro para rectangulares, etc.). Minimizando $U(f)$ la función f será la mejor función que preserva la forma de g ; pero, que minimiza al mismo tiempo la diferencia entre los valores de probabilidad de las vecindades. Esta diferencia se pondera por el parámetro μ (que para los experimentos realizados se le han dado valores entre 0.35 y 0.5). La Figura 5.7 muestra la probabilidad antes y después del proceso de regularización utilizando la Ecuación 5.4. Más información acerca de el proceso se encuentra en [11](Marroquin). Como se describe en la subsección 5.1 el *conjunto de trabajo de Pareto* debe de ser vacío después de la regularización de la probabilidad. Esto obedece a un sesgo muy fuerte que ocurre en al búsqueda si se mantienen las soluciones encontradas durante todo el proceso.

5.7 Cálculo de la varianza

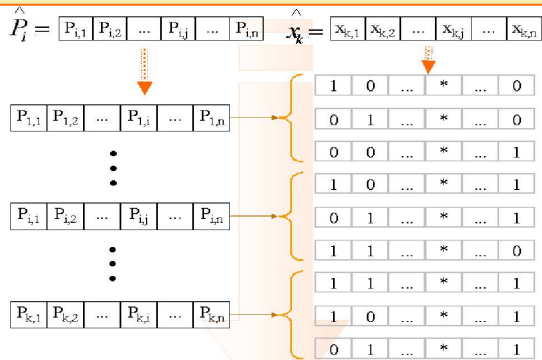
En la subsección 5.6, se abordó el problema de la exploración pobre, y se dio una técnica para mejorarla. Ahora, queda la cuestión de cuándo debe de ser aplicada la regularización de la probabilidad. Se propone una medida para detectar la pérdida de la exploración. Básicamente, se cuenta el número de valores en cada vector de probabilidad que se encuentran dentro de los límites de probabilidad (no muy cercanos a cero ni a uno); se toma el valor mayor (el vector con mayor número de valores dentro de los límites de probabilidad), y este valor se considera una medida de la varianza de las muestras que pueden ser producidas. Cuando este valor es menor que un umbral dado por el usuario, se regularizan los vectores de probabilidad. Se sugiere que el valor de este parámetro sea aproximadamente el 12% del número de elementos del espacio de búsqueda. El algoritmo en la Figura 5.8 describe el cálculo de la medida de varianza.

Inicializar la Probabilidad



Los vectores de probabilidad se inicializan tratando de generar en su mayor parte soluciones factibles en diferentes zonas del espacio de búsqueda.

Generar la Población



Cada vector de probabilidad genera un subconjunto de los individuos de la población.

Evaluar la aptitud y Selección

Cada individuo es evaluado por el método del elemento finito, y se cuentan el número de agujeros y de objetos. Se seleccionan los individuos utilizando el criterio de Dominancia de Pareto

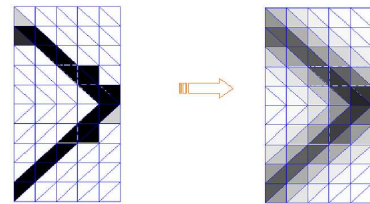
Actualizar los conjunto de Pareto

Se actualiza el conjunto de trabajo de Pareto, el cual actualiza las distribuciones de probabilidad. También se actualiza el conjunto conocido de Pareto, el cual tiene todos los no-dominados encontrados durante todo el proceso.

Ordenar por peso

Los individuos en el conjunto de trabajo de Pareto son ordenados por el peso.

Regularizar las probabilidades



$$\min U(f) = \sum_{j=1}^{NoEle} (f_j - g_j)^2 + \mu \sum_{j=1}^{NoEle} \sum_{l \in Vecindad} (f_j - f_l)^2$$

Se regulariza la probabilidad para incrementar la exploración en la vecindad de los mejores individuos.

Calcular la varianza

Una medida de la varianza ayuda a determinar cuando se debe de hacer el regularizado de la probabilidad.

Actualizar probabilidad

$$P_{i,1} P_{i,2} \dots P_{i,j} \dots P_{i,n}^{t+1} = \lambda * \mu_{i,1} \mu_{i,2} \dots \mu_{i,j} \dots \mu_{i,n} + (1-\lambda) * P_{i,1} P_{i,2} \dots P_{i,j} \dots P_{i,n}^t$$

$$P_{k,1} P_{k,2} \dots P_{k,j} \dots P_{k,n}^{t+1} = \lambda * \mu_{k,1} \mu_{k,2} \dots \mu_{k,j} \dots \mu_{k,n} + (1-\lambda) * P_{k,1} P_{k,2} \dots P_{k,j} \dots P_{k,n}^t$$

Un subconjunto de los individuos en el conjunto actual de Pareto actualiza a cada vector de probabilidad.

Figura 5.2: Descripción esquemática del algoritmo.

```

Insertar los siguientes parámetros:
μ:      Parámetro de suavizado, peso de la diferencia con las vecindades
        (ver [11] y la subsección 5.6 en la Ecuación 5.4)
MinVar: Parámetro de varianza mínima, criterio para realizar el regularizado
LSP,LIP: Límites superior e inferior de probabilidad
NV:     Número de vectores de probabilidad.
NIM:    Número de individuos generados por cada vector de probabilidad.
λ:      Parámetro de aprendizaje para la actualización de la probabilidad
        (ver subsección 5.5 ).
NG:     Número máximo de generaciones que se procesarán.
NIS:    Número máximo de regularizaciones de la probabilidad.

Inicialización_de_Probabilidad ();
k = 0
i = 0
Mientras (i < NG & k<NIS)
{
  Generar_Población ();
  Evaluar_Objetivos_y_Restricciones ();
  Seleccionar_No-Dominados_por_Objetivos_y_Restricciones();
  Seleccionar_No-Dominados_Factibles ();
  Seleccionar_No-Dominados_No-Factibles ();
  Ordenar_por_Peso();
  Actualizar_Distribución_de_Probabilidad ();
  Calcular_Varianza ();
  Si (VarianzaCalculada < MinVar)
  {
    Regularizar_Probabilidad ();
    k++;
  } #Fin Si
  i++;
} #Fin Mientras

```

Figura 5.3: Pseudo-código del Algoritmo Multiobjetivo para Optimización de Formas

T y Z son dos configuraciones de estructuras.
 $\max\text{VonMisses}$ y $\min\text{VonMisses}$ son el máximo y el mínimo esfuerzo
en T (todos los elementos presentes).

```

Para i=1..NoElementos
   $T_i = 1$  #Fin Para NoElementos
Resolver_VonMisses(T);
Intervalo = ( $\max\text{VonMisses} - \min\text{VonMisses}$ ) / (NoElementos)
umbral =  $\max\text{VonMisses}$  - Intervalo
Hacer
{
  Para i=1..NoElementos
  {
    Si ( $\text{EsFuerzoVonMisses}_i < \text{Umbral}$ )
       $Z_i = 0$ 
    Sino
       $Z_i = 1$ 
  } #Fin Para NoElementos
  umbral = umbral - Intervalo
} mientras (  $\text{EsNoFactible}(Z)$  ) # Verdadero cuando Z es no factible
Intervalo = ( $\text{umbral} - \text{VonMisses}$ ) / ( NoVectores-1 );
Para i=1..NoVectores
{
  Para k=2..NoElementos
  {
    Si (  $\text{EsFuerzoVonMisses}_i < \text{umbral}$  )
       $\text{VectorProbabilidad}_{k,i} = \text{LimInfProb}$ ;
    Sino
       $\text{VectorProbabilidad}_{k,i} = \text{LimInfProb}$ ;
  } #Fin Para NoElementos
  umbral = umbral - Intervalo;
} #Fin Para NoVectores

```

Figura 5.4: Pseudo-código para la inicialización de la probabilidad

```

Para i=1..NoVectoresProbabilidad
{
  Para j=1..NoIndividuosPorVector
  {
    Para k=1..NoDeElementos
    {
      Si (Aleatorio()) < Pi,k )
        x(i-1)(NoIndividuosPorVector)+j,k = 1
      Sino
        x(i-1)(NoIndividuosPorVector)+j,k = 0
    } # Fin Para k
  } # Fin Para j
} # Fin Para i

```

Figura 5.5: Pseudo-código para la generación de la población

$$\begin{matrix}
 \begin{matrix} P_{1,1} & P_{1,2} & \dots & P_{1,j} & \dots & P_{1,n} \end{matrix} \\
 \vdots \\
 \begin{matrix} P_{i,1} & P_{i,2} & \dots & P_{i,j} & \dots & P_{i,n} \end{matrix} \\
 \vdots \\
 \begin{matrix} P_{k,1} & P_{k,2} & \dots & P_{k,j} & \dots & P_{k,n} \end{matrix}
 \end{matrix}
 = \lambda * \begin{matrix}
 \begin{matrix} \mu_{1,1} & \mu_{1,2} & \dots & \mu_{1,j} & \dots & \mu_{1,n} \end{matrix} \\
 \begin{matrix} \mu_{i,1} & \mu_{i,2} & \dots & \mu_{i,j} & \dots & \mu_{i,n} \end{matrix} \\
 \begin{matrix} \mu_{k,1} & \mu_{k,2} & \dots & \mu_{k,j} & \dots & \mu_{k,n} \end{matrix}
 \end{matrix}
 + (1-\lambda) * \begin{matrix}
 \begin{matrix} P_{1,1} & P_{1,2} & \dots & P_{1,j} & \dots & P_{1,n} \end{matrix} \\
 \begin{matrix} P_{i,1} & P_{i,2} & \dots & P_{i,j} & \dots & P_{i,n} \end{matrix} \\
 \begin{matrix} P_{k,1} & P_{k,2} & \dots & P_{k,j} & \dots & P_{k,n} \end{matrix}
 \end{matrix}$$

Un subconjunto de los individuos en el conjunto actual de Pareto actualiza a cada vector de probabilidad.

Figura 5.6: Esquema de actualización de la probabilidad

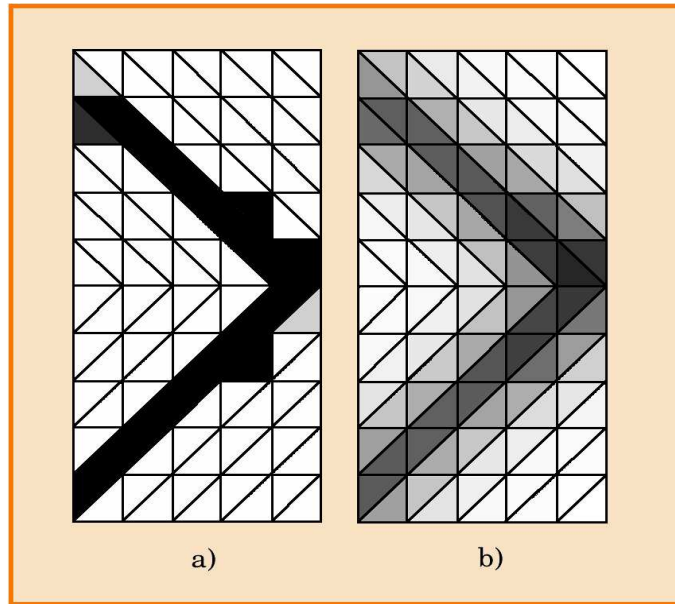


Figura 5.7: Probabilidad: (a) Antes de la regularización, (b) Después de la regularización.

```

VarianzaCalculada=0
Para l=1.. NoVectores
{
  Varianza=0
  Para i=1..NoElementos
  {
    Si (VectorProbabilidadl,i ≤ LimInfProb
      && VectorProbabilidadl,i ≥ LimSupProb)
      Varianza=Varianza+1;
  } #Fin Para NoElementos
  Si (Varianza > VarianzaCalculada)
    VarianzaCalculada =Varianza;
  }#Fin Para NoVectores

```

Figura 5.8: Pseudo-código del algoritmo de cálculo de la varianza

Capítulo 6

Mejoras del Algoritmo

En este capítulo se tratan algunas mejoras, que van desde modificaciones en la selección y criterio de regularización de la probabilidad, hasta técnicas de discriminación por densidad en el frente (cantidad de soluciones concentradas en una zona), para el uso de un tamaño fijo del conjunto de soluciones que actualizan las probabilidades. También se muestra un esquema básico de paralelización de las evaluaciones de las funciones. Para diferenciar el algoritmo resultante de estas mejoras con el algoritmo presentado en el capítulo 5 le llamaremos MASO-II.

6.1 Modificación en al selección

La selección se realiza de manera semejante a como se menciona en la sección 5.4, la diferencia radica en que se deja de seleccionar no factibles en el momento en que el número de factibles se vuelve igual o mayor que el número de vectores de probabilidad.

6.2 Varianza mínima para la regularización

En lugar de dar un valor de varianza mínima para que se realice la regularización de la probabilidad, se guarda el valor mínimo de la medida de varianza propuesta en la sección 5.7, desde la última regularización (o desde el comienzo del proceso de optimización, según sea el caso), cuando este valor mínimo no cambia en k_{var} veces, entonces se aplica el proceso de regularización y el valor mínimo de la varianza se reinicia igual al número de elementos. El valor de k_{var} se calcula de acuerdo con la ecuación 6.1:

$$k_{var} = \frac{\log(LIP)}{\log(1 - \lambda)} + 1 \quad (6.1)$$

Donde LIP es el límite inferior de la probabilidad y $(1 - \lambda)$ es la *tasa de aprendizaje* (ver 5.5). Este valor es calculado para dar tiempo suficiente a que algunos valores de la probabilidad vayan desde el límite superior hasta el inferior, en caso de que se repita k_{var} veces el valor del bit que esta actualizando el vector de probabilidad en cuestión.

6.3 Uso de archivo fijo

Basado en una técnica presenta por Zitzler en el algoritmo SPEA2 en [19], se eliminan las soluciones que presentan una mayor densidad de puntos en su zona, dentro del espacio de las soluciones. La información de densidad se incorpora para discriminar entre individuos que tienen valores de aptitud semejantes (o que son todos no

dominados). La técnica de estimación de la densidad utilizada es una adaptación del método de la distancia al k -ésimo vecino más cercano, donde la densidad a cada punto es una función decreciente a la distancia del k -ésimo punto más cercano. Para cada individuo i la distancia (en el espacio de las funciones objetivo) se calcula a todos los individuos j en el archivo, y se guarda en una lista. Después de ordenar la lista en orden creciente, el k -ésimo elemento da la distancia buscada, denotada como σ_i^k . Como uso común k es igual a la raíz cuadrada del tamaño de la muestra. Después, la densidad $D(i)$ correspondiente al punto i se define por $D(i) = \frac{1}{\sigma_i^k + 2}$.

La única diferencia del cálculo de la densidad descrito en el párrafo anterior, utilizado por el SPEA2 y la usada en el algoritmo de optimización de formas es el uso de valores normalizados entre 0 y 1 de las funciones objetivos. Utilizando la densidad de cada punto se eliminan los puntos con mayor valor en la función de densidad (en caso de ser necesario), hasta que quedan en el archivo fijo el doble del número de vectores de probabilidad. El utilizar esta técnica tiene un efecto positivo en la dispersión del frente y la amplitud del mismo, además de minimizar el costo de evaluación, pudiendo trabajar con un número pequeño de vectores de probabilidad.

Después de la regularización se llena el archivo fijo con los no dominados del *conjunto conocido de Pareto*, eliminando los necesarios con la función de densidad, para utilizar nuevamente solo el doble del número de vectores.

6.4 Paralelización del cómputo

Para resolver problemas de gran escala en un tiempo relativamente corto, se paralelizo el algoritmo utilizando la librería *mpi* y el ambiente *lam*. Esto fue echo bajo un esquema maestro-esclavo, donde el nodo maestro genera la población, envía un número proporcional de individuos a cada nodo esclavo para su evaluación (codificados a nivel bit), y recibe las aptitudes con las que realiza la selección y actualización de las probabilidades. La reducción del tiempo de cómputo es de aproximadamente 68% para el experimento mostrado en la figura 6.2 (hipotético diseño de un puente), la figura 6.3 muestra la solución de mínimo peso utilizando condiciones de simetría, en un espacio de búsqueda de tamaño 2^{632} . Utilizando 6 nodos (1 maestro y 5 esclavos), la mejora del tiempo depende del número de variables y nodos (un mayor tiempo de evaluación de cada solución eficiente la utilización del cluster). La figura 6.4 corresponde a un problema que tiene las mismas condiciones de carga que la figura 6.2, pero esta discretizado en 1184 elementos (espacio de búsqueda de 2^{1184}), esto da una idea clara de la capacidad del algoritmo y los alcances reales referentes a la dimensionalidad del problema, utilizando la paralelización de las evaluaciones. La figura 6.1 muestra esquemáticamente la forma en que se paraleliza el algoritmo.

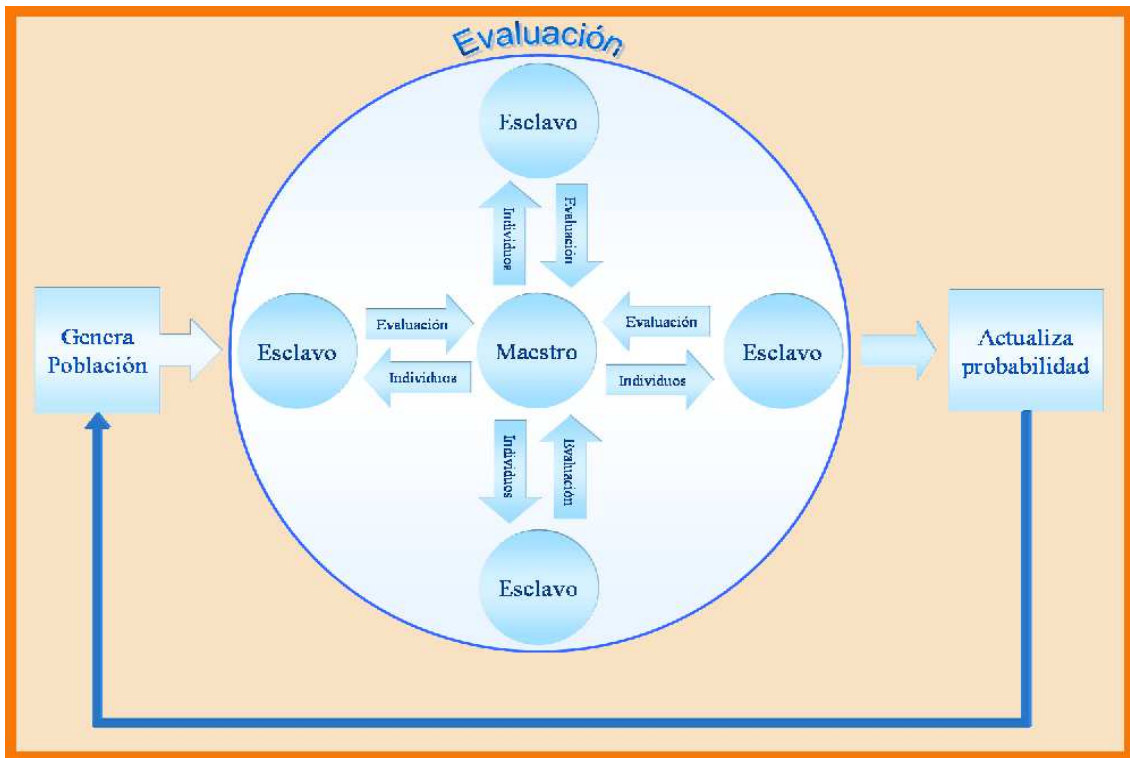


Figura 6.1: Esquema de paralelización del algoritmo.

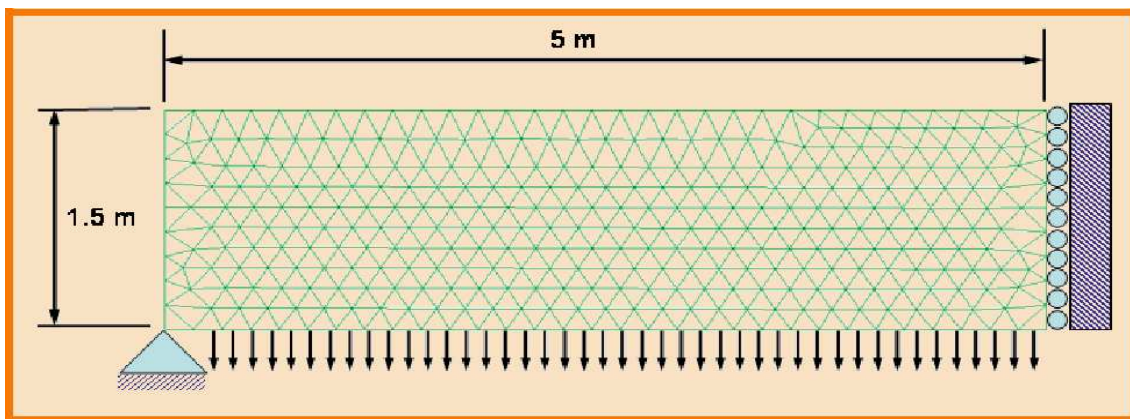


Figura 6.2: Estado de cargas para diseño del puente

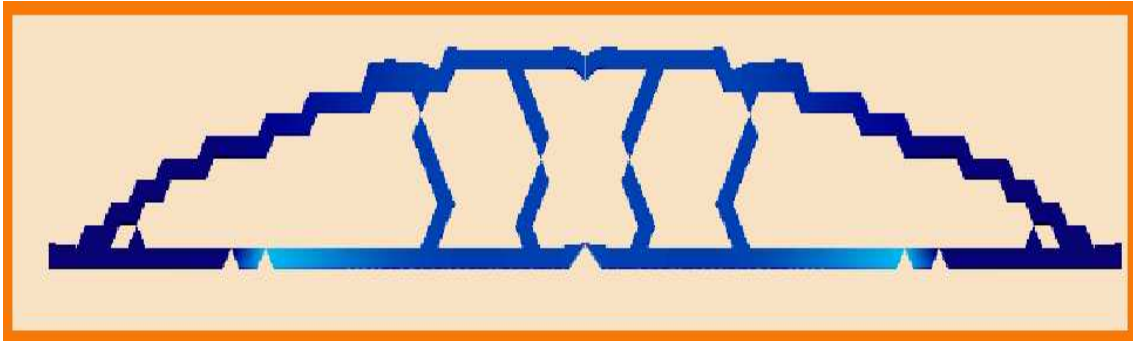


Figura 6.3: Solución de mínimo peso, evaluación paralelizada, 632 elementos.

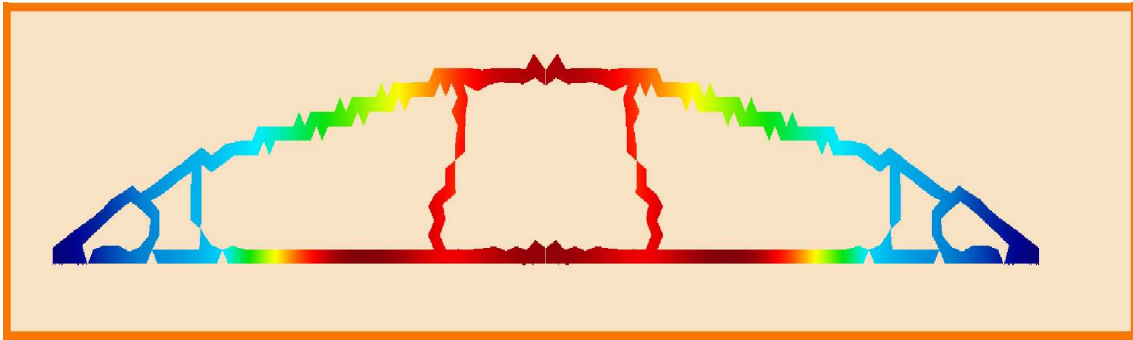


Figura 6.4: Solución de mínimo peso, evaluación paralelizada, 1184 elementos.

Capítulo 7

Métricas

Dentro de este capítulo se presentan algunas métricas de desempeño, que ayudan a mostrar el comportamiento del algoritmo y su capacidad para encontrar las soluciones. Las métricas miden convergencia y diversidad.

7.1 Métricas de desempeño

A fin de mostrar el comportamiento y capacidad del algoritmo, se utilizaron dos métricas de desempeño propuestas por Deb y Jain en [5]. Estas métricas miden convergencia y diversidad en las soluciones encontradas.

En [5], sugieren que las propiedades que deben de ser consideradas para las métricas de los MOEAS (Multi-Objective Evolutionary Algorithm), deben de ser las siguientes:

1. La métrica debe de tomar un valor entre cero y uno en un sentido absoluto. Para que esta pueda ser comparada entre generaciones, una escala absoluta de una métrica entre cero y uno permitirá evaluar el cambio en el valor de la métrica entre una generación y otra.
2. El valor objetivo (o deseado) de la métrica (calculado para un idealmente convergido o diversificado conjunto de puntos) debe ser conocido.
3. La métrica debe proveer un incremento o decremento monotono en su valor, cuando la población mejore o se deteriore en poco. Esto también ayudará a evaluar que tan superior es un conjunto de aproximación con respecto de otro.
4. La métrica debe ser escalable a cualquier número de objetivo. Sin embargo, esta no es una propiedad absolutamente necesaria, pero si se tiene, seguramente será conveniente para evaluar problemas referentes a la escalabilidad de los MOEAs en términos del número de objetivos.
5. La métrica podría ser computacionalmente barata, sin embargo esta no es una condición estricta que deba tener.

7.1.1 Métrica de convergencia

Un conjunto de referencia P^* se determina de la unión de varias corridas independientes (del *conjunto conocido de Pareto* obtenido por cada corrida). Esta métrica básicamente mide la distancia normalizada entre el frente que se está midiendo $F^{(t)}$, y el conjunto de referencia P^* . Toma un valor en el intervalo $[0, 1]$; cercano a 0 significa que el valor de convergencia $C(P^{(t)})$ es mejor, el valor en $C(P^{(N)})$ (al final de la corrida), indica que tan lejos se encuentra el frente medido del conjunto de referencia, que es grosso modo la efectividad del algoritmo

para encontrar las mejores soluciones (suponiendo que el conjunto de referencia tiene las mejores soluciones), el medir $C(P^{(t)})$ durante las generaciones nos indica cómo el conjunto evoluciona y se acerca cada vez más a el conjunto de referencia. La métrica de convergencia se mide como se muestra a continuación:

1. Identificar el conjunto no dominado $F^{(t)}$ de $P^{(t)}$ (una población).
2. Para cada punto i en $F^{(t)}$, calcular la menor distancia euclideana normalizada a P^* como en la Ecuación 7.1, f_k^{max} y f_k^{min} son el máximo y el mínimo valor de la función en la k^{esimo} función objetivo en P^* .

$$d_i = \min_{j=1}^{|P^*|} \sqrt{\sum_{k=1}^M \left(\frac{f_k(i) - f_k(j)}{f_k^{max} - f_k^{min}} \right)^2} \quad (7.1)$$

3. Calcular la métrica de convergencia promediando la distancia normalizada para todos los puntos en $F^{(t)}$:

$$C(P^{(t)}) = \frac{\sum_{i=1}^{|F^{(t)}|} d_i}{|F^{(t)}|} \quad (7.2)$$

Deb y Jain en [5] proponen normalizar la métrica de convergencia entre el valor máximo (usualmente $C(P^{(0)})$): $\hat{C}(P^{(t)}) = C(P^{(t)})/C(P^{(0)})$. Esta normalización no fue calculada para este trabajo.

7.1.2 Métrica de diversidad

Usando el mismo conjunto de referencia que en la subsección 7.1.1, se calcula la métrica de diversidad propuesta por Deb y Jain en [5]. A continuación se describe el cálculo de la métrica de diversidad, tal como la proponen Deb y Jain en [5].

Basado en un estudio de Farhang-Mehr y Azarm (ver [6]) sugirió una técnica basada en entropía.

La idea esencial es que los puntos no dominados de cada generación se proyecten en un hiperplano conveniente, perdiendo dimensionalidad de los puntos. El plano se divide en un número de pequeñas celdas (o $(M - 1) - dimensional$ cajas). Dependiendo de si cada celda contiene un punto no dominado o no, la métrica de diversidad es definida. Si todas las celdas tienen por lo menos un punto, la mejor medida de diversidad posible es alcanzada (con respecto a el número de celdas escogido). Si algunas celdas no están representadas por un punto no dominado, entonces la diversidad es pobre. Los parámetros requeridos del usuario son la dirección del coseno del plano de referencia, el número de celdas (G_i) en cada una de las $(M - 1)$ dimensiones, y el conjunto de puntos objetivo (o de referencia) P^* . El procedimiento es el siguiente:

1. De $P^{(t)}$, encuentre el conjunto $F^{(t)}$ que son los no dominados por P^* .
2. Para cada celda indexada por (i, j, \dots) , calcular los siguientes dos arreglos:

$$H(i, j, \dots) = \begin{cases} 1, & \text{Si la celda tiene un punto representativo en } P^*. \\ 0. & \text{De otra forma.} \end{cases} \quad (7.3)$$

$$h(i, j, \dots) = \begin{cases} 1, & \text{Si } H(i, j, \dots) \text{ y la celda tienen un punto representativo en } F^{(t)}. \\ 0. & \text{De otra forma.} \end{cases} \quad (7.4)$$

CD-R.desktop

3. Asignar un valor $m(h(i, j, \dots))$ a cada celda dependiendo de su función $h()$ y la de sus vecinos. De la misma manera, calcular $m(H(i, j, \dots))$ usando $H()$ para los puntos de referencia.
4. Calcular la métrica de diversidad promediando los valores de $m()$ para $h()$ con respecto a aquellos para $H()$:

$$D(P^{(t)}) = \frac{\sum_{i,j,\dots \text{ para } H(i,j,\dots) \neq 0} m(h(i, j, \dots))}{\sum_{i,j,\dots \text{ para } H(i,j,\dots) \neq 0} m(H(i, j, \dots))} \quad (7.5)$$

En el caso simple, el valor de la función $m()$ para una celda puede ser calculado utilizando su $h()$ y la de dos vecinos por dimensión. Con un conjunto de tres valores binarios consecutivos de $h()$, existen un total de 8 posibilidades. Cualquier función debe ser asignada manteniendo en mente lo siguiente:

- Un 111 es el mejor distribuido y un 000 es el peor.
- Un 010 o un 101 significan un patrón periódico con una buena dispersión y puede ser mejor valuado que un 110 o un 011. Por ejemplo, el valor de arriba puede estar cubriendo un 50% de las celdas pero con una amplia dispersión (tal como 10101010) mejor que otro conjunto que tenga la misma cobertura pero con una menos dispersión (tal como 1111100000).
- Un 110 o un 011 deben de valer más que un 001 o un 100, por que cubren más celdas.

Basado en las observaciones de arriba, los valores de $m()$ para $h()$ mostrados en la Figura 7.1 son sugeridos:

| $h(\dots j - 1\dots)$ | $h(\dots j\dots)$ | $h(\dots j + 1\dots)$ | $m(h(\dots j\dots))$ |
|-----------------------|-------------------|-----------------------|----------------------|
| 0 | 0 | 0 | 0.00 |
| 0 | 0 | 1 | 0.50 |
| 1 | 0 | 0 | 0.50 |
| 0 | 1 | 1 | 0.67 |
| 1 | 1 | 0 | 0.67 |
| 0 | 1 | 0 | 0.75 |
| 1 | 0 | 1 | 0.75 |
| 1 | 1 | 1 | 1.00 |

Figura 7.1: Valores sugeridos para $m()$

Mayores detalles con respecto de esta métrica pueden ser encontrados en [5].

Capítulo 8

Experimentos

Los experimentos mostrados en este capítulo utilizan las métricas del capítulo anterior y se muestran de manera gráfica las soluciones encontradas, todos los experimentos presentan buenas soluciones, se compara también con algunas soluciones encontradas por búsqueda exhaustiva para un problema de pocas variables.

Se muestran 3 casos de estudio, el primero es un problema con un espacio de búsqueda pequeño del cual se pueden conocer algunas soluciones en el *conjunto verdadero de Pareto*, lo cual sirve para comparar la efectividad del algoritmo. El segundo experimento tiene las mismas condiciones que el primero, pero con un espacio de búsqueda mayor; además, se calculan las métricas antes mencionadas para mostrar el comportamiento y capacidad del algoritmo. El último experimento aumenta la complejidad del problema, al hacer más complejo el estado de cargas, utilizar mallas no estructuradas, y aumentar la dimensionalidad (número de elementos) del problema.

8.1 Experimento 1: Placa simplemente apoyada, 36 elementos.

El primer experimento está definido por una placa simplemente soportada en las dos esquinas inferiores, con una carga puntual en la parte superior-media de la misma. Se encontraron a través de búsqueda exhaustiva algunas soluciones en el *conjunto verdadero de Pareto* (las soluciones simétricas), lo cual es posible para un problema de pocas variables binarias (pocos elementos), para mallas más refinadas esto se vuelve imposible desde el punto de vista de costo computacional, dado que la cantidad de soluciones posibles es 2^N siendo N el número de elementos, en el espacio físico de búsqueda. Este experimento se realizó utilizando al algoritmo MASO-I (ver 5.1), con 25 vectores de probabilidad, cada vector genera 10 vectores de bits (individuos), el parámetro λ es de 0.85, la varianza mínima permitida es de 4.0, el parámetro μ es de 0.5 para la regularización de la probabilidad, el límite superior de probabilidad es de 0.9998 y el inferior de 0.0002, el esfuerzo máximo permisible es de 25×10^6 , se realizaron 1000 generaciones, y el desplazamiento se minimizó en el punto de aplicación de la carga. La Figura 8.1 muestra el estado de cargas y las dimensiones de la placa.

8.1.1 Experimento 1. Comparación con el Conjunto Verdadero de Pareto

Para el experimento 1 se calcularon las soluciones simétricas que pertenecen al *conjunto verdadero de Pareto*. Se realizaron 30 corridas y se compararon contra las soluciones conocidas. Se tienen 13 soluciones conocidas en el conjunto verdadero. El promedio de estas mismas soluciones encontradas por el algoritmo en las corridas realizadas es de 12.6667 (de las 13 conocidas), con una desviación estándar de 0.479463. El promedio de las soluciones totales encontradas es de 20.667 con una desviación estándar de 1.3218. El promedio de las soluciones encontradas que son dominadas por alguna solución de las conocidas en el conjunto verdadero es de 0.066667

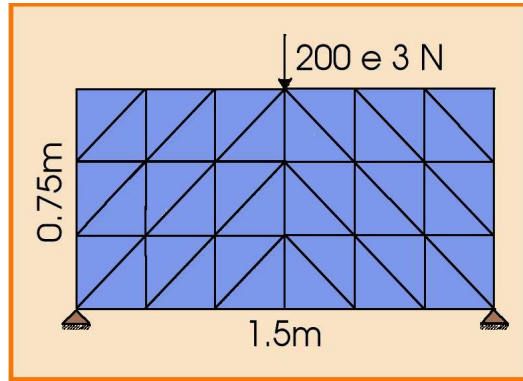


Figura 8.1: Dimensiones y estado de carga para el experimento 1

| | |
|-----------------------------------|-------------------------|
| <i>Módulo de Young</i> | $2.1 \times 10^{11} Pa$ |
| <i>Módulo de Poisson</i> | 0.2 |
| <i>Espesor</i> | 0.10m |
| <i>Esfuerzo máximo permisible</i> | $25 \times 10^6 Pa$ |

Figura 8.2: Propiedades del material del experimento 1

con una desviación estándar de 0.253708 en la Figura 8.3, se puede observar el *frente de Pareto verdadero* y los puntos no dominados encontrados en las 30 corridas.

8.1.2 Experimento 1. Métrica de convergencia

Se calculó la métrica de convergencia $C(P^{(N)})$ para el experimento 1, utilizando un frente de referencia que contiene las 13 estructuras conocidas del conjunto verdadero de Pareto y las no dominadas de las 30 corridas independientes. La media de la métrica calculada para todas las 30 corridas en la última generación es de 0.000377365 (mejor mientras está más cerca de 0) con una desviación estándar de 0.0010709297. En la Figura 8.5 se muestra una gráfica con los valores de la métrica de convergencia para las 30 corridas y en la Figura 8.6 se muestra el comportamiento en la métrica de convergencia a través de las generaciones para una corrida típica del algoritmo.

8.1.3 Experimento 1. Métrica de diversidad

La métrica de diversidad es mostrada en la figura 8.7, fue calculada con el frente de referencia y una línea de proyección encontrada utilizando mínimos cuadrados. Se utilizaron 100 celdas para el cálculo de esta métrica, las medidas muestran en general un comportamiento bueno, un valor de 1 en la métrica, representa la mejor diversidad posible, en este caso la media es de 0.90407, con una desviación estandar de 0.0538451.

8.1.4 Experimento 1: Conclusiones

El experimento 1 muestra la capacidad del algoritmo de encontrar las soluciones óptimas. Aunque las métricas arrojan valores muy buenos, se debe considerar que este caso es un problema de muy baja dimensionalidad en comparación con los problemas reales. Experimentos posteriores demostrarán la capacidad del algoritmo de encontrar buenas soluciones aún en problemas de alta dimensionalidad. El fin de resolver este problema de

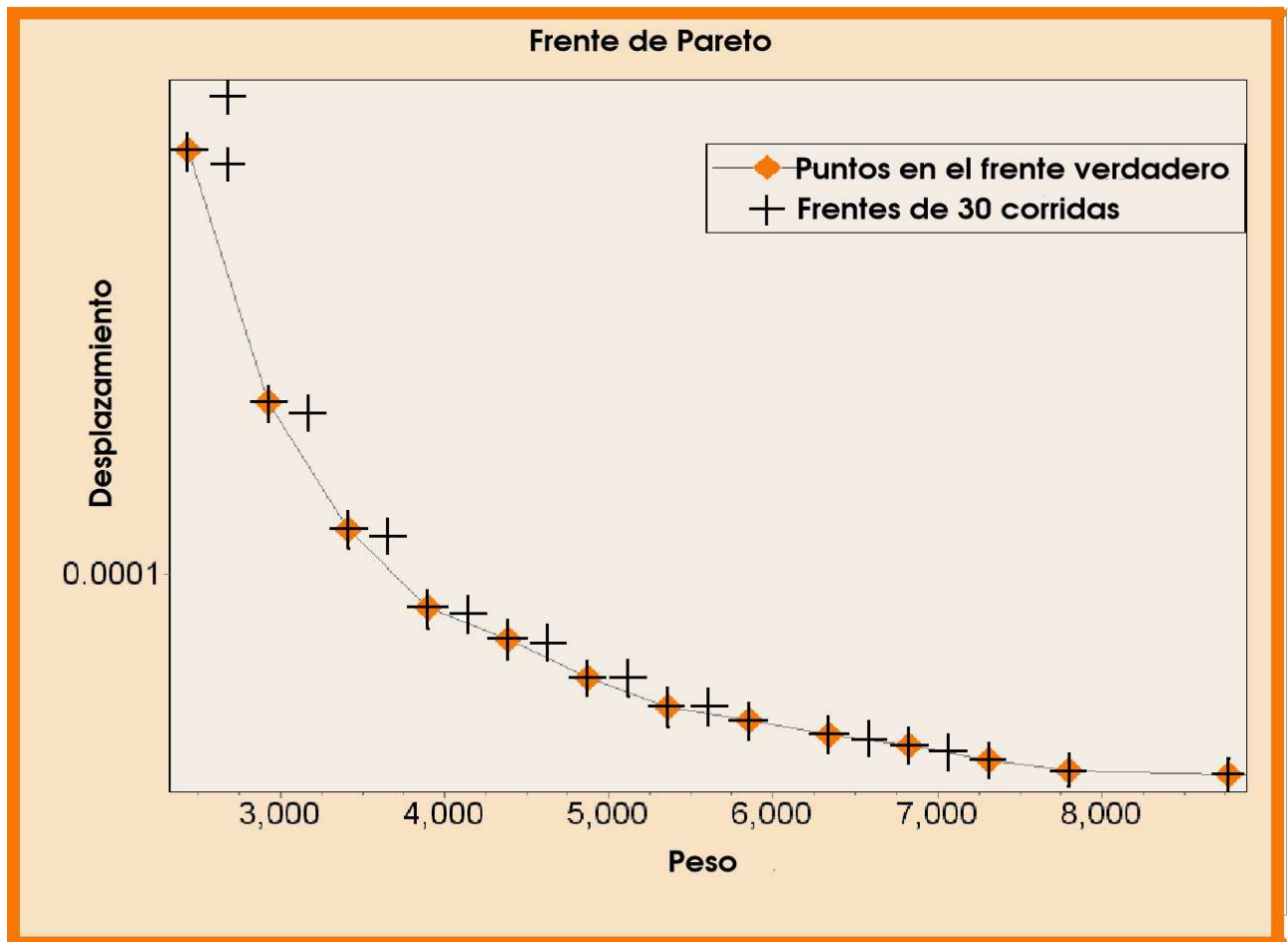


Figura 8.3: Frente Verdadero con diamantes, y frentes de las 30 corridas con cruces.

baja dimensionalidad es la de comparar las soluciones obtenidas por el algoritmo con óptimos conocidos. la gráfica de convergencia contra generaciones comienza en 0, lo que no es de esperarse de manera general en los algoritmos multiobjetivo; en este caso se debe a que las soluciones encontradas en un principio (que son muy pocas, en este caso 1 o 2 puntos), se encuentran ya en el conjunto de referencia, debido en su mayor parte al proceso de inicialización de las probabilidades, donde las soluciones de menor desplazamiento (y más pesadas) se encuentran fácilmente. Conforme avanzan las generaciones, las soluciones que se van encontrando ya no son iguales a las que se encuentran en el conjunto de referencia; sin embargo estas se van mejorando durante el proceso evolutivo, hasta que se logra llegar cerca del conjunto de referencia con un número grande de puntos (alrededor de 20 en este caso). La métrica de diversidad indica que los puntos están dispersos en el frente, para una dimensionalidad mayor sería muy importante poder mantener esta capacidad de dispersión. El experimento 1 reporta resultados promisorios, en experimentos posteriores se mostrará, como se conserva la dispersión del frente y la convergencia en una buena medida, cabe hacer notar que la mayoría de los algoritmos entregan buenos resultados para baja dimensionalidad.

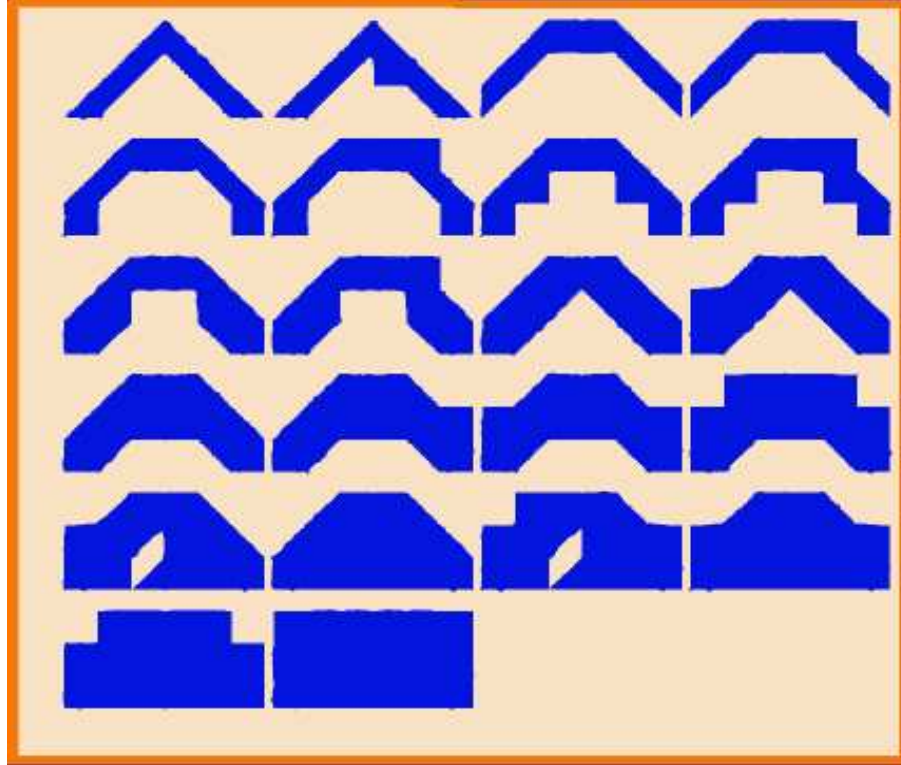


Figura 8.4: Estructuras resultantes de una corrida típica.

8.2 Experimento 2: Placa simplemente apoyada, 144 elementos.

El experimento 2 tiene las mismas condiciones de carga que el experimento 1, como se muestra en la Figura 8.8; pero en este caso el espacio de búsqueda se ha discretizado en 144 elementos. Dado que el frente de Pareto no es conocido, las comparaciones se han llevado a cabo contra un frente de referencia, resultado de 30 corridas independientes. Se corrió utilizando el algoritmo MASO-I, el número de vectores que se utilizaron en este caso es de 100, la varianza mínima es de 12.0, se corrieron 1800 generaciones. Los demás parámetros son los mismos que en el experimento 1.

8.2.1 Experimento 2. Frente de referencia

La Figura 8.10 muestra el frente de referencia y todos los individuos no dominados encontrados en 30 corridas. Aunque, algunos de ellos son dominados por el frente de referencia (en rombos), no fueron dominados en su corrida independiente.

8.2.2 Experimento 2. Métrica de convergencia

La Figura 8.12 muestra el valor $C(P^{(t)})$ calculado para la última generación. Como se puede observar, el valor está muy cercano a 0, con media de 0.00304246875 y desviación estándar de 0.00126309733 lo cual implica una buena convergencia. La Figura 8.13 muestra el comportamiento de la métrica de convergencia en una corrida típica.

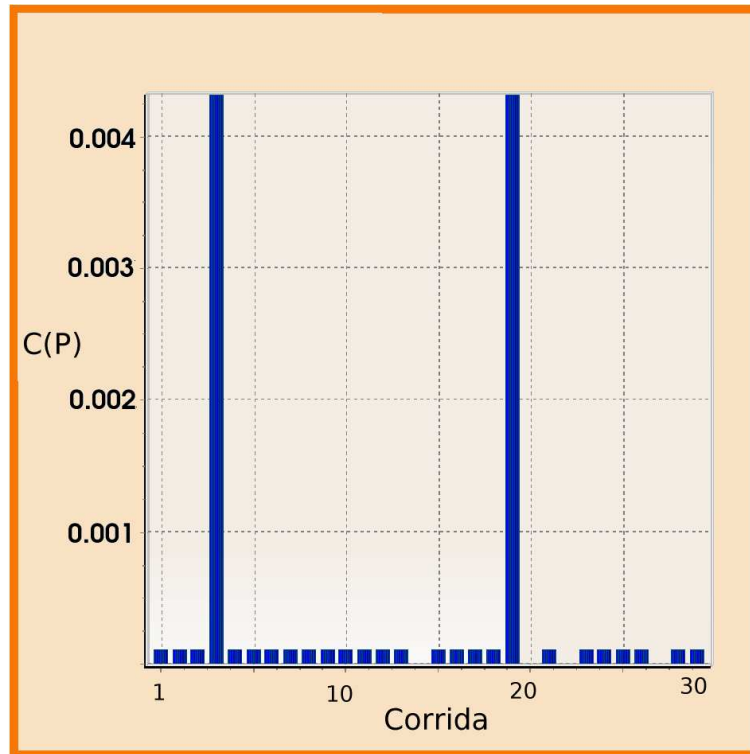


Figura 8.5: Valores de $C(P^{(N)})$ (convergencia) para 30 corridas del experimento 1.

8.2.3 Experimento 2. Métrica de diversidad

La figura 8.14 muestra los valores de la métrica de diversidad para el experimento 2, se utilizaron 100 celdas para su calculo, la línea de referencia sobre la que se proyectaron las soluciones se calculo por mínimos cuadrados. La media de la métrica de diversidad para las 30 corridas es de 0.89477576 con una desviación estándar de 0.04209184

8.2.4 Experimento 2. Conclusiones

El algoritmo MASO-I continua mostrando un buen desempeño aunque el espacio de búsqueda se aumente a 2^{144} ; sin embargo, se aumento el número de vectores y generaciones, lo que incrementa el costo computacional. Si es necesario aumentar el número de vectores y generaciones en las proporciones mostradas en el experimento 2 cada vez que se aumente la dimensionalidad del problema, sería muy difícil resolver problemas de miles de elementos (más parecidos a las aplicaciones reales) debido al costo computacional. Las mejoras mostradas en el capítulo 6 ayudan a minimizar el costo computacional, sin perder la capacidad de encontrar buenas soluciones. Notese que la métrica de convergencia que básicamente representa la distancia media entre el frente de referencia y el frente que esta evolucionando, puede ayudar a mostrar la exploración realizada por el algoritmo, en los puntos donde crece más rapidamente, significa que esta insertando nuevas soluciones en el frente evolutivo, y que estas soluciones se encuentran relativamente lejos del frente de referencia. Cuando comienza a mostrar un comportamiento descendente nuevamente, es por que las soluciones se estan mejorando y se acercan al frente de referencia, cuando los cambios son mínimos en la métrica, también lo son en las soluciones encontradas hasta eso momento por el algoritmo.

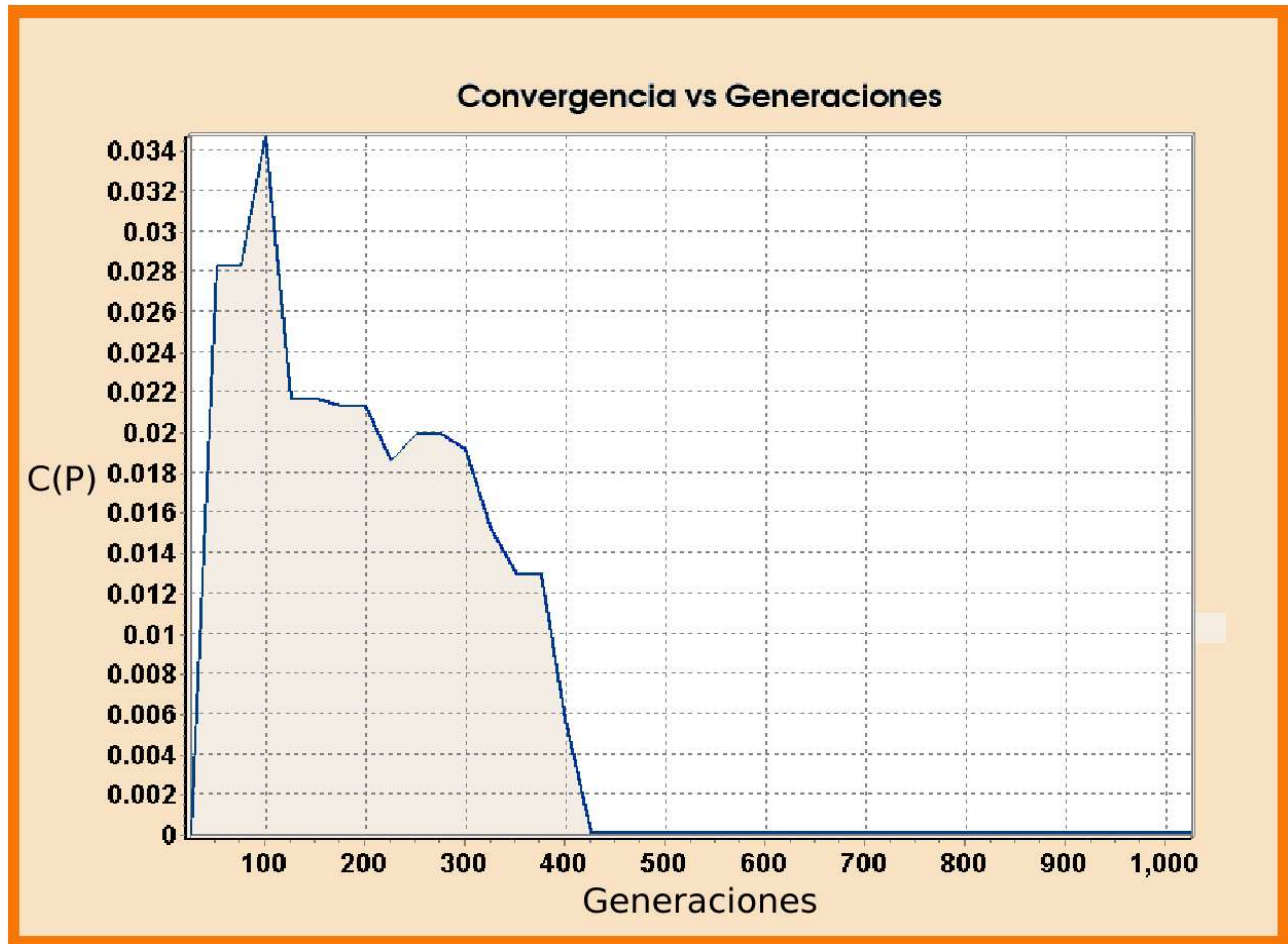


Figura 8.6: Comportamiento de la convergencia para el experimento 1.

8.3 Experimento 3: Diseño de un marco para bicicleta.

El experimento 3 consiste en el diseño hipotético de un marco para bicicleta. La métrica de convergencia se calculó utilizando un frente de referencia de 20 corridas independientes. Se utilizó una malla no estructurada con 403 elementos triangulares (es decir un espacio de búsqueda de 2^{403}) para evaluar la aptitud con el método del elemento finito. La Figura 8.15 muestra el espacio inicial y las condiciones de carga, la Figura 8.16 muestra las propiedades del material. Algunas estructuras para una corrida se muestran en la Figura 8.18. El frente de referencia (en gris) y algunos frentes de corridas (en rojo) se presentan en la figura 8.17. Se utilizaron 32 vectores de probabilidad, cada uno de los cuales generó 10 individuos. La razón de aprendizaje fue de 0.2 ($\lambda = 0.8$). El desplazamiento se minimizó en los 3 nodos con carga. El peso propio no fue considerado como carga en el problema de optimización. El algoritmo utilizado para el proceso de optimización tiene implementadas las mejoras del capítulo 6 (MASO-II).

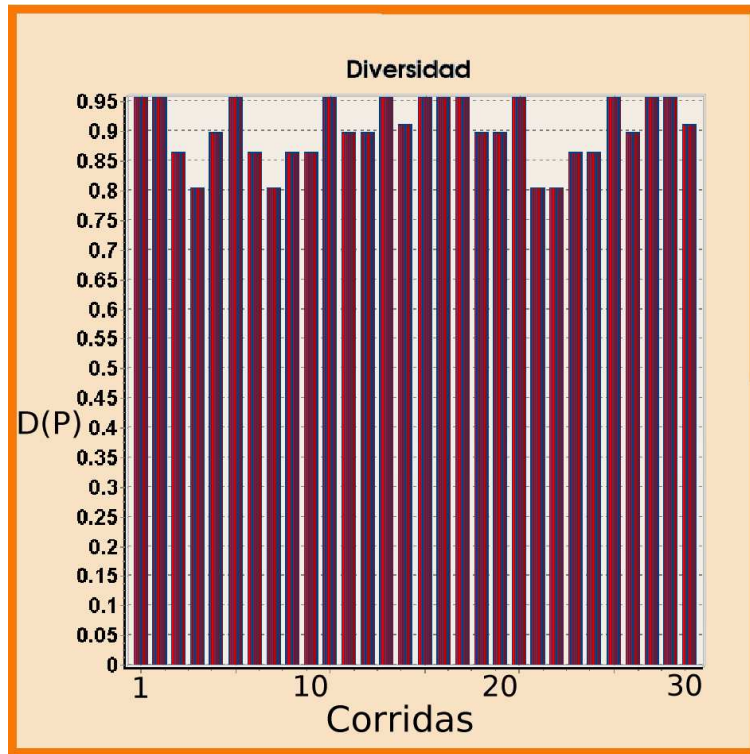


Figura 8.7: Métrica de diversidad para el experimento 1.

8.3.1 Experimento 3. Métrica de convergencia

Se calculó la métrica de convergencia para las 20 corridas. La Figura 8.19 muestra la métrica de convergencia $C(P^t)$ en la última generación para todas las corridas. La media fue de 0.0055549539 (0 es mejor) con una desviación estándar de 0.004197812. La Figura 8.20 muestra el comportamiento de la convergencia, la gráfica muestra el valor de convergencia $C(P^t)$ calculado cada 25 generaciones.

8.3.2 Experimento 3. Conclusiones

El experimento 3 se corrió utilizando el algoritmo mejorado con la adición de las técnicas mostradas en el capítulo 6, el número de vectores utilizados es bajo en comparación con los que se tendrían que utilizar en el algoritmo original, donde el número de vectores debe ser cercano al número de soluciones en el frente, que en este caso son alrededor de 900. el uso de la paralelización del algoritmo hizo que el tiempo de computo fuese relativamente corto de aproximadamente 90 minutos, para realizar 1.35×10^6 evaluaciones en promedio, sin utilizar los nodos a su máxima capacidad, por cuestiones de transmisión (se utilizan al 40% de su capacidad aproximadamente, 8 nodos esclavos y un maestro). Aunque el número de vectores es bajo, las soluciones son bastantes y están distribuidas en el frente, una solución de menor peso que generalmente es la más difícil de alcanzar (aunque no podemos asegurar que sea el mínimo) es mostrada en la figura 8.18 para una corrida cualquiera, es practicamente líneas trazadas de los puntos de apoyo a los puntos de carga, lo que nos dice que si no es el óptimo de un extremo del frente está muy cercano a este, pues es difícil construir una estructura de menor peso que la mostrada. El conocer las soluciones reales por búsqueda exhaustiva es practicamente imposible, y seguramente lo será durante mucho tiempo, dado que realizar 2^{403} evaluaciones de la estructura no

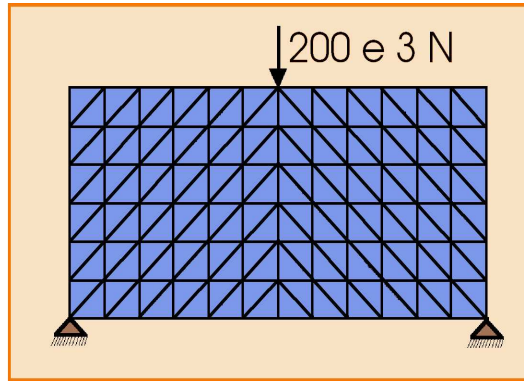


Figura 8.8: Placa simplemente apoyada, discretizada en 144 elementos para el experimento 2.

| | |
|-----------------------------------|-------------------------|
| <i>Módulo de Young</i> | $2.1 \times 10^{11} Pa$ |
| <i>Módulo de Poisson</i> | 0.2 |
| <i>Espesor</i> | 0.10 m |
| <i>Esfuerzo máximo permisible</i> | $25 \times 10^6 Pa$ |

Figura 8.9: Propiedades del material para el experimento 2.

puede ser considerado siquiera aún con la paralelización del computo. La gráfica de convergencia contra tiempo (o generaciones) muestra un comportamiento ascendente lo que no quiere decir que se aleje de la solución; sino que va encontrado más puntos, y estos puntos están relativamente lejos del frente de referencia. El que no regrese a una tendencia descendente muy posiblemente tenga que ver con el número de generaciones, tal vez el algoritmo aún estaba en una etapa de exploración al momento en que se finalizó. Para tratar de determinar si el número de generaciones es el adecuado, se realizó una corrida de 15000 generaciones y se midió la convergencia a través de las generaciones. Los resultados se muestran en la gráfica de la figura 8.21, donde podemos ver que después de las 2500 generaciones los cambios bruscos en la métrica disminuyen, y se atean más después de las 8000 generaciones. Estos movimientos en la métrica están relacionados con los puntos nuevos que entran y salen del *conjunto conocido de Pareto*; se concluye que las 5000 generaciones; si bien, no son consideradas como el mejor número de generaciones para el problema, si representan una buena aproximación ya que el beneficio en exploración de más generaciones es poco. Resulta interesante notar que aunque las soluciones están distribuidas en el frente y el número de vectores es bajo, las soluciones mostradas en la figura 8.17 no están tan cercanas al frente como las mostradas en los primeros experimentos, debido principalmente al incremento en la complejidad del problema, que es una malla no estructurada que incrementa el número de combinaciones factibles y de mínimos locales. La dimensionalidad del problema afecta también en buena medida, y la reducción de los vectores de probabilidad que ahora exploran el frente por partes utilizando la estrategia presentada en la sección 6.3; se debe observar que esta forma de discriminación de las soluciones favorece a los extremos y a los puntos más aislados, lo que vuelve a los extremos en atrayentes de la exploración. Con respecto a la nueva forma de exploración, notese que lugares que tienen un grupo de no dominados en una misma zona del frente en el tiempo t difícilmente serán explorados en el tiempo $t + 1$, aunque este conjunto de no dominados (por la población en ese instante) sean realmente mínimos locales.

Por otro lado, el espacio de búsqueda es exponencial con respecto a las variables y el costo del computacional del algoritmo (MASO-II) utilizando un archivo de tamaño fijo para la actualización de las probabilidades crece de forma mucho menor. Los resultado preliminares del problema de 1184 elementos se lograron con

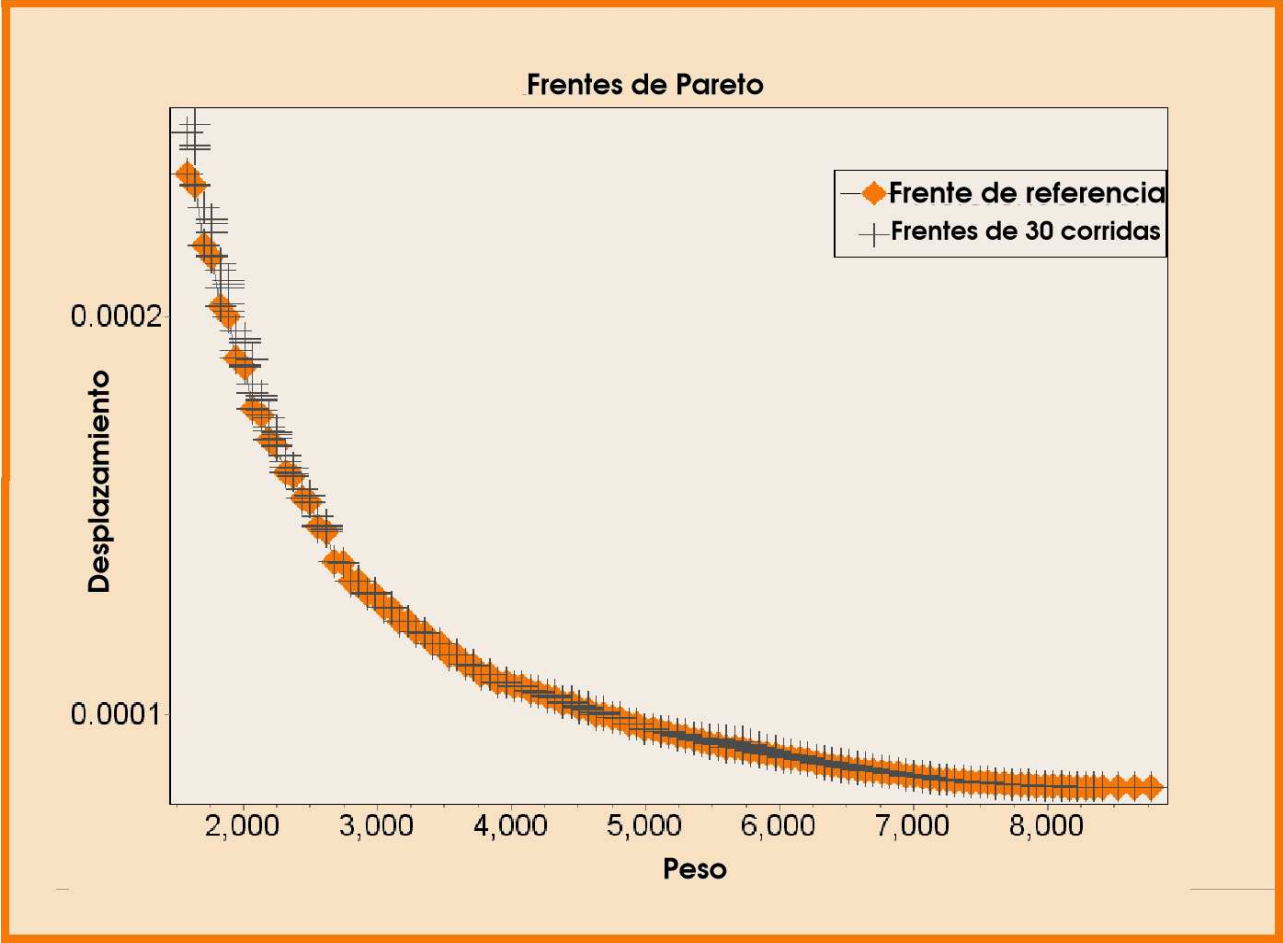


Figura 8.10: Frente de referencia y no dominados en cada corrida, para experimento 2.

aproximadamente el doble de evaluaciones que los resultado de este experimento, aunque el espacio de búsqueda creció de 2^{403} a 2^{1184} y la relación entre el número de variables es de 2.93. Como se mostró en la sección 6.4, al estrategia resulta adecuada para problemas multiobjetivo de alta dimensionalidad.

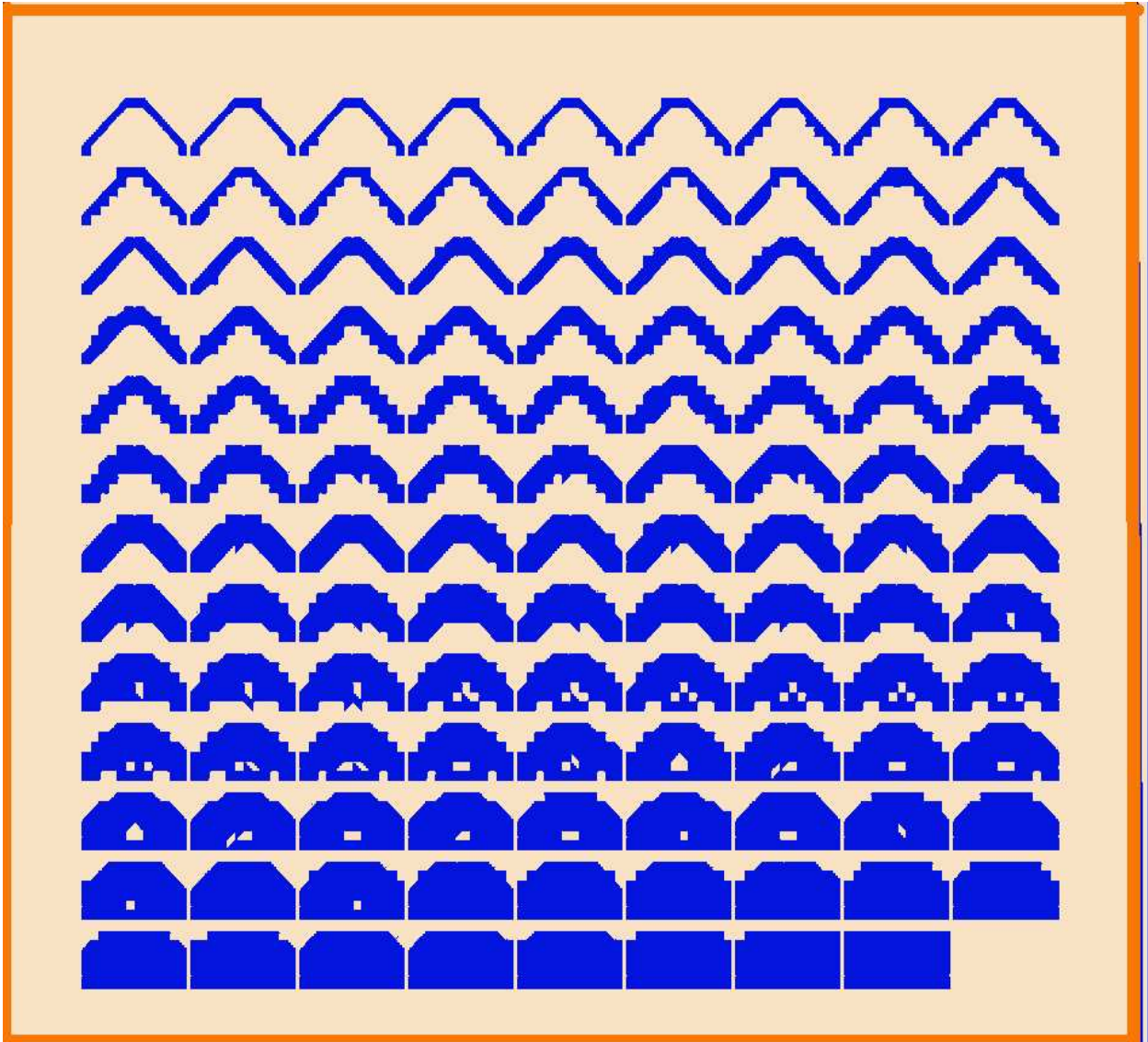


Figura 8.11: Estructuras para una corrida típica, experimento 2 .

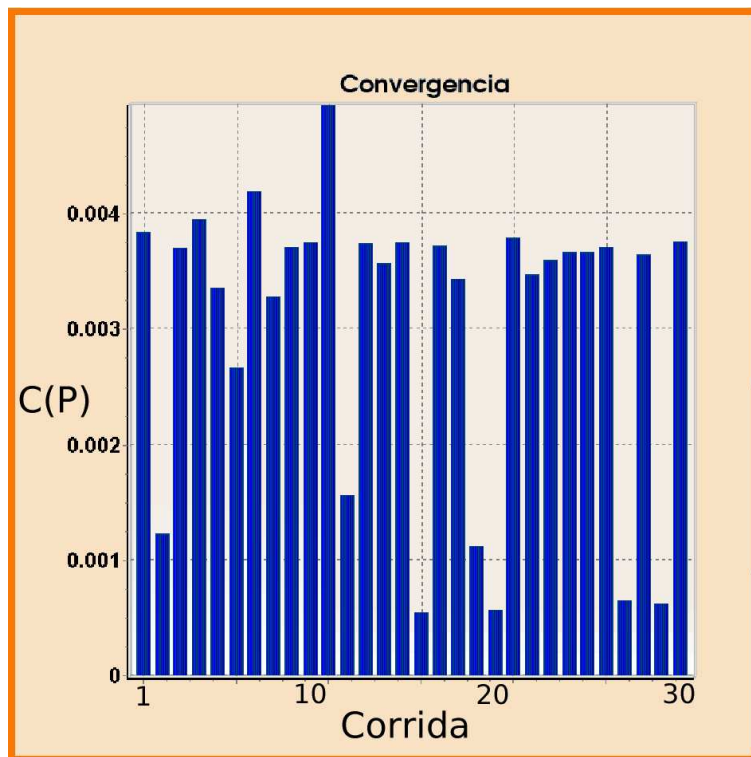


Figura 8.12: Valores de convergencia en 30 corridas, del experimento 2.

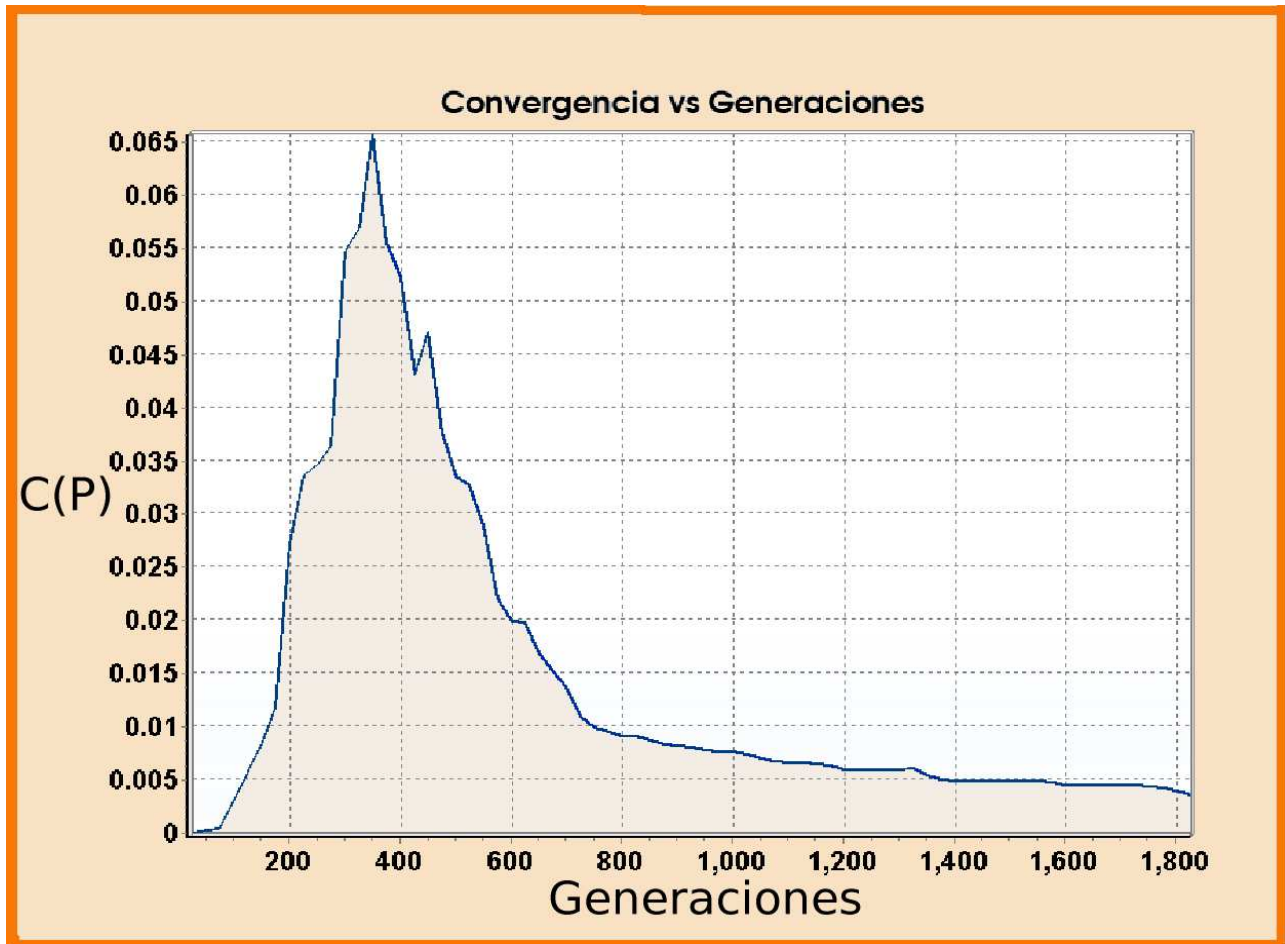


Figura 8.13: Comportamiento de la convergencia en una corrida típica, del experimento 2.

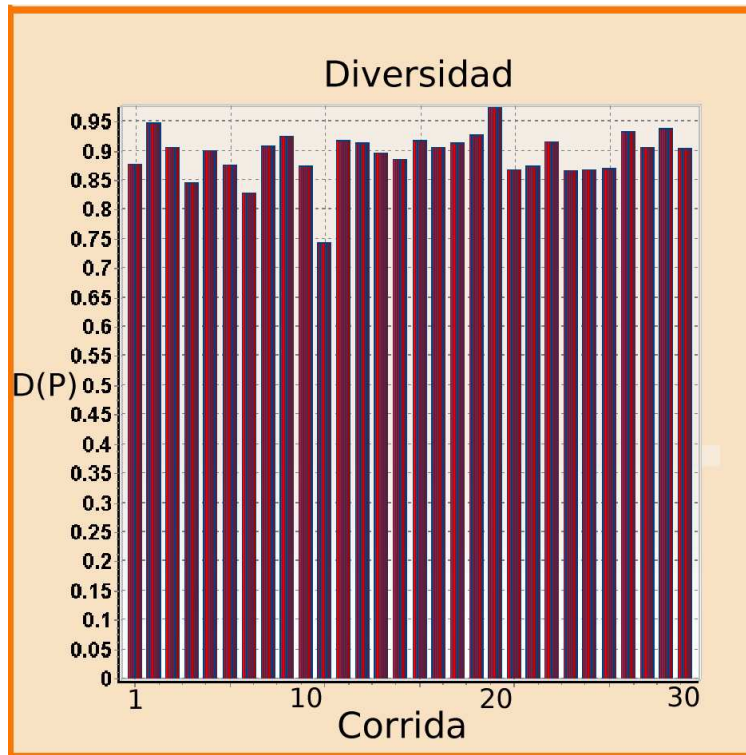


Figura 8.14: Valores de la métrica de diversidad para 30 corridas del experimento 2.

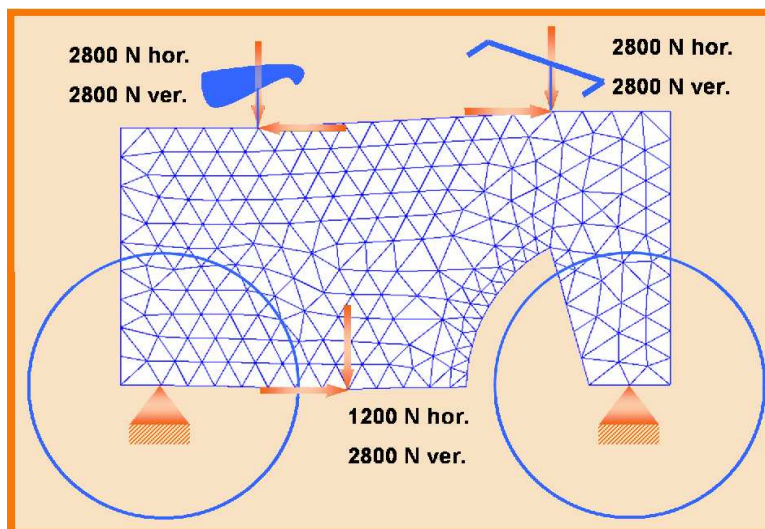


Figura 8.15: Definición del problema para el experimento 3.

| | |
|-----------------------------------|----------------------|
| <i>Módulo de Young</i> | $70 \times 10^9 Pa$ |
| <i>Módulo de Poisson</i> | 0.2 |
| <i>Espesor</i> | 0.005m |
| <i>Esfuerzo máximo permisible</i> | $100 \times 10^6 Pa$ |

Figura 8.16: Propiedades del material para el experimento 3.

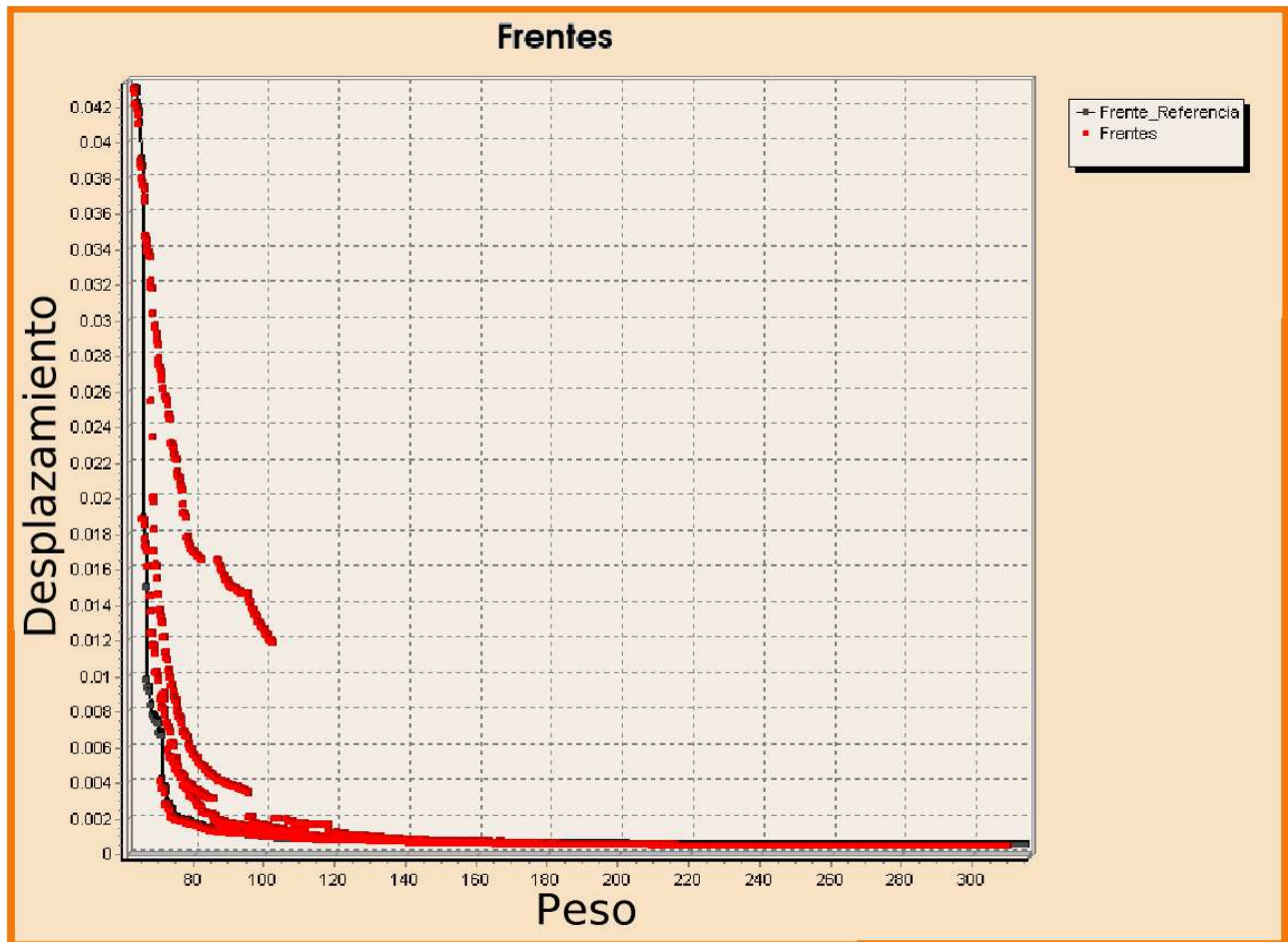


Figura 8.17: Frente de referencia (gris) y frentes de algunas corridas (rojo) del experimento 3.

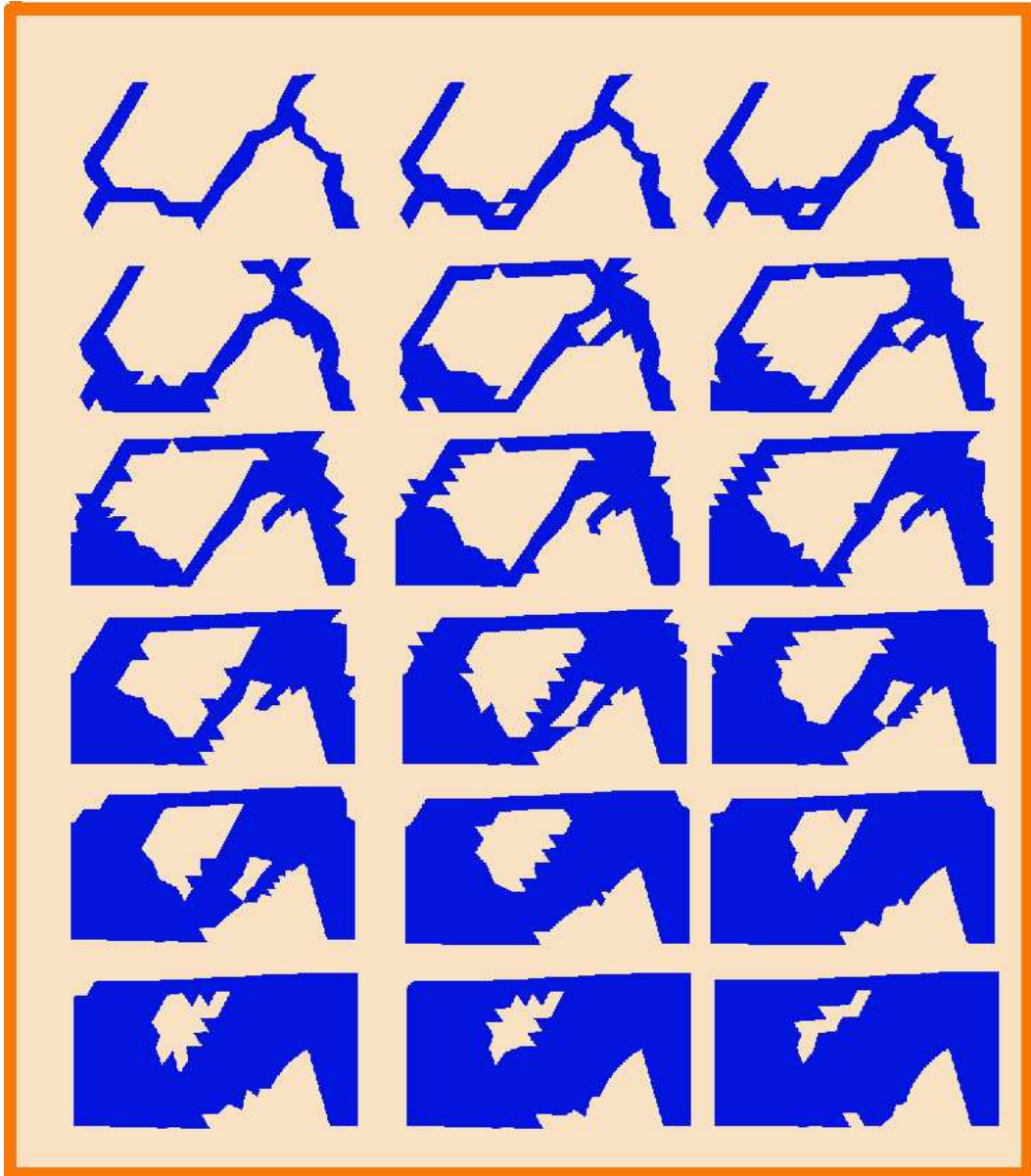


Figura 8.18: Algunas estructuras de una corrida típica del experimento 3.

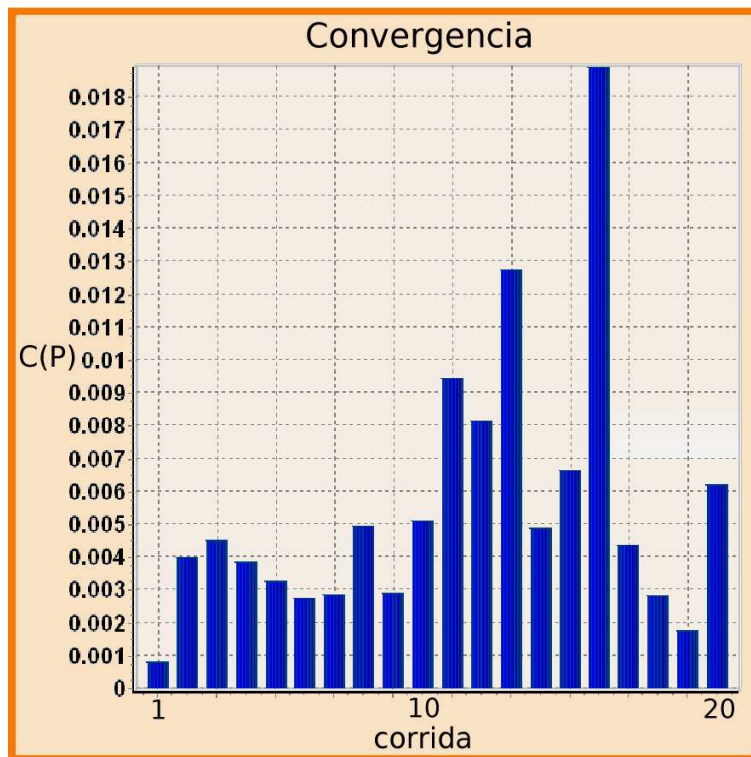


Figura 8.19: Métrica de convergencia de 20 corridas para el experimento 3.

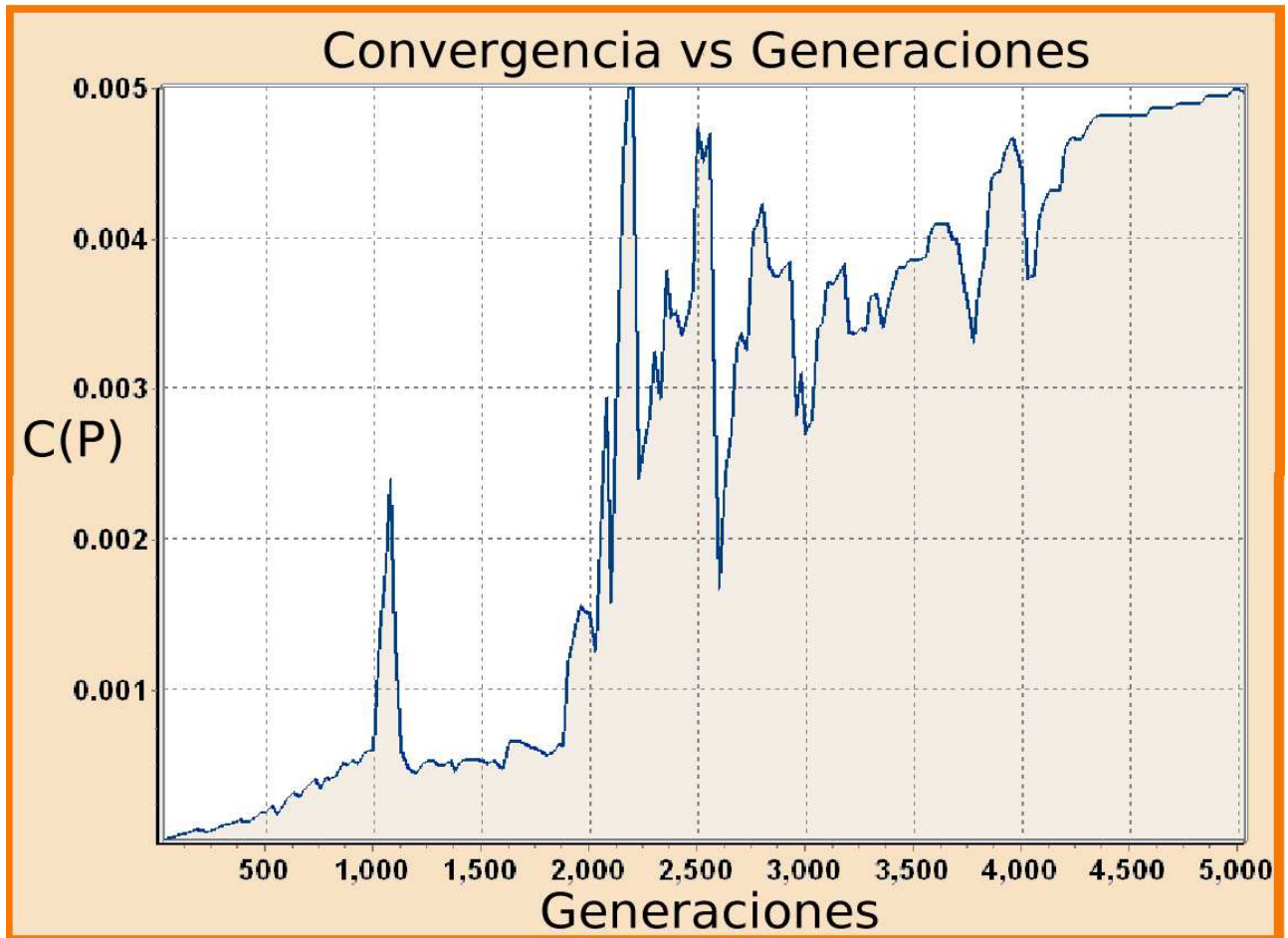


Figura 8.20: Gráfica de convergencia en una corrida típica del experimento 3.



Figura 8.21: Gráfica de convergencia para 15000 generaciones del experimento 3.

Capítulo 9

Comentarios y conclusiones

Se presentó un nuevo algoritmo para resolver el problema de optimización de formas, utilizando dominancia de Pareto para el manejo de las restricciones. El algoritmo regresa un conjunto de soluciones que satisfacen las condiciones de servicio y minimizan el peso y el desplazamiento nodal. El comportamiento mostrado por el algoritmo de acuerdo con las métricas utilizadas muestra una buena diversidad, esparciendo las soluciones en el frente; aunque las soluciones en el lado más pesado se ven más juntas que en el lado más ligero, se debe notar que la relación entre peso y desplazamiento no es lineal, por lo tanto, el quitar un elemento cuando la estructura tiene casi el mínimo peso causa un desplazamiento mucho mayor que quitar el mismo elemento en una estructura más pesada. El comportamiento de la convergencia es descendente a través de las generaciones, y todas las corridas llegan a estar muy cerca del frente de referencia.

La metodología utilizada, basada en la estimación de probabilidades marginales, representa una propuesta innovadora que se ha probado con un buen grado de éxito. Cabe mencionar que las propuestas existentes tratan en general con experimentos de menor dimensionalidad que los presentados en este trabajo, presentado los problemas de convergencia prematura y estancamiento en mínimos locales que se han mencionado anteriormente. Los algoritmos de estimación de distribución presentan de una forma más clara que los algoritmos genéticos el grado de exploración (en este caso dada por las probabilidades marginales cercanas a 0 o 1), proporcionando también de forma más transparente la vecindad o zona de exploración principal. Esto sucede de manera genérica, aún con otro tipo de distribuciones; por ejemplo para una distribución normal, la media proporciona un referencia de la vecindad que está siendo explorada y la desviación estándar una medida del grado de exploración, estas características no son muy claras en las estrategias evolutivas y los algoritmos genéticos. Además una de las suposiciones de los algoritmos de estimación de distribución, es que pueden encontrar cualquier solución dentro del espacio de búsqueda en cualquier momento, razón por la cual no se permite que la probabilidad de observar una solución específica sea igual a 1 o 0, debido a lo anterior la probabilidad de que se pueda observar cierto elemento de una muestra dada una distribución de probabilidad, puede ser calculado fácilmente.

Como se menciona en la sección 3.1, los algoritmos de estimación de distribución han dividido en dos partes el problema, por un lado la estimación del modelo gráfico de probabilidad y por otro la estimación de las probabilidades conjuntas y marginales. En el caso de este trabajo, se considera en un futuro cercano explorar la posibilidad de utilizar información de dependencias calculadas durante el mismo proceso. Sin embargo, se debe de considerar que aún para casos simples como el de probabilidades bivariadas, el cálculo de los valores de la probabilidad tanto marginales como conjuntas modifica la complejidad de n a n^2 . La posibilidad de extraer mayor información del conjunto de datos (configuraciones de estructuras que se les ha evaluado su aptitud o que pueden ser comparadas bajo algún criterio de optimalidad) debe ser una meta del trabajo futuro en esta área y en general en los problemas de optimización, principalmente los basados en heurísticas, que no necesitan del conocimiento explícito de las características de la(s) función(es) objetivo y que solamente entregan un conjunto

de datos (variables objetivo) y reciben un conjunto de valores que caracterizan a cada punto (valores de las funciones objetivo). El extraer la mayor cantidad de información para mejorar el proceso de generación de soluciones definirá la eficiencia y eficacia de la metodología para la solución de problemas de optimización.

La forma de actualización de las probabilidades hace que exista una conexión implícita entre los vectores de probabilidad, ya que al entrar o salir alguna estructura del *conjunto de trabajo de Pareto*, un vector de bits que actualiza al vector de probabilidad i actualizará al $i + 1$ o al $i - 1$; aunque este efecto no es demasiado fuerte como para impedir que formas diferentes pero óptimas puedan ser encontradas.

Las diferencias del algoritmo propuesto con los trabajos existentes básicamente son: un algoritmo de estimación de distribuciones que trata con un problema multiobjetivo con restricciones; y , la dimensionalidad de los problemas es alta en comparación con otras propuestas multiobjetivo para variables binarias.

Se logró desarrollar una estrategia para tratar con problemas de alta dimensionalidad a un costo factible; sin embargo, como se puede ver en la figura 8.17, algunas zonas del frente no llegaron a evolucionar hasta el frente de referencia. El trabajo futuro debe tratar de mejorar este acercamiento, tal vez reduciendo el total de soluciones encontradas y la dimensionalidad del problema para ciertas zonas del frente (fijar algunas variables).

Se debe notar que el problema de optimización depende de la malla, dado que una malla con pocos elementos (cientos) condiciona a la forma dentro de ciertas regiones con cambios bruscos en el desplazamiento y peso. Por lo tanto, para trabajos futuros se debe explorar la dependencia del problema de optimización a la malla, y cómo inicializar las probabilidades en una estrategia multimalla, considerando la dependencia de la malla y la reducción del espacio de búsqueda. Los resultados que se presentan muestran soluciones buenas a pesar de la malla no estructurada (otros autores han utilizado en una gran mayoría mallas estructuradas). Por último el trabajo futuro debe considerar también la reducción del costo computacional.

Bibliografía

- [1] S. Baluja. Population based incremental learning: A method for integrating genetic search based function optimization and competitive learning. *School of Computer Science Carnegie Mellon University, Pittsburgh, Pennsylvania 1523*, CMU-CS-94-163, 1996.
- [2] C. Chapman, K. Saitou, and M. Jakiela. Genetic algorithms as an approach to configuration and topology design. *Journal of Mechanical Design*, 116:1005–11, 1994.
- [3] K. Deb and S. Chaudhuri. Automated discovery of innovative designs of mechanical components using evolutionary multi-objective algorithms. *KanGal report*, (2004006), 2004.
- [4] K. Deb and T. Goell. Multiobjective evolutionary algorithms for engineering shape optimization. *KanGal report*, (2000003), 2000.
- [5] K. Deb and S. Jain. Running performance metrics for evolutionary multi-objective optimization. *KanGal report*, (2002004):1–18, 2000.
- [6] A. Farhang-Mehr and S. Azarm. Diversity assesment of pareto-optimal solution sets: An entropy approach. In *Proceedings of the 1990 World Congress on Computational Intelligence*, pages 723–728, 2002.
- [7] F. J. Grégoire Allaire and H. Maillot. Topology optimization for minimum stress design with the homogenization method. *Centre de Mathématiques Appliquées and Laboratoire MAPMO*, 2003.
- [8] C. Kane and M. Schoenauer. Topological optimum design using genetic algorithms. *Control and Cybernetics*, 25(5), 1996.
- [9] H. Li, Q. Zhang, E. Tsang, and J. Ford. Hybrid estimation of distribution algorithm for multiobjective knapsack problem. *Proceedings of the 4th European Conference on Evolutionary Computation in combinatorial Optimization*, 2004.
- [10] L. Malvern. *Introduction to the Mechanics of a Continuous Medium*. Prentice Hall Inc., Englewood Cliffs New Jersey, 1969.
- [11] J. Marroquín, F. V. M. Rivera, and M. Nakamura. Gauss-markov measure field models for low-level vision. *IEEE Trans. On PAMI*, 23(4):337–348, 2001.
- [12] H. Muhlenbein, T. Mahning, and A. O. Rodriguez. Schemata, distributions and graphical models in evolutionary optimization. *GMD- Forschungszentrum Informationstechnik*.
- [13] H. Muhlenbein and G. PaaB. From recombination of genes to the estimation of distributions i. binary parameters. *Parallel problem Solving form Nature*, PPSN(IV):178–187, 1996.
- [14] M. Pelikan, D. Goldberg, and C. Paz. Linkage problem, distribution estimation and bayesian networks. *IlliGal Report*, 1(98013), 1998.

- [15] M. Pelikan, D. Goldberg, and C. Paz. Boa: The bayesian optimization algorithm. In *Proceedings of the Genetic and Evolutionary Computation Conference*, 1999.
- [16] R. A. Richards. *Zeroth-order shape optimization utilizing a learning classifier system*. Copyright by Robert A. Richards. All Rights Reserved., 1995.
- [17] E. J. Sandgren, E. and J. Welton. Topological design of structural components using genetic optimization methods. In *Proceedings of the 1990 Winter Annual Meeting of the American Society of Mechanical Engineers*, pages 31–43. American Society of Mechanical Engineers, 1991.
- [18] O. Zienkiewicz and R. Taylor. *El Método de los Elementos Finitos*. Mc. Graw Hill-CIMNE., 1995.
- [19] E. Zitzler, M. Laumanns, and L. Thiele. Spea2: Improving the strength pareto evolutionary algorithm for multiobjective optimization. *Evolutionary methos for design, optimisation and control, CIMNE, Barcelona, Sapin 2002*, 2002.

Indice

- Actualización de la probabilidad, 17
- Agujeros pequeños, 13
- Alcance, 6
- Algoritmo propuesto, 15
- Algoritmos de Estimación de Distribución (EDA's), 8
- Algoritmos genéticos (GA), 8
- Allaire, 6
- Análisis del diseño, 5
- archivo fijo, 24

- Baluja, 1

- Chapman, 2
- conclusiones -experimento 1, 32
- conclusiones -experimento 2, 34
- condiciones de servicio, 5
- condiciones térmicas, 5
- Conjunto conocido de Pareto, 15
- Conjunto de trabajo de Pareto, 15
- convergencia -experimento 1, 32
- convergencia -experimento 2, 34
- convergencia prematura, 3
- convergencia, experimento 4, 36
- convergencia, métrica, 28
- corrosión y desgaste, 5
- costo computacional, 6, 7
- criterio de optimalidad, 6

- Deb, 2
- definición del problema, 10
- Descripción del algoritmo, 15
- discretización y representación, 11
- diseño, 5
- Diseño de detalle, 5
- Diseño conceptual, 5
- diversidad -experimento 1, 32
- diversidad -experimento 2, 34
- diversidad, métrica, 29

- efectividad, 31

- eficiencia, 6
- envejecimiento, 5
- esfuerzos térmicos, 5
- Estrategias Evolutivas (EE's), 8
- expansión del alcance, 7
- experimento 1, 31
- experimento 2, 34
- experimento 3, 35
- Experimentos, 31

- frente de referencia -experimento 2, 34
- fuerzas, 5

- Generación de la población, 16

- importacia económica, 4
- importancia científica, 4
- importancia social, 4
- Inicialización de la probabilidad, 15

- Kane, 2

- Li, 1

- Máximo esfuerzo Von Misses permisible, 12
- méticas de desempeño, 28
- métricas, 28
- Malvern, 12
- Mejoras del Algoritmo, 24
- metodología, 5
- Minimización del desplazamiento, 11
- Minimización del peso, 11
- Motivación, 4
- Muhlenbein, 1

- Número de objetos, 13

- objetivos de la optimización de formas, 5
- Optimización de dimensiones, 1
- Optimización de formas, 1, 4
- Optimización topológica, 1

- Paralelización del cómputo, 25

Pelikan, 1
problemas en CE, 2
proceso de diseño, 5
Programación Evolutiva (EP), 8
Programación genética (GP), 8

Regularización de la probabilidad, 17
requerimientos específicos, 5
Richards, 5

selección (modificación de la), 24
Selección y manejo de restricciones, 17

taza de aprendizaje, 17

varianza mínima, 24
varianza, cálculo de la, 18
vibraciones, 5

Zienkiewicz, 12
Zitzler, 24