Uso de la Técnica Multiobjetivo NPGA para el Manejo de Restricciones en Algoritmos Genéticos

Tesis que para obtener el grado de Maestro en Inteligencia Artificial
Presenta

Efrén Mezura Montes
Maestría en Inteligencia Artificial
Universidad Veracruzana

Asesor: Dr. Carlos A. Coello Coello
CINVESTAV-IPN

Revisor: Dr. Edgar Vallejo Clemente
ITESM Campus Edo. de México

Revisor: Dr. Arturo Hernández Aguirre
CIMAT A.C.

Septiembre, 2001

Resumen

En este trabajo se propone una técnica para manejo de restricciones en algoritmos genéticos. Esta propuesta se encuentra basada en una técnica multiobjetivo llamada Niched Pareto Genetic Algorithm (NPGA), la cual funciona mediante un proceso de selección basado en no dominancia. Originalmente, esta técnica utiliza nichos para mantener diversidad en la población.

Este trabajo demuestra la viabilidad de utilizar técnicas de optimización multiobjetivo para el manejo de restricciones en problemas con una sola función a optimizar. El principal problema de estas técnicas es la convergencia prematura debido a la fuerte presión de selección durante el proceso. Una manera de evitar este comportamiento es el uso de nichos, aunque su uso incrementa de forma importante su costo computacional.

Para manejar este problema se propone el uso de un parámetro llamado porcentaje de selección, el cual permite que un porcentaje de los individuos sea seleccionado de manera probabilística. De esta manera, se promueve la reproducción de individuos no factibles en el transcurso del proceso evolutivo y ,con ello, se aspira a lograr un acercamiento a la zona factible del espacio de búsqueda desde diversas direcciones, tratando de muestrearlo lo suficiente como para encontrar el óptimo global.

La presente propuesta no utiliza nichos y no requiere la definición de factores de penalización como en el caso de las funciones de penalización, que constituyen las técnicas más utilizadas para manejo de restricciones en algoritmos evolutivos.

Un conjunto de 9 funciones de prueba con distintas características fue utilizado para evaluar el desempeño y comportamiento de la técnica. Además, se realizó un análisis de sensibilidad con respecto a los parámetros propios del algoritmo.

Agradecimientos

A Dios, por regalarme la vida y permanecer siempre cerca.

A Margarita, por ser el amor, la razón y el motivo para ser mejor cada día.

A mis padres, Efraín y Guadalupe, por ser el ejemplo a seguir en esta vida y porque todo lo que soy se lo debo a ellos.

A mis hermanos, Irazema y José Miguel, por apoyarme y estar conmigo en todo momento.

A mi Abuelita Anselma, por consentirme y quererme tanto.

Al Doctor Carlos A. Coello Coello por su apoyo, tiempo, entusiasmo y dedicación para con este proyecto.

A todos mis amigos y compañeros de la MIA y el LANIA por su apoyo y amistad incondicionales.

Al Laboratorio Nacional de Informática Avanzada por las facilidades técnicas para la culminación de este trabajo.

Al CONACyT el apoyo otorgado mediante una beca para cursar la Maestría en Inteligencia Artificial de la Universidad Veracruzana/LANIA.

Este trabajo de tesis se derivó del proyecto CONACyT titulado "Estudio y Desarrollo de Técnicas Avanzadas de Manejo de Restricciones para Algoritmos Evolutivos en el Contexto de Optimización Numérica" (Ref. 32999-A), cuyo responsable es el Dr. Carlos A. Coello Coello.

Índice General

Índice de Figuras

Índice de Tablas

Introducción

El algoritmo genético (AG) ha demostrado ser una técnica que ofrece buenos resultados al aplicarse a problemas de optimización global. Sin embargo, para problemas que presenta el mundo real, el AG requiere de un mecanismo para incorporar en la función de aptitud las restricciones que normalmente acompañan a la función objetivo a optimizarse.

Existen varias propuestas para lidiar con este problema, de entre las que destaca el uso de funciones de penalización. La idea subyacente radica en extender la función de aptitud de manera que ésta se mantenga sin cambios si el individuo en cuestión es factible, o que se le "penalice" (o sea, que se disminuya la aptitud) en proporción a su violación de restricciones. Sin embargo, las técnicas de penalización requieren de la definición de factores *ad-hoc* que dependen de las características específicas del problema a resolverse. Lo que es más, el éxito o fracaso del proceso de búsqueda depende de la correcta elección de estos factores, así como de la forma que adopte la función de penalización.

Por otro lado, se han desarrollado recientemente diversas técnicas evolutivas para resolver problemas de optimización con objetivos múltiples, las cuales han demostrado tener un desempeño razonable en una amplia gama de problemas [20, 23]. Un mecanismo que resulta interesante y novedoso para el manejo de restricciones es precisamente el utilizar una técnica multiobjetivo de tal forma que las restricciones del problema y la función objetivo original sean tratados, todos, como objetivos a optimizarse. En este trabajo se presenta una propuesta de este tipo, utilizando la técnica multiobjetivo denominada *Niched Pareto Genetic Algorithm* (NPGA) [57].

El contenido de esta tesis se divide en seis capítulos: El capítulo 1 contiene una introducción a la computación evolutiva, incluyendo sus fundamentos y principales paradigmas. El capítulo 2 proporciona una revisión del estado del arte en lo que se refiere a técnicas para manejo de restricciones usadas con los algoritmos evolutivos, haciendo énfasis en las funciones de penalización. El capítulo 3 es una breve introducción a la optimización evolutiva multiobjetivo, señalando los principales enfoques y propuestas existentes. El capítulo 4 proporciona una revisión breve del trabajo previo relacionado con la propuesta de esta tesis. El capítulo 5 analiza a fondo el NPGA y describe en detalle la propuesta para extenderlo hacia el manejo de restricciones. Por último, el capítulo 6 contiene las descripciones de nueve funciones de prueba usadas para validar el método propuesto. Asimismo, se incluyen los resultados obtenidos y su comparación con otras técnicas reportadas previamente en la literatura. Se concluye esta tesis con una discusión de los resultados, las conclusiones y algunas de las rutas posibles de trabajo futuro.

Capítulo 1

Computación Evolutiva

La computación evolutiva es un conjunto de técnicas que basan su funcionamiento en modelar procesos evolutivos fundamentados en la supervivencia de los individuos más aptos en una población. Son utilizadas con frecuencia en problemas del mundo real donde la complejidad y la dimensionalidad son considerables.

En este capítulo se comentan los antecedentes históricos que dieron origen a estas técnicas, así como sus principales paradigmas y los términos asociados a ellas.

1.1 Antecedentes históricos

El hombre siempre ha buscado la manera de explicarse cómo se originó la vida. A lo largo de la historia han surgido varias teorías que proponen posibles respuestas. La primera, cronológicamente hablando, fue el *Creacionismo*, según la cual Dios creó a todas las especies y el hombre ocupa el nivel más alto dentro de ellas. Tal vez el primero en contradecir el *Creacionismo* fue George Louis Lecrerc quien especuló que las especies se originaron entre sí, e incluso observó similutudes entre los simios y el hombre.

Dos teorías siguieron a estas ideas: Jean Baptiste Lamarck [63] a inicios del siglo XIX enunció que las características adquiridas por un organismo durante su vida podrían ser transmitidas a sus descendientes. Por otro lado, Charles Darwin [31] en 1858 propuso que la evolución se genera con base en cambios aleatorios de caracteres hereditarios y a un proceso de selección natural.

El científico alemán August Weismann [63, 5] enunció a mediados del siglo XIX su teoría conocida como *Germoplasma*, de acuerdo a la cual la información hereditaria es transmitida por células llamadas “Germinales” y las llamadas “Somáticas” no tienen esta facultad. Además, realizó un experimento cortando la cola a ratas en 22 generaciones sin que las nuevas la perdieran. Con ello, perdieron valor las ideas de Lamarck.

Johann Gregor Mendel [5] enunció las leyes de la herencia como resultado de sus experimentos con chícharos. A través de dichas leyes estableció los conceptos de genes dominantes y recesivos y sus relaciones al momento de su transmisión de padres a hijos.

El planteamiento de que un individuo que tiene mayor capacidad de supervivencia tendrá más descendientes y de que sus genes ocurrirán con mayor frecuencia en la población fue propuesto por James Mark Baldwin en 1896. Con ello se sugiere que el aprendizaje tiene un efecto indirecto en el proceso de la evolución. Así se desecha la idea de que la transmisión de lo aprendido es directa como sugería Lamarck.

Conjuntando las ideas de Darwin, Weismann y Mendel surge el concepto de *Neo-Darwinismo* [36], que es el fundamento biológico de la computación evolutiva (ver figura 1.1). Según esta idea, la vida en este planeta se puede entender como un proceso que involucra los siguientes pasos:

1. Reproducción
2. Mutación
3. Competencia
4. Selección

Este proceso de evolución natural ha sido conceptualizado por algunos investigadores como un proceso de aprendizaje. En 1950, Alan Turing [141] por ejemplo, publicó un artículo relacionado con esta idea. Sin embargo, fue hasta fines de los 50’s en que comenzaron a publicarse una serie de trabajos

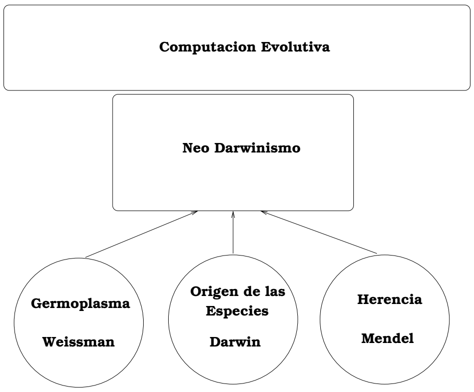

Figura 1.1: Neo-Darwinismo como base de la computación evolutiva

donde se simulaba el proceso evolutivo en una computadora, (Fraser y Box [41]).Una de las primeras aplicaciones de las técnicas evolutivas fue la reportada por Friedman [42], quien propuso la evolución de circuitos de control, similares a las actuales redes neuronales.

Uno de los primeros intentos en resolver problemas de optimización a través de la emulación de un proceso evolutivo fue el realizado por Hans J. Bremermann [13], quien utilizó un modelo muy similar al algoritmo genético actual.

Despues del auge de los 50's y 60's donde estas técnicas no fueron muy existosas surgieron los primeros éxitos reconocidos en la década de los 80's: Hicklin [53] y Fujiki [43] en teoría de juegos, la Programación genética de Koza [74], Thomas S. Ray con su sistema Tierra para simular ecosistemas [114], etc.

1.2 Fundamentos biológicos

1.2.1 Genética

Todos los organismos vivos tienen un material genético asociado llamado Ácido Desoxirribonucleico (ADN). El ADN esta compuesto por moléculas de ácido nucleico, de desoxirribosa y fosfato. Un **gene** es una sección de ADN con una función bioquímica definida. Además es propiamente la unidad de herencia. Un **cromosoma** es una cadena de ADN y es responsable de transmitir la información genética del organismo. El **Genoma** es el conjunto total de cromosomas y por ende de genes de un organismo.

Existen dos tipos de células, dependiendo del número de cromosomas que contienen: Una célula Haploide contiene un sólo conjunto de cromosomas, mientras que las células Dipliodes tienen un par de cromosomas donde existen genes dominantes y recesivos que pudieran ser distintos entre cada uno.

1.2.2 Selección Natural

El proceso evolutivo natural se basa en la existencia de **poblaciones** de **individuos**, donde se lleva a cabo la interacción entre ellos. Cada individuo tiene rasgos visibles, llamados en conjunto **fenotipo**; también contiene su composición genética, llamada **genotipo**. La relación consiste en que el genotipo da origen al fenotipo después de un proceso de desarrollo.

Una **especie** es una colección de criaturas vivientes que tienen características similares y que se reproducen entre sí. Su origen radica en el fenómeno conocido como especiación, donde un conjunto de individuos se aislan geográficamente de la población inicial.

Un **nicho ecológico** consiste en el ambiente limitado geográficamente donde una especie sobrevive. Es considerada una estrategia de supervivencia, siempre y cuando no exista otra especie que compita y con el paso del tiempo se extinga la más débil.

Las relaciones principales que se presentan entre los individuos son:

- **Reproducción**: que da origen a nuevos elementos dentro de la población; puede ser asexual o sexual; un progenitor y dos progenitores respectivamente. La probabilidad de que un individuo se reproduzca se conoce como aptitud.
- **Mutación**: donde nucleótidos individuales son cambiados o alterados de padres a hijos.
- **Migración**: donde se transfieren individuos, de una subpoblación a otra.

El proceso de **selección** consiste en escoger, con base en su aptitud a individuos para que se reproduzcan. Se distinguen dos tipos de selección: selección dura, donde sólo los mejores individuos se mantienen para procrear la futura generación, y la selección blanda, donde por medio de medios estocásticos se mantienen individuos poco aptos para reproducirse.

1.2.3 Fenómenos genéticos

En el transcurso del proceso evolutivo se suceden fenómenos a nivel genético como lo son:

- **Pleitropía**: Un gen afecta a varios rasgos a nivel fenotipo (formación de hemoglobina).
- **Poligenia**: Una característica fenotípica se determina por la acción de varios genes (color del cabello y piel).
- **Epístasis**: La presencia de un gen suprime el efecto de otro gen en otra posición.
- **Desvío Genético**: Cambio en la frecuencia de los genes en una población con el paso del tiempo y por efecto del azar. Su consencuencia directa es la pérdida de diversidad en la población. Cuando una subpoblación de una especie se separa geográficamente durante un periodo de tiempo significativo, sus genes divergirán.

1.3 Conceptos de Computación Evolutiva

La combinación de los conceptos asociados a la Genética (representación y relaciones entre individuos) con el proceso de selección natural (aptitudes, estrategias de supervivencia, etc.) discutidos anteriormente fundamentan el modelo general en el cual se basa la computación evolutiva.

Cinco son los elementos primordiales para modelar un proceso evolutivo en una computadora [85]:

1. Una representación adecuada para las soluciones potenciales del problema (individuos)

2. Una manera de generar una población inicial de individuos.

3. Un ambiente, normalmente modelado por una función de evaluación la cual comparará a los individuos de acuerdo a su aptitud.

4. Operadores de reproducción para generar nuevos individuos.

5. Valores para los parámetros propios de la técnica (probabilidades cruza y mutación, número de individuos).

De manera genérica, la solución de un problema utilizando Algoritmos Evolutivos (AE) se realiza de la siguiente manera:

1. Generar una **población aleatoria inicial** de ($n > 0$) individuos donde cada uno de ellos representa una solución potencial del problema.

2. **Seleccionar** a los individuos más aptos de acuerdo al resultado de evaluarlos con una **Función de Aptitud**.

3. Aplicar **operadores de reproducción** (cruza, mutación, etc) para generar nuevos individuos.

4. Iterar hasta satisfacer una condición de paro.

Para llevar a cabo este proceso se establece la siguiente terminología tomada de los fundamentos biológicos.

1.3.1 Representación de individuos

Un **cromosoma** es una estructura de datos que representa a un individuo de la población, normalmente un arreglo de enteros, donde cada posición es conocida como **gen**, el cual usualmente codifica el valor de un solo parámetro de la solución del problema. El valor que puede tomar cada gen se conoce como **alelo** (ver figura 1.2).

Un **esquema** representa un conjunto de cromosomas donde los valores de los genes incluyen estados donde no importa el valor del alelo que ahí se encuentre. De esta manera, conforman hiperplanos (ver figura 1.3).

A la codificación que representan tanto los cromosomas como los genes se denomina **genotipo**. El **fenotipo** consiste en decodificar los valores del

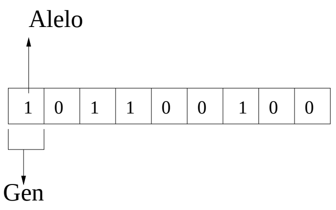

Figura 1.2: Cromosoma con representación binaria

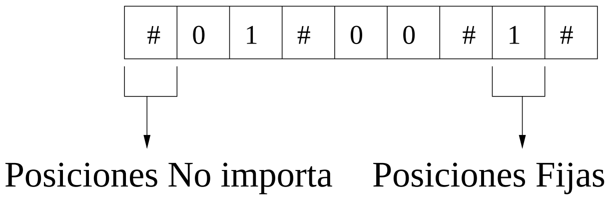

Figura 1.3: Esquema

cromosoma de manera que puedan sustituirse en la función de aptitud. Así se constituyen dos niveles de representación: genotípico y fenotípico. Si se utiliza condificación binaria ésta puede ser 0 ó 1 (ver figura 1.4).

La **aptitud** es el valor procesado por la función de aptitud y expresa qué tan bueno es un individuo (solución del problema) con respecto a los demás. A cada iteración del proceso se le conoce como **generación**.

El conjunto total de individuos que como tal conforman una población puede dividirse en **subpoblaciones**. De esta manera se modela la **especiación** natural, ya que sólo se cruzan individuos de la misma subpoblación. Sin embargo puede permitirse que algunos individuos transfieran sus genes de una subpoblación a otra. A esto se le conoce como **migración**. Cuando

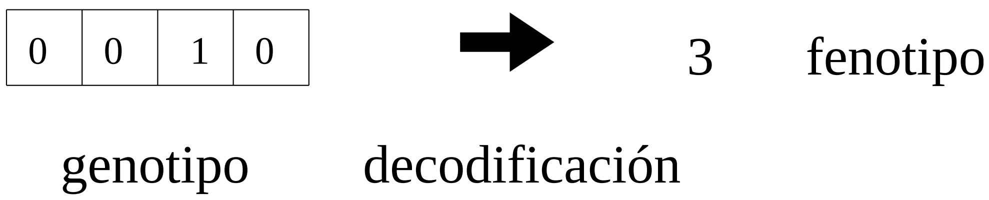

Figura 1.4: Genotipo y Fenotipo

un individuo puede reproducirse con cualquier otro, sin importar en qué subpoblación esté y que ello sólo dependa de su aptitud se dice que la **población es panmítica**.

Las representaciones más utilizadas son:

- Binaria.
- Binaria con Código Gray.
- Entera
- Real.
- Expresiones S en LISP (árboles)[1].
- Listas Binarias de Longitud variable.
- Híbridos.

1.3.2 Operadores de reproducción

Los operadores de reproducción también llamados Operadores Genéticos son aquellos que modifican la manera en que se transmite la información genética de padres a hijos. Se distinguen 3 categorías:

1. **Cruza**: Utiliza partes de ambos padres para generar un nuevo individuo. En optimización numérica se usa cruza:
    - De un punto [55].
    - De dos Puntos [68].
    - Uniforme [1, 134].

    Existen variantes que depende del tipo de representación utilizada.
2. **Mutación**: Forma un nuevo individuo a partir de modificaciones pequeñas al contenido genético de un solo padre.
3. **Reordenamiento**: Altera el orden de los genes de un individuo.

[1]Esta representación en forma de árbol da origen a la programación genética propuesta por Koza [74]

Existen dos subprocesos relacionados con los operadores genéticos. Uno es la explotación, la cual consiste en utilizar la información obtenida anteriormente para decidir cuál lugar es conveniente utilizar a continuación mediante movimientos finos. El operador de cruza permite explotar una zona prometedora del espacio de búsqueda del problema. Con ello se pueden encontrar óptimos locales en ese lugar.

La exploración consiste en encontrar zonas prometedoras del espacio de búsqueda e impedir el quedar atrapado en óptimos locales. La mutación permite dar saltos significativos en el espacio para lograr este cometido.

1.3.3 Selección

La selección de los individuos es de suma importancia en un algoritmo evolutivo ya que es la que guía la búsqueda hacia una solución buena. Existe una clasificación general de técnicas de selección.

- Selección proporcional [46]: Se eligen individuos de acuerdo a su contribución de aptitud con respecto al total de la población. Ejemplos:
  1. Ruleta.
  2. Sobrante Estocástico.
     - Con reemplazo.
     - Sin reemplazo.
  3. Universal Estocástica.
  4. Muestreo Determinístico.
- Selección Mediante Torneo: Propuesta por Wetzel [144].Se basa en comparaciones directas de aptitud entre individuos. Existen dos tipos:
  1. Determinística.
  2. Probabilística.
- Selección de Estado Uniforme: Propuesta por Whitley [145] utilizada en algoritmos genéticos no generacionales y aprendizaje incremental. Útil cuando los individuos resuelven el problema de manera colectiva.

1.3.4 Fenómenos y técnicas

Para evitar que en etapas avanzadas del proceso evolutivo se pierda diversidad en los individuos de la población debido a efectos probabilísticos existen técnicas como los **nichos**, emulando a los nichos naturales, que sirven para evitar que toda la población converja a una solución única.

Entre los fenómenos que ocurren a nivel de los cromosomas está la **epístasis**, que se refiere a la medida en que la contribución al valor de la aptitud de un gen depende del valor de otros genes. Esta es una medida también de la complejidad del problema a resolver para un algoritmo evolutivo (AE). Una epístasis moderada es un indicativo de que el problema tiene un grado de dificultad adecuado para resolverse con un AE.

Para asegurar que los miembros de una generación con las aptitudes más altas se preserven, (esto asumiendo que cada individuo tiene un tiempo de vida de una sola generación), se utiliza el **elitismo**. Así, los mejores individuos pasan intactos a la siguiente generación y no entran al proceso de selección.

1.4 Paradigmas

Aunque el proceso para solucionar un problema es similar en un algoritmo evolutivo se distinguen tres maneras de hacerlo. Son distintas en el nivel en el cual simulan la evolución, en sus operadores primarios y secundarios, además del nivel de representación en el cual trabajan.

1.4.1 Programación Evolutiva

Originalmente propuesta por Fogel [38] en 1960, donde remarca los nexos hereditarios y de comportamiento entre padres y sus descendientes. La adaptación se concibe en esta técnica como una forma de inteligencia.

El modelo de la Programación Evolutiva (PE) es a nivel de las especies, por lo tanto no existe un operador genético de cruza. La selección de individuos se realiza de manera probabilística y el operador genético principal es la mutación. Además opera a nivel fenotipo. En la propuesta original de la

PE, no se considera la auto-adaptación de sus parámetros.

El algoritmo básico de la PE es el siguiente:

1. Generar aleatoriamente una población inicial.

2. Aplicar mutación.

3. Calcular aptitud de todos los individuos de la población

4. Seleccionar mediante torneo (normalmente probabilístico) los individuos que sobrevivirán.

Las aplicaciones de la PE cubren las siguientes tareas genéricas [37, 38]:

- Predicción.
- Generalización.
- Juegos.
- Control Automático.
- Problema del Viajero.
- Planeación de Rutas.
- Diseño y Entrenamiento de Redes Neuronales.
- Reconocimiento de Patrones.

Se han realizado trabajos teóricos sobre la velocidad de convergencia para casos especiales como $(1+1)-PE$ y de convergencia global para $META-PE$.

1.4.2 Estrategias Evolutivas

Las Estrategias Evolutivas (EE) se desarrollaron en 1964 en Alemania con el objeto de resolver problemas de Aerodinámica con alto grado de complejidad. Los investigadores relacionados fueron Ingo Rechenberg, Hans-Paul Schwefel y Paul Bienert [127].

Las EE trabajan con una abstracción a nivel de los individuos, por lo que sí existe un operador de cruza, sexual (dos padres) o panmítica (un solo padre). Sin embargo, es un operador secundario. La mutación es el operador primario y se utiliza con valores Gaussianos. Estos valores son los que permiten que las EE sean auto-adaptativas, ya que el valor de mutación varía en el tiempo. El nivel de operación de las EE es fenotípico. Su selección es determinística y extintiva (los peores individuos tienen probabilidad cero de sobrevivir).

La versión original, llamada $(1 + 1) - EE$ no contempla el concepto de población. Existe un solo padre y a partir de él se genera un nuevo individuo mediante la siguiente expresión: $x^{t+1} = x^t + N(0, \bar{\sigma})$ donde t se refiere a la generación y $N(0, \bar{\sigma})$ es un vector de números Gaussianos independientes con media cero y desviación estándar $\bar{\sigma}$. Si el hijo es mejor en aptitud, se mantiene; de lo contrario se elimina.

La EE llamada $(\mu + 1) - EE$ ya utiliza una población. Hay μ padres y se genera un solo hijo, el cual puede reemplazar al peor padre. Por otro lado, Schwefel introdujo en 1975 el uso de múltiples hijos en las EE llamadas $(\mu + \lambda) - EE$ y $(\mu, \lambda) - EE$. En la primera, los mejores μ mejores individuos obtenidos de la unión de padres e hijos se preservarán (población traslapable). En la segunda, sólo los mejores μ hijos de la siguiente generación sobreviven (población no traslapable).

Las EE se han aplicado a problemas de [126]:

- Ruteo y redes.
- Bioquímica.
- Óptica.
- Diseño en Ingeniería.
- Magnetismo.

Existen pruebas teóricas de la velocidad de convergencia de una EE para casos especiales $(1 + 1) - EE$, $(1 + \lambda) - EE$ y de convergencia global para $(\mu + \lambda) - EE$.

1.4.3 Algoritmos Genéticos

Originalmente llamados "Planes Reproductivos" fueron concebidos por John H. Holland [55] a inicios de la década de los 60's. La motivación primordial fue el aprendizaje de máquina. Goldberg [45] los define como:

> Algoritmos de búsqueda basados en el mecanismo de selección natural y genética natural. Combinan la supervivencia de la más apta entre una estructura dada de cadenas con un intercambio aleatorio de información para conformar un algoritmo de búsqueda con algo del talento de la búsqueda humana

El Algoritmo Genético (AG) trabaja a nivel del genotipo y su operador primario es la cruza sexual ya que modela la evolución a nivel de los individuos; su operador secundario es la mutación. La manera de seleccionar individuos es probabilística basada en sus aptitudes. Normalmente no son auto-adaptativos.

El algoritmo básico de un AG es el siguiente [14]:

1. Generar aleatoriamente una población inicial.
2. Calcular la aptitud de cada individuo.
3. Aplicar operadores de reproducción para producir la siguiente generación (cruza y mutación).
4. Ciclar hasta satisfacer cierta condición.

Algunas aplicaciones de los AG's se muestran a continuación [45]:

- Optimización (estructural, de topologías, numérica, combinatoria).
- Aprendizaje de máquina (sistemas clasificadores).
- Bases de datos (optimización de consultas).
- Reconocimiento de patrones (imágenes)
- Generación de gramáticas (regulares, libres de contexto).

- Planeación de movimientos de robots.
- Predicción.

Entre los aspectos teóricos destacables de los AG's está el Teorema de los Esquemas, el cual da una idea del funcionamiento de este paradigma. Por otro lado, existe una prueba de convergencia del AG utilizando elitismo.

Para efectos de esta tesis el paradigma evolutivo utilizado para optimización global (no lineal) es el algoritmo genético, al cual se le agrega una técnica para manejo de restricciones basada en conceptos de optimización multiobjetivo.

Capítulo 2

Manejo de restricciones

Los algoritmos evolutivos como tales son técnicas de búsqueda sin restricciones. De ahí surge el interés por implementar mecanismos que les permitan trabajar con restricciones, tanto lineales como no lineales, de igualdad o desigualdad. Para lograrlo, se requiere incorporar la información de infactibilidad de un individuo (o su distancia a la zona factible) a su aptitud correspondiente, para que el proceso de selección preserve tanto a las soluciones factibles como aquellas soluciones prometedoras (o cercanas a ser factibles).

En este capítulo se profundiza en una de las técnicas más utilizadas: las funciones de penalización. En la parte final, se mencionan otros enfoques para el manejo de restricciones incorporados a algoritmos evolutivos, su filosofía, además de sus principales ventajas y desventajas.

2.1 Definición del problema

Refiriéndonos al dominio de la optimización numérica, un problema a resolverse se describe de la siguiente manera:

Encontrar $\mathbf{X}$ tal que optimice $f(\mathbf{X})$
Sujeta a:

$$g_i(\mathbf{X}) \leq 0, i = 1, \ldots, n \tag{2.1}$$

$$h_j(\mathbf{X}) = 0, j = 1, \ldots, p \tag{2.2}$$

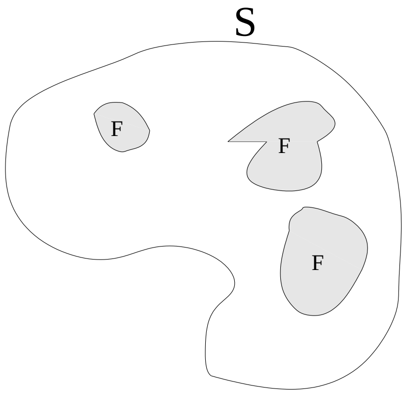

Figura 2.1: Espacio de búsqueda y zonas factibles

Donde $\mathbf{X}$ es un vector de soluciones $\mathbf{X} = [x_1, x_2, \ldots, x_r]^T$, n es el número de restricciones de desigualdad y p es el número de restricciones de igualdad. Ambos tipos de restricciones pueden ser tanto lineales como no lineales.

Denominaremos $\mathcal{F}$ a la zona factible del espacio de búsqueda $\mathcal{S}$ ($\mathcal{F} \subseteq \mathcal{S}$) (ver figura 2.1). Cuando una restricción de desigualdad cumple con $g_i(\mathbf{X}) = 0$ se dice que está activa en $\mathbf{X}$.

Existen problemas en los cuales el espacio de búsqueda puede ser muy grande y la zona factible muy pequeña y difícil de localizar (p. ej. disjunta). Es particularmente en este tipo de problemas en los cuales resulta más útil el uso de heurísticas tales como el algoritmo genético. De tal forma, las técnicas evolutivas suelen utilizarse en optimización no lineal y con problemas que tienen funciones objetivo no diferenciables, con ruido o dinámicas. Asímismo, suelen usarse en problemas en los que la forma del espacio de búsqueda y el óptimo global se desconocen.

2.2 Funciones de penalización

Este enfoque se basa en transformar el problema de optimización con restricciones a uno sin restricciones. Para ello, se agrega/quita una cierta cantidad del valor de la aptitud del individuo, de acuerdo a la cantidad de violación de las restricciones. Normalmente se consideran dos tipos de funciones de penalización: exterior e interior. En el primer caso, se inicia con una solución no factible y desde ese punto el proceso se moverá hacia la zona factible. Para el caso de las funciones de penalización interior el término de penalización se escoge de manera que su valor sea pequeño en puntos lejanos a los límites entre la zona factible y la no factible, y que tienda a infinito para puntos cercanos a ese límite. Una limitante de este último enfoque es que se requiere una solución factible inicial. Por ello, las funciones de penalización externas son más comúnmente utilizadas.

La fórmula general de la función de penalización externa es:

$$\phi(\mathbf{X}) = f(\mathbf{X}) \pm \left[\sum_{i=1}^{n} r_i \cdot G_i + \sum_{j=1}^{p} c_j \cdot L_j\right] \tag{2.3}$$

Donde $\phi(\mathbf{X})$ es la función objetivo expandida a optimizarse, G_i y L_j son funciones de las restricciones $g(\mathbf{X})$ y $h(\mathbf{X})$, respectivamente y r_i y c_j son constantes positivas llamados comúnmente "factores de penalización".

Tres criterios se manejan para definir la relación entre un individuo no factible y la region factible del espacio de búsqueda [115]:

- Un individuo se penaliza sólo por ser no factible.
- La penalización corresponde a su cantidad de no factibilidad.
- Se penaliza de acuerdo al esfuerzo de reparación del individuo.

El problema principal con las funciones de penalización en general consiste en elegir los valores adecuados de los factores de penalización para permitir que el algoritmo pueda dirigir la búsqueda hacia la zona factible y que la pueda recorrer lo suficiente como para llegar al óptimo global.

Se han generado varias alternativas para contrarrestar esta fuerte dependencia del valor de los factores de penalización. Entre las más destacadas se encuentran las técnicas descritas a continuación.

2.2.1 Pena de muerte

Es uno de los enfoques más socorridos y populares debido a su sencillez conceptual y de implemetación [4, 126]. Consiste en asignar una aptitud de cero a los individuos no factibles. No requiere de mayores cálculos. Se recomienda su uso para espacios de búsqueda convexos y de un tamaño considerable de la zona factible con respecto al tamaño total del espacio de búsqueda. Por otro lado, no utiliza información de los individuos no factibles, aunque se encuentren cerca de la zona factible.

Su principal desventaja se presenta en problemas complejos donde en la población inicial no existen individuos factibles. Esto conlleva a un estancamiento del algoritmo porque todos los individuos tendrán la misma aptitud. Por otro lado, este método sólo trabaja con restricciones de desigualdad, debido a la obvia dificultad de generar soluciones que satisfagan de manera exacta restricciones de igualdad.

2.2.2 Penalizaciones estáticas

Las implementaciones basadas en esta idea asumen que los factores de penalización permanecen sin cambios durante el proceso. Existen varias propuestas: Kuri [92] propone calcular la aptitud de un individuo de la siguiente manera:

$$fitness(\mathbf{X}) = \begin{cases} f(\mathbf{X}) & \text{Si la solución es factible} \\ K - \sum_{i=1}^{s}\left(\frac{K}{m}\right) & \text{En otro caso} \end{cases} \quad (2.4)$$

donde s es el número de restricciones satisfechas, m es el número total de restricciones, de igualdad y desigualdad y K es una constante ($1\text{x}10^9$). Cuando un individuo es no factible, no se computa su valor en la función de evaluación. Por otro lado, si el individuo viola un número s de restricciones, se penaliza igual que todos aquellos que violen también s restricciones sin importar si se encuentran cerca o lejos de la zona factible.

Esta técnica ha funcionado bien con una gama de funciones de prueba, pero en algunos casos requiere de generar la población inicial con otro AG al no existir individuos factibles en la población producida aleatoriamente.

Homafair, Lai y Qi [56] proponen establecer niveles de violación, donde los coeficientes de penalización se asignan a cada nivel de manera que un individuo se penaliza más si su índice de violación es mayor. Un individuo se evalúa con la siguiente expresión:

$$fitness(\mathbf{X}) = \begin{cases} f(\mathbf{X}) & \text{Si la solución es factible} \\ f(\mathbf{X}) + \sum_{i=1}^{m} R_{k,i} max[0, g_i\mathbf{X}]^2 & \text{En otro caso} \end{cases} \tag{2.5}$$

donde k son los niveles de penalización, $R_{k,i}$ son los coeficientes estáticos de penalización, m es el número total de restricciones y $f(\mathbf{X})$ es la función objetivo sin penalizar.

El excesivo número de parámetros que requiere este enfoque lo hace difícil de manejar para un número significativo de restricciones.

2.2.3 Penalizaciones dinámicas

El tiempo (o sea, la generación en que se encuentra el algoritmo evolutivo) influye en el cálculo de la penalización de un individuo en esta propuesta. Normalmente, en etapas avanzadas del proceso de búsqueda la penalización será más severa que en etapas iniciales, pues se presume que en esos instantes el algoritmo ya se encuentra dentro de la zona factible.

Joines y Houck [66] utilizan el número de generación para adaptar el valor del coeficiente de penalización.

$$fitness(\mathbf{X}) = f(X) + (C \cdot t)^{\alpha} \cdot SVC(\beta, \mathbf{X}) \tag{2.6}$$

donde C, α y β son constantes definidas por el usuario($C = 0.5$, $\alpha = 1$ ó 2 y $\beta = 1$ ó 2) y $SVC(\beta, \mathbf{X})$ se define como:

$$SVC(\beta, \mathbf{X}) = \sum_{i=1}^{n} D_i^{\beta}(\mathbf{X}) + \sum_{j=1}^{p} D_j(\mathbf{X}) \tag{2.7}$$

$$D_i(\mathbf{X}) = \begin{cases} 0 & g_i(\mathbf{X}) \leq 0 \\ |g_i(\mathbf{X})| & \text{en otro caso} \end{cases} \quad 1 \leq i \leq n \tag{2.8}$$

$$D_j(\mathbf{X}) = \begin{cases} 0 & -\epsilon \leq h_j(\mathbf{X}) \leq \epsilon \\ |h_j(\mathbf{X})| & \text{en otro caso} \end{cases} \quad 1 \leq j \leq p \tag{2.9}$$

Una de las desventajas de esta técnica es su sensibilidad a los parámetros C, α y β. Otra propuesta para modificar la penalización de un individuo dependiendo del número de generación es la de Kazarlis & Petridis [70]:

$$fitness(\mathbf{X}) = f(\mathbf{X}) + V(g) \times \left(A \sum_{i=1}^{m} (\delta_i \cdot w_i \cdot \Phi(d_i(S))) + B \right) \times \delta_s \quad (2.10)$$

donde A es un factor de severidad, m es el total de restricciones, δ_i es 1 si la restricción i es violada y 0 en otro caso. Se asignan pesos w a cada restricción.

La parte dinámica consiste en la función $V(g)$ la cual aumenta con respecto al número de generación en la que se encuentra el proceso. Los autores sugieren:

$$V(g) = \left(\frac{g}{G} \right)^2 \quad (2.11)$$

Donde g es el número de generación actual y G es el número total de generaciones.

Esta técnica requiere información del problema que afectará los valores de los parámetros; además, no ha sido probada con un conjunto significativo de funciones.

2.2.4 Penalizaciones adaptativas

En esta propuesta, se utiliza información del mismo proceso de búsqueda para actualizar los factores de penalización. Bean y Hadj-Alouane [49, 8] proponen evaluar al individuo mediante:

$$fitness(\mathbf{X}) = f(\mathbf{X}) + \lambda(t) \left[\sum_{i=1}^{n} g_i^2(\mathbf{X}) + \sum_{j=1}^{p} |h_j(\mathbf{X})|^2 \right] \quad (2.12)$$

donde $\lambda(t)$ se actualiza cada t generaciones de acuerdo con:

$$\lambda(t+1) = \begin{cases} (1/\beta_1) \cdot \lambda(t), & \text{Si ocurre caso \#1} \\ \beta_2 \cdot \lambda(t), & \text{Si ocurre caso \#2} \\ \lambda(t), & \text{en otro caso,} \end{cases} \quad (2.13)$$

$\beta_1, \beta_2 > 1$, $\beta_1 > \beta_2$, y $\beta_1 \neq \beta_2$ (para evitar ciclos).

El caso #1 ocurre cuando el mejor individuo en las últimas k generaciones fue siempre factible. El caso #2 sucede cuando este individuo nunca fue factible.

Como puede observarse, la penalización aumenta cuando no se encuentran individuos no factibles y disminuye cuando se encuentran de manera regular, buenos individuos factibles. Si el algoritmo presenta cambios intermitentes la penalización no cambia ya que esto sugiere que el algoritmo explora los límites entre la zona factible y la no factible. Su principal problema consiste en establecer los parámetros extras β_1, β_2 y λ_0.

2.2.5 Recocido simulado

Varios autores han propuesto el uso del Recocido Simulado [72] para controlar los factores de penalización, el cual se cambia cada determinado intervalo de tiempo. La penalización incrementa con respecto al tiempo (con la misma filosofía de penalizaciones dinámicas) y debe construirse una relación de las restricciones activas. Michalewicz y Attia [86, 84] proponen utilizar:

$$fitness(\mathbf{X}) = f(\mathbf{X}) + \frac{1}{2\tau} \sum_{i \in \mathcal{A}} \phi_i^2(\mathbf{X}) \tag{2.14}$$

donde τ representa el horario de enfriamiento [72], y m es el número total de restricciones.

$$\phi_i(\mathbf{X}) = \begin{cases} \max[0, g_i(\mathbf{X})] & \text{si } 1 \leq i \leq n \\ |h_i(\mathbf{X})| & \text{si } n+1 \leq i \leq m \end{cases} \tag{2.15}$$

Uno de los problemas de este enfoque es el establecer el intervalo de modificación de los factores de penalización (horario de enfriamiento).

2.2.6 Penalizaciones co-evolutivas

La idea de que el mismo proceso evolucione los factores de penalización fue utilizada por Coello [27, 23]. Propone el uso de dos subpoblaciones. En una evolucionarán las soluciones codificadas del problema y en la otra evolucionarán los factores de penalización los cuales multiplicarán a los siguientes factores: *coef* que es la suma de los valores con los que las restricciones han sido violadas por un individuo y *viol*, el cual almacena el número de restricciones que han sido violadas por ese mismo individuo. La aptitud de una solución se calcula con:

$$fitness(\mathbf{X}) = f(\mathbf{X}) - (coef \times w_1 + viol \times w_2) \tag{2.16}$$

Las subpoblaciones implican la definición de parámetros extra como lo son el número de generaciones para la subpoblación 1 y otro para la subpoblación 2, además del número de individuos para ambas subpoblaciones. Por otro lado, la implementación es más laboriosa ya que son realmente dos AG's ejecutándose de manera alterna. Por último, este enfoque sólo trabaja con restricciones de desigualdad.

2.2.7 Algoritmo genético segregado

Consiste en utilizar dos funciones de penalización y por ende, dos poblaciones. La idea subyacente es tener un balance entre las penalizaciones fuertes y las moderadas. Fue propuesto por Le Riche et al. [116]. Los padres de la siguiente generación se eligen de entre los mejores individuos de cada subpoblación. Así, se busca mantener diversidad ya que se espera se combinen individuos factibles con no factibles. Se utiliza selección proporcional con jerarquías lineales con el objeto de disminuir la presión de selección.

Su principal desventaja es la de escoger los dos factores de penalización (tanto el fuerte como el moderado) para ambas subpoblaciones.

2.2.8 Penalización basada en factibilidad

Propuesta por Deb y Agrawal [33]. Consiste en evaluar a un individuo de acuerdo con la siguiente expresión:

$$fitness_i(\mathbf{X}) = \begin{cases} f_i(\mathbf{X}) & \text{Si la solución es factible} \\ f_{peor} + \sum_{j=1}^{n} g_j(\mathbf{X}) & \text{En otro caso} \end{cases} \tag{2.17}$$

donde f_{peor} es el valor de la función objetivo de la peor solución factible de la población. Si no existen soluciones factibles, f_{peor} se hace 0.

Utiliza torneo binario como técnica de selección y agrega los siguientes criterios:

- Una solución factible siempre es preferible a una no factible.
- Entre dos soluciones factibles, aquella con mayor valor (o menor, según sea el caso) en su función objetivo es preferible.
- Entre dos soluciones no factibles, aquella que viole el menor número de restricciones es preferible.

Las desventajas de esta técnica es el requerir el uso de nichos [33] y porcentajes de mutación altos para mantener diversidad.

2.3 Otras técnicas

Coello [22] incluye a las funciones de penalización en su taxonomía y agrega las siguientes:

1. Mantener una población factible mediante representaciones especiales y operadores genéticos.
2. Algoritmos de reparación.
3. Separación de objetivos y restricciones.
4. Enfoques híbridos.

2.3.1 Representaciones y operadores especiales

Como alternativa para lidiar con determinado tipo de problemas donde la representación binaria no es conveniente se han propuesto otras maneras de representar individuos y por ende, se han generado nuevos operadores que emulen la función de los operadores tradicionales.

Un cambio en la representación provoca que se simplifique la forma del espacio de búsqueda y los operadores especiales permiten preservar la factibilidad de las soluciones en el proceso evolutivo. Entre las más destacadas se tienen:

- **Aplicaciones de Lawrence Davis** [32]: En su *Handbook of Genetic Algorithms* propone varios ejemplos de algoritmos evolutivos con representaciones y operadores especiales para resolver problemas del mundo real. Este cambio es útil en la mayoría de los casos, pero no es fácil generalizarlo para problemas similares.

- **Llaves aleatorias**: Propuestas por James C. Bean [6, 7], consisten de una representación llamada "Random Key Enconding" la cual se utiliza para eliminar la necesidad de operadores especiales de cruza y mutación codificando las soluciones con números aleatorios los cuales serán las llaves para decodificar sus valores. La propuesta es interesante, pero los resultados obtenidos con ella no han sido muy exitosos.

- **GENOCOP**: GEnetic algorithm for Numerical Optimization for COnstrained Problems propuesto por Michalewicz [85] asume un(os) punto(s) inicial(es) para iniciar la búsqueda y sólo admite restricciones lineales. Los resultados obtenidos con él han sido exitosos.

- **Algoritmo Genético Consistente en Restricciones**: Propuesto por Kowalczyk [73] para acotar el espacio de búsqueda previniendo la instanciación de variables que no satisfacen a las restricciones. Por lo tanto, sólo se producen soluciones factibles.

- **Localización de los límites de la región factible**: Su motivación radica en que en optimización no lineal se suelen tener restricciones activas en el óptimo global, razón sugerente para explorar el límite entre la zona factible y la no factible del espacio de búsqueda. Varios autores han publicado propuestas sobre este enfoque [88, 124, 123, 44].

- **Decodificadores**: Consiste en que un cromosoma proporcione la manera de construir una solución factible. Cada decodificador impone una relación entre una solución factible y una solución decodificada. Propuesta por Koziel y Michalewicz [75, 76].

2.3.2 Algoritmos de reparación

La idea de tomar una solución no factible y hacerla factible mediante un operador especial ha sido utilizada ampliamente. En algunos enfoques, la solución "reparada" sustituye en la población a su original no factible y en otros sólo se utiliza con fines de evaluar la aptitud. Para ciertos problemas, esta opción ha presentado buenos resultados, como en GENOCOP III [87], aplicaciones para robótica de Xiao et al. [90, 148, 149], y en el trabajo de

Orvosh y Davis [96, 97] que proponen cierto porcentaje (5 %) de sustitución de individuos “reparados” por su correspondiente infactible.

El principal problema con este enfoque es que el proceso de reparación de individuos depende directamente de la naturaleza del problema. Por otro lado, puede caerse en la situación de que el algoritmo de reparación resulte más costoso que el mismo algoritmo evolutivo de búsqueda.

2.3.3 Separación de restricciones y objetivos

Entre los enfoques más destacados se tienen los siguientes:

- **Co-Evolución**: Propuesta por Paredis [103] consiste en tener dos poblaciones, una que contiene las restricciones a satisfacer y la segunda contiene soluciones potenciales al problema (no necesariamente factibles). Además existe una relación de dependencia de la presión de selección de una población con respecto al valor de aptitud de los miembros de la otra población. Con ello se logra incrementar la aptitud de aquellas restricciones que son difíciles de satisfacer para que el proceso se centre en ellas. Los resultados publicados son bastante buenos. Sin embargo, como se requiere de una memoria de las relaciones entre individuos de ambas poblaciones en algo llamado *encounters* donde se evalúan individuos de cada una de ellas, se requiere de estructuras de datos con mayor grado de complejidad que las manejadas normalmente en un algoritmo evolutivo.

- **Superioridad de Puntos factibles**: Propuesta originalmente por Powell y Skolnick [108]. Incorporan una regla heurística para procesar soluciones no factibles con el objeto de valuar como superiores a las soluciones factibles sobre las infactibles. Entre sus principales desventajas se encuentra la falta de diversidad durante el proceso.

- **Memoria Conductista**: Schoenauer y Xanthakis [125] proponen una manera de satisfacer secuencialmente, una a una, las restricciones del problema. De manera implícita se utiliza *pena de muerte* ya que las soluciones que en el proceso no cumplan con al menos una restricción se eliminan de la población.

- **Técnicas de Optimización Multiobjetivo**: Se propone la redefinición del problema de optimización simple a uno multiobjetivo. Varias propuestas han surgido con esta idea; entre ellas están COMOGA de Surry [133, 132], donde se combina la técnica multiobjetivo VEGA [122] y el Pareto ranking [45] para manejar restricciones. Camponogara y Talukdar [15], redefinen el problema para utilizar dos objetivos. Jiménez y Verdegay [64] proponen un enfoque min-max con comparaciones simples para el proceso de selección. Coello [26] utiliza a VEGA en un enfoque donde divide a la población en varias subpoblaciones; esto se hace en cada generación con el objeto de que cada subpoblación se asocie con una restricción del problema y ayude a minimizar la cantidad con la cual es violada. Además, una de las subpoblaciones se encargará de evaluarse con la función objetivo. El principal problema con estos enfoques es la pérdida de diversidad que suele caracterizar a la optimización multiobjetivo.

2.3.4 Métodos híbridos

Consisten en acoplar un algoritmo evolutivo con otro método de optimización numérica. Por ejemplo:

- **Multiplicadores de Lagrange**: Ofrecen buenos resultados, pero agregan más parámetros que deben ser ajustados *a priori* [2, 93, 71].

- **Evolución aleatoria**: Combinación de búsqueda evolutiva con una técnica de programación matemática sin restricciones (simplex). Se reduce a un algoritmo de reparación [10].

- **Lógica difusa**: Reemplaza la definición de las restricciones a restricciones lógicas. Reemplaza a los factores de peso en la función objetivo por grados de satisfacción de restricciones [78].

- **Sistema Inmune**: Modela la actividad del sistema inmune biológico. Un conjunto de anticuerpos evolucionan para contrarrestar el efecto de antígenos. En principio se utilizaron cadenas binarias, pero ya hay trabajos donde se utilizan otras representaciones [29, 50, 51].

- **Algoritmos Culturales**: Utiliza mecanismos sofisticados de reparación de individuos. Otras alternativas sugieren el uso de células de creencia *belief-cell* para adquirir, almacenar e integrar el conocimiento de las restricciones no lineales [65, 18].

- **Colonia de Hormigas**: Consiste en una meta-heurística diseñada para problemas complejos de optimización combinatoria que se ha extendido a optimización numérica. La idea subyacente de esta técnica es el comportamiento de las hormigas, las cuales depositan una sustancia química la cual influye en el comportamiento de las demás. Bilchev y Parmee utilizan el concepto de *Comida Inaceptable*, la cual irá desapareciendo conforme avanza el proceso y el algoritmo converge. Esta técnica es sencilla de implementar, pero requiere muchos parámetros [11, 12].

Capítulo 3

Optimización multiobjetivo

Generalmente los problemas que presenta el mundo real tienen un nivel de complejidad considerable. Muchas veces esa complejidad radica en que consisten de varios objetivos y restricciones que deben satisfacerse. Los algoritmos evolutivos son especialmente aplicables para resolver este tipo de problemas ya que trabajan de manera simultánea con un conjunto de soluciones las cuales pueden evolucionar y obtener aquellas que sean consideradas como aceptables por parte del diseñador, puesto que aún no existe una definición de "óptimo" como es el caso de la optimización con un solo objetivo.

En este capítulo se define un problema multiobjetivo, se establecen los conceptos de "óptimo de Pareto" y "Frente de Pareto" y se describen las principales técnicas evolutivas para optimización multiobjetivo reportadas en la literatura.

3.1 Definición del problema

Un problema de optimización multiobjetivo se establece como el proceso de encontrar [102]:

> Un vector de variables de decisión que satisfaga restricciones y optimice una función de vector cuyos elementos representan las funciones objetivo. Estas funciones conforman una descripción matemática de criterios de desempeño los cuales usualmente se encuentran en conflicto uno con otro. De ahí que el término "optimizar" signifique el encontrar una solución que proporcione

valores para todas las funciones objetivo que sean aceptables para el diseñador.

Esto se traduce de manera formal a:

Encontrar un vector $\bar{x}^* = [x_1^*, x_2^*, \ldots, x_n^*]^T$ el cual debe satisfacer las m restricciones de desigualdad:

$$g_i(\bar{x}) \geq 0, i = 1, 2, \ldots, m \tag{3.1}$$

las p restricciones de igualdad:

$$h_i(\bar{x}) = 0, i = 1, 2, \ldots, p \tag{3.2}$$

y optimiza la función vector:

$$\bar{f}(\bar{x}) = [f_1(\bar{x}), f_2(\bar{x}), \ldots, f_k(\bar{x})]^T \tag{3.3}$$

Donde $\bar{x} = [x_1, x_2, \ldots, x_n]^T$ es un vector de variables de decisión.

Esto quiere decir que se requieren determinar los números que pertenecen a la zona factible $\mathcal{F}$ (o sea, que satisfacen (3.1) y (3.2)) y que obtienen el valor óptimo para todas las funciones objetivo de (3.3).

3.1.1 Óptimo de Pareto

El concepto de óptimo de Pareto fue formulado por Vilfredo Pareto en el siglo XIX [104]. Este se considera el origen de toda la investigación en optimización multiobjetivo. Se dice que un punto $\bar{x}^* \in \mathcal{F}$ es óptimo de Pareto si para cada $\bar{x} \in \mathcal{F}$,

$$\bigwedge_{i \,\in\, I} (f_i(\bar{x}) = f_i(\bar{x}^*)) \tag{3.4}$$

o, existe al menos algún $i \in I$ (donde I representa el conjunto de funciones objetivo a optimizarse) que

$$f_i(\bar{x}) > f_i(\bar{x}^*) \tag{3.5}$$

Un punto del espacio de búsqueda $\bar{x}^*$ se considera óptimo de Pareto si no existe ningún vector factible que pueda decrementar algún valor de un objetivo sin aumentar de manera simultánea el valor de al menos otro objetivo,

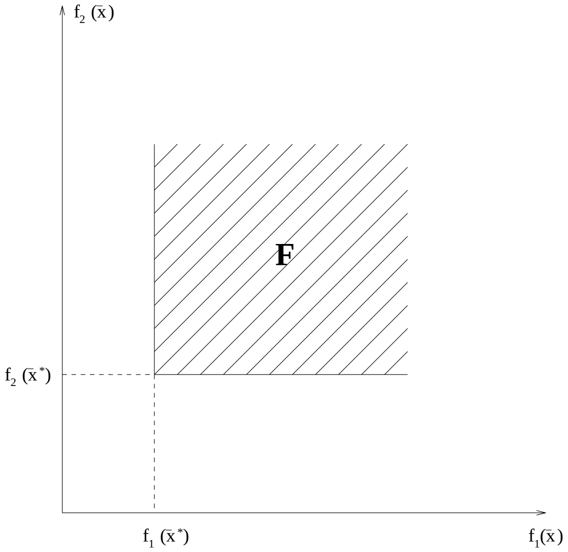

Figura 3.1: Solución ideal donde se tiene el óptimo en un punto (óptimo de Pareto) para el caso de dos funciones objetivo.

suponiendo minimización (ver figura 3.1).

3.1.2 Frente de Pareto

En la mayoría de los casos, no se obtienen soluciones únicas, sino un conjunto de ellas llamadas *no inferiores o no dominadas* cuya ubicación se encuentra en los límites de la zona factible (ver figura 3.2). Normalmente es imposible encontrar una expresión analítica para esa línea o superficie que contiene los puntos correspondientes a las soluciones no dominadas. El procedimiento que normalmente se sigue es procesar los puntos $\mathcal{F}^k$ y sus correspondientes $f(\mathcal{F}^k)$. A este conjunto de soluciones se les conoce como *Frente de Pareto.*

Un vector $\bar{x}^*$ es una *solución débilmente no dominada* si no existe $\bar{x} \in \mathcal{F}$ tal que

$$f_i(\bar{x}) < f_i(\bar{x}^*), i = 1, \ldots, n \tag{3.6}$$

Un vector $\bar{x}^*$ es una *solución fuertemente no dominada* si no existe $\bar{x} \in \mathcal{F}$

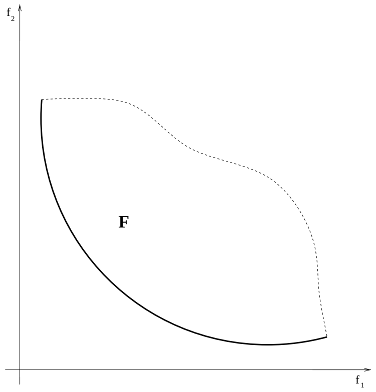

Figura 3.2: Frente de pareto (remarcado) para el caso de dos funciones objetivo.

tal que

$$f_i(\bar{x}) \leq f_i(\bar{x}^*), i = 1, \ldots, n \tag{3.7}$$

y para al menos un valor de i

$$f_i(\bar{x}) < f_i(\bar{x}^*), i = 1, \ldots, n \tag{3.8}$$

Por lo tanto, una solución fuertemente no dominada es también una solución débilmente no dominada, pero no necesariamente se cumple la conversa.

Para efectos de este trabajo se utilizará el concepto de no dominancia con respecto a soluciones fuertemente no dominadas a la cual se les llamará simplemente soluciones no dominadas.

De manera gráfica se pueden visualizar ambos tipos de soluciones para el caso de dos funciones objetivo [22]. La posición de los puntos de las soluciones fuertemente no dominadas corresponden a la llamada *curva mínima fuerte* y la posición de los puntos de las soluciones débilmente no dominadas se ubica sobre la *curva mínima débil* (ver figura 3.3).

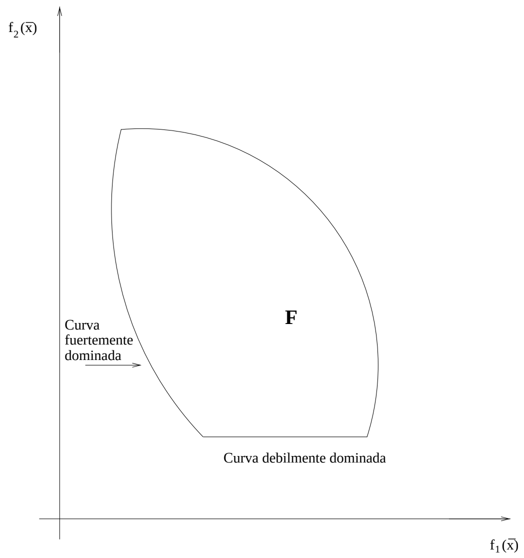

Figura 3.3: Curvas débil y fuertemente dominadas para el caso de dos funciones objetivo.

3.2 Técnicas para preservar diversidad

Un algoritmo genético (AG) tiende a converger a una solución única debido al tamaño finito de la población y a errores estocásticos asociados a los operadores genéticos [68], lo que es deseable cuando se quiere alcanzar el óptimo global o algo cercano a él. Sin embargo, en optimización multiobjetivo se requiere encontrar todo el frente de Pareto consistente en un conjunto de soluciones. Para mantener la diversidad de soluciones durante el proceso evolutivo (es decir, para evitar que el AG converja a una solución única) han surgido varias propuestas.

Holland [54] propone el uso de un operador de cercanía y similitud (*crowding*) el cual fue implementado por DeJong [68]. Él utilizó esta idea para permitir que un descendiente de un individuo reemplace a aquél con el cual tenga mayor grado de similitud a nivel genotípico.

Goldberg y Richardson [48] dividen a la población en subpoblaciones de acuerdo a su similitud, ya sea en el espacio genotípico (codificación de las variables de decisión), o en el espacio fenotípico (valores de las variables de decisión) lo que involucra la existencia de un parámetro que establezca el grado de similitud entre los individuos, todo esto asumiendo una representación binaria. Los individuos que pertenecen a una misma subpoblación (o nicho[1]) compartirán su aptitud mediante una función de distribución, de manera que disminuya. Con ello se premia a aquellos individuos que sean distintos.

Existen también otras propuestas. Por ejemplo, Smith, Forrest y Perelson [128] utilizan una emulación del sistema inmune humano para mantener diversidad sin la necesidad de establecer una función de similitud. Ver Mahfoud [83] para una revisión más detallada de técnicas para mantener diversidad.

3.3 Técnicas evolutivas para optimización multiobjetivo

Coello [22, 23] propone una taxonomía para agrupar las técnicas evolutivas para optimización multiobjetivo:

- Funciones de agregación.
- No basadas en la noción de óptimo de Pareto
- Basadas en el óptimo de Pareto

A continuación se describen las características generales de cada grupo de técnicas y se mencionan las propuestas más representativas.

3.3.1 Funciones de agregación

La manera más intuitiva de manejar varios objetivos es la de combinarlos en uno solo y con ello obtener un único valor de aptitud para que el algoritmo evolutivo pueda operar de manera usual. La manera de unirlos puede ser mediante una simple operación aritmética (p. ej. suma o multiplicación).

[1]Estos “nichos” están inspirados en los nichos biológicos (lugares donde individuos similares comparten recursos)

Esta idea presenta ciertos problemas: se requiere tener información escalar del rango de los objetivos para evitar que uno domine a los demás. Este proceso es (especialmente para problemas de la vida real) muy complicado o costoso y en ocasiones no se puede llevar a cabo.

Por otro lado, este método en su sencillez encuentra la eficiencia, porque no requiere de la decisión del diseñador para determinar si una solución es aceptablemente buena. Además, si el algoritmo logra optimizar la función de aptitud resultante, al menos habrá alcanzado un óptimo local.

Entre los técnicas más representativas de este grupo se encuentran:

Suma ponderada

Consiste en sumar todas las funciones objetivos utilizando diferentes pesos como coeficientes para cada una. El problema se define como:

$$minimizar \sum_{i=1}^{k} w_i f_i(\bar{x}) c_i \tag{3.9}$$

donde $w_i \geq 0$ son los coeficientes que representan la importancia de cada función objetivo (de las k existentes) y c_i son factores constantes que escalan los objetivos. Esto con el objeto de mantener valores numéricos similares para todas las funciones objetivo. Los pesos se normalizan de manera que:

$$\sum_{i=1}^{k} w_i = 1 \tag{3.10}$$

Su principal ventaja es su eficiencia computacional y su utilidad para generar una solución inicial para otras técnicas. Su desventaja radica en la necesidad de que el usuario defina los pesos para cada objetivo. Además, el uso de una combinación lineal de pesos no permite generar ciertas porciones de los frentes de pareto no convexos [117].

Algunas aplicaciones de este enfoque se describen en los trabajos de Syswerda y Palmucci [135], Jacob et al. [62], Jones et al. [67], Wilson & Macleod [147], Liu et al. [81], y Yang y Gen [150].

Programación por metas

En esta propuesta hecha por Charnes y Cooper [16] e Ijiri [60] el diseñador debe asignar objetivos o metas que él desee alcanzar para cada función objetivo. Estos valores se incorporan al problema como restricciones adicionales. La función objetivo consiste en minimizar las desviaciones absolutas de las metas de las funciones objetivos. El problema se plantea como:

$$minimizar \sum_{i=1}^{k} |f_i(\bar{x}) - T_i|, \quad \text{sujeta a } \bar{x} \in \mathcal{F} \tag{3.11}$$

Donde T_i es la meta propuesta por el diseñador para la i-ésima función objetivo $f_i(\bar{x})$ y $\mathcal{F}$ representa la zona factible del espacio de búsqueda. La forma de trabajar de este modelo es minimizar la suma de valores absolutos de las diferencias entre las metas y los valores alcanzados hasta un momento determinado. Una generalización de esta técnica es la *programación por metas generalizada* [58, 59] también conocida como *optimización de vector de metas* [19].

Sus ventajas son su eficiencia computacional pero tiene problemas si el diseñador no asigna correctamente los valores de las metas. Además, si la zona factible es difícil de alcanzar, la técnica se volverá ineficiente.

Entre las aplicaciones de esta técnica se encuentran las de Sandgren [120] y la de Wienke et al. [146].

Alcance de metas

En esta propuesta el diseñador define un vector de pesos $w_1, w_2, \ldots, w_k$ que refleja el alcance de las metas además del vector de metas $b_1, b_2, \ldots, b_k$ para las funciones objetivo $f_1, f_2, \ldots, f_k$. El problema se plantea de la siguiente forma:

$$minimizar \;\; \alpha \tag{3.12}$$

sujeta a:

$$\begin{aligned} g_j(\bar{x}) \leq 0; \quad j = 1, 2, \ldots, m \\ b_i + \alpha \cdot w_i \geq f_i(\bar{x}); \quad i = 1, 2, \ldots, k \end{aligned} \tag{3.13}$$

donde α es una variable escalar sin restricción de signo y los pesos $w_1, w_2, \ldots, w_k$ están normalizados de manera que:

$$\sum_{i=1}^{k} |w_i| = 1 \tag{3.14}$$

El valor óptimo de α indicará si las metas son alcanzables o no. Si el valor de α es negativo implica que la meta propuesta por el diseñador es alcanzable y que se obtendrá una solución mejorada. Si $\alpha > 0$, entonces la meta es inalcanzable.

Su principal ventaja es su sencillez y eficiencia computacional. Sin embargo, produce una presión de selección que no siempre causa convergencia hacia buenas soluciones compromiso, lo cual fue reportado por Wilson y Macleod [147].

Restricción ε

La idea subyacente de esta propuesta hecha por Ritzel y Wayland [117] es la minimización de un solo objetivo y tomar a los otros como restricciones limitadas por niveles conocidos como ε_i, los cuales indican el valor que cada objetivo i no debe exceder. Estos niveles se modifican con el objeto de generar el frente de Pareto. El método se define como:

Encontrar el mínimo de la r-ésima función objetivo, (p. ej. encontrar $\bar{x}^*$) tal que

$$f_r(\bar{x}^*) = \underset{x \,\in\, \mathcal{F}}{min} \; f_r(\bar{x}) \tag{3.15}$$

sujeta a

$$f_i(\bar{x}) \leq \varepsilon_i \;\;,\; i = 1, 2, \ldots, k \quad i \neq r \tag{3.16}$$

El proceso se repite para varios valores de ε_i. La búsqueda termina cuando el diseñador encuentra una solución satisfactoria. Sin embargo puede requerirse repetir todo el proceso para distintos valores de r.

Esta técnica es popular para resolver problemas multiobjetivo, pero su codificación consume mucho tiempo. Además, se vuelve difícil de manejar cuando el problema a resolver tiene demasiados objetivos.

De las aplicaciones más destacadas de esta propuesta se tienen las de Quagliarella y Vicini [109], Ranjithan et al. [110] y Loughlin and Ranjithan [82].

3.3.2 No basadas en la noción de óptimo de Pareto

Otra manera de manejar múltiples objetivos se basa en el manejo de ellos de manera separada o con políticas especiales del manejo de individuos dentro de la población. Los enfoques más conocidos se presentan a continuación:

VEGA

David Shaffer [122] extiende el uso del algoritmo genético simple en su propuesta llamada *Vector Evaluated Genetic Algorithm* (VEGA), el cual se distingue por la manera en la que se implementa la selección de individuos.

En cada generación se divide a la población en k subpoblaciones, donde k es el número de objetivos y el tamaño de cada subpoblación es de N/k individuos, donde N es el número total de individuos de la población. Para la subpoblación i se seleccionan los individuos de acuerdo a la i-ésima función objetivo $i = 1, \ldots, k$. Una vez terminado este proceso se unen todos los individuos seleccionados en una sola población a la cual se le aplican operadores genéticos para obtener una nueva generación.De esta manera, se pretende elegir individuos que sean no dominados de manera local (en un objetivo) de manera que se recombinen con otros que tengan las mismas características pero con otro objetivo y de esta manera, obtener soluciones no dominadas a nivel global y conformar el frente de Pareto correspondiente.

La principal ventaja de este método es su sencillez. Sin embargo, presenta dificultades para producir soluciones óptimas de Pareto cuando el frente de Pareto es no convexo.

Varios autores han utilizado VEGA para resolver problemas. Entre ellos están Ritzel y Wayland [117], Surry et al. [133], Cvetković et al. [30] y Tamaki et al. [138, 137]

Orden léxicográfico

El funcionamiento de este enfoque se basa en ordenar a los objetivos con respecto a su importancia. Esta tarea la realiza el diseñador. Para obtener la solución óptima se minimizan (o maximizan, según sea el caso) las funciones objetivo comenzando con la más importante, continuando con las demás hasta terminar con todas. Una vez que se obtiene la solución para un objetivo, ésta se agrega como restricción para la solución del siguiente objetivo. De ahí que el problema se plantea de manera general para el objetivo i:

$$Minimizar \;\; f_i(\bar{x}) \tag{3.17}$$

sujeta a

$$g_j(\bar{x}) \leq 0; \quad j = 1, 2, \ldots, m \tag{3.18}$$

$$f_l(\bar{x}) = f_l^*, \quad l = 1, 2, \ldots, i-1 \tag{3.19}$$

Este enfoque tiene la ventaja de ser sencillo de implementarse por ello ha sido muy utilizado. Sin embargo, tiende a favorecer a algunos objetivos. Esto ocasiona que la población converja a una zona particular del frente de Pareto en lugar de formarlo por completo.

Entre los autores que han utilizado esta técnica se encuentran Fourman [40] y Kursawe [77].

Teoría de juegos

Esta técnica basa su funcionamiento en la asignación de objetivos y variables del problema a dos jugadores de manera que compitan entre ellos para minimizar (o maximizar, según sea el caso) su correspondiente objetivo. La interpretación geométrica para el caso de 2 objetivos f_1 y f_2 se refiere a la intersección de las líneas que representan las opciones para el primer y segundo jugador para valores fijos de las variables de decisión (que para este ejemplo de dos objetivos también son 2) x_1 y x_2. Si se presenta esta intersección, se considera a este punto un candidato para la minimización de ambos objetivos f_1 y f_2, asumiendo un juego no cooperativo. Este punto se conoce como la *solución de equilibrio de Nash* y representa una condición de equilibrio donde ninguno de los jugadores puede incrementar (o decrementar) de manera unilateral su criterio (objetivo) [94].

La eficiencia computacional de este enfoque es muy buena. Sin embargo Périaux et al. [106] han reportado en su trabajo que esta técnica no puede generar más de una solución no dominada, la cual no necesariamente es la mejor. En la literatura se encuentran intentos de extender esta idea a más de dos jugadores y utilizar un enfoque cooperativo en lugar de competitivo [112, 113].

Entre las aplicaciones de esta propuesta se encuentran la del mismo Périaux et al. [106] donde se ocupan estas ideas para un problema biobjetivo.

Uso del género para identificar objetivos

Este enfoque comparte la idea de VEGA de dividir a la población dependiendo de los objetivos. La diferencia es que en este caso se asigna sexo a cada individuo al momento de crearse, lo cual indica a qué objetivo corresponde cada uno.

Existen propuestas como la de Allenson [3] donde se tienen dos sexos (funciones objetivo) y la cruza es sexual sólo con individuos del sexo contrario y el sexo de los hijos se asigna de manera aleatoria. Por otro lado, Lis y Eiben [80] proponen utilizar más de dos sexos (objetivos), cruza panmítica[2] donde el padre que más información herede a su hijo también le heredará su sexo. Los empates entre padres se resuelven al azar. Además, la población es traslapable[3] y la mutación se restringe de forma que no pueda cambiar el sexo de un individuo.

Esta técnica ha mostrado buen desempeño, pero su manejo se complica cuando el número de objetivos crece y, por ende, crece el número de sexos. Además, la cruza se complica ya que se requieren muchos padres para generar

[2] En la cruza panmítica más de dos padres participan en la creación de un hijo, aportando cada uno información al descendiente

[3] Una población traslapable se refiere a aquella en donde un mismo indivuduo tiene la posibilidad de sobrevivir por más de una generación

un solo hijo y la población debe ser numerosa para mantener una diversidad de sexos razonable.

Entre los reportes de uso de esta técnica se encuentra el de Lis y Eiben [80] en un problema con dos objetivos propuesto por Srinivas y Deb [130].

Otras propuestas

Existen otras técnicas poblacionales o con manejo individual de objetivos:

- **Min-Max con pesos**: Propuesto para un modelo lineal por Jutler [69] y Solich [129] y utilizado por Osyczka [98, 99, 101], Rao [111] y Tseng y Lu [140]. Sugiere utilizar los extremos de cada función objetivo y calcular la solución multiobjetivo por medio de los valores de los incrementos relativos de todas las funciones objetivos. Coello [19, 28] propuso dos variantes de este enfoque.

- **Uso del teorema del contacto**: Osyczka and Kundu [100] proponen el uso de uno de los principales teoremas en optimización multiobjetivo [79] para determinar las distancias relativas de un vector solución con respecto al frente de Pareto.

- **Algoritmo genético no generacional**: Basado en el funcionamiento de los Sistemas Clasificadores [45], donde se reemplaza al peor individuo y se actualiza la aptitud de las restantes soluciones. Valenzuela-Rendón y Uresti-Charre [142] transforman el problema multiobjetivo en uno biobjetivo y lo resuelven usando una combinación lineal de pesos.

- **Pesos aleatorios y elitismo**: Similar a una aplicación basada en el enfoque Min-Max con pesos de Hajela [52], pero generando los pesos de manera distinta y separando del resto de la población el conjunto de soluciones no dominadas obtenidas en cada generación. Propuesta por Ishibuchi y Murata [61]

3.3.3 Basadas en el óptimo de Pareto

Las técnicas descritas a continuación están basadas en el uso de jerarquías de Pareto (*Pareto ranking*) propuesto por Goldberg [45] quien sugiere el uso de jerarquías basadas en el concepto de no dominancia para efectuar la selección, a fin de mover una población de soluciones potenciales hacia el frente de Pareto. La idea general consiste en encontrar un conjunto de soluciones no dominadas por el resto de la población actual y asignarles la jerarquía más alta. Posteriormente, este conjunto se aparta y se genera un nuevo conjunto de soluciones no dominadas de la población restante y se les asigna una jerarquía menor a la previamente asignada. Este proceso termina cuando toda la población está adecuadamente jerarquizada. Para mantener diversidad, el mismo Goldberg [45] propone el uso de una técnica de nichos. Con estos elementos, tenderán a generarse las soluciones no dominadas a lo largo del frente de Pareto.

Multiple Objective Genetic Algorithm (MOGA)

Propuesto por Fonseca y Fleming [39]. Basa su funcionamiento en otorgar un rango a un individuo x de acuerdo con las $p_i^{(t)}$ soluciones de la población en la generación t por las cuales es domimado:

$$rango(x_i, t) = 1 + p_i^{(t)} \tag{3.20}$$

Los individuos no dominados tienen un rango de 1. Por otro lado, los individuos dominados obtienen una jerarquía de acuerdo a la densidad de población en la región.

El algoritmo es el siguiente:

1. Ordenar la población de acuerdo al rango de cada individuo.
2. Asignar el valor de aptitud a cada individuo interpolando a partir del mejor (rango 1) hasta el peor rango ($n \leq N$) siguiendo la propuesta de Goldberg [45].
3. Promediar la aptitud de los individuos con el mismo rango. Esto mantiene constante la aptitud global de la población y, por ende, se mantiene una presión de selección moderada.

La manera en la que MOGA asigna la aptitud de cada individuo genera una presión de selección muy fuerte, ocasionando convergencia prematura de acuerdo con lo reportado por Goldberg y Deb [46]. Por ello, Fonseca y Fleming [39] usan nichos para distribuir la población a lo largo de la región óptima de Pareto. Para dividir la aptitud de los individuos cercanos utilizan el valor de evaluación de las funciones objetivo en lugar de los valores de las variables de decisión (nivel fenotípico) [131].

Entre los autores que han reportado el uso de MOGA se encuentran Tan y Li [139] para optimización de controladores; Obayashi [95] reporta la utilización de este enfoque para el diseño óptimo de modelos de cuchillas de compresión; Chipperfield y Fleming [17] lo usan para el diseño de un sistema de control para una turbina de gas. Por último, se encuentra Rodríguez Vázquez et al. [118], quienes extienden MOGA a la programación genética y lo usan para identificación de sistemas [74].

Las principales ventajas de MOGA son su eficiencia y sencillez para implementarse. Entre sus desventajas se encuentra su dependencia del parámetro que determina la distribución de aptitud entre individuos similares σ_{share}. Otro problema con la técnica es que no pueden existir al mismo tiempo dos vectores diferentes con los mismos valores para las funciones objetivo debido a que la distribución de aptitud se hace en el espacio de los valores de las funciones objetivo y no en el de las variables de decisión [131].

Non-dominated Sorting Genetic Algorithm (NSGA)

Propuesto por Srinivas y Deb [130], NSGA genera varios niveles de clasificación de los individuos previo al proceso de selección. Esta taxonomía se genera bajo un criterio de no dominancia conformado por los siguientes puntos:

1. Todos los individuos no dominados se clasifican en una categoría donde se les asigna un valor de aptitud temporal proporcional al tamaño de la población para igualar la capacidad reproductiva de ese grupo de soluciones.

2. Con el objeto de mantener diversidad, se comparte la aptitud de este conjunto de individuos (uso de nichos).

3. Se separa momentáneamente este grupo de indiviudos.

4. Con los individuos restantes de la población original se genera un nuevo grupo de soluciones no dominadas asignando un valor temporal de aptitud menor al del grupo anterior repitiendo los 3 pasos anteriores.

5. Parar este primer ciclo cuando todos los individuos de la población se encuentren clasificados.

6. Realizar el proceso de selección proporcional estocástica.

7. Aplicar operadores genéticos (cruza y mutación).

8. Parar cuando se cumpla el criterio de terminación del algoritmo (número máximo de generaciones).

Todo este proceso permite que los individuos con alta jerarquía adquieran una alta aptitud y a su vez, una mayor probabilidad de ser seleccionados, lo que permite dirigir rápidamente la búsqueda hacia regiones no dominadas del espacio de soluciones.

Cabe mencionar que a diferencia de MOGA, NSGA divide la aptitud de los individuos (nichos) en el dominio de las variables del problema (nivel fenotípico) con el objeto de asegurar una mejor distribución de los individuos y permitir la coexistencia de soluciones equivalentes.

En la literatura se encuentran aplicaciones de esta técnica hechas por Périaux et al. [106] para reflectores aerodinámicos; Veradajan [143] reporta el uso de NSGA con torneo binario para optimizar portafolios de inversión y Michielssen and Weile [91] usaron también NSGA para el diseño de un sistema electromagnético.

Entre las principales ventajas de NSGA se encuentra su eficiencia gracias a que objetivos múltiples se reducen a un valor temporal de aptitud el cual jerarquiza a la población en subconjuntos de individuos. Sin embargo, en la literarura se han reportado pruebas donde NSGA demuestra un desempeño inferior a MOGA con respecto a su eficiencia computacional y las soluciones generadas, y una sensibilidad mayor hacia el factor de distribución de aptitud σ_{share} [19].

Niched Pareto Genetic Algorithm (NPGA)

Horn y Nafpliotis [57] proponen el uso de un torneo basado en no dominancia para la selección de individuos. La competencia no se realiza entre dos individuos como en el torneo binario tradicional. Sin embargo, en este caso se selecciona un subconjunto de soluciones de la población para competir contra dos candidatos a selección; gana aquél que sea no dominado por ese subconjunto de individuos. En caso de empates (ambos no dominados o ambos dominados), se disminuye la aptitud de ambos competidores [47].

NPGA utiliza una distribución de la aptitud en el dominio de los valores de las funciones objetivo. Horn y Nafpliotis [57] sugieren el uso de una métrica que combina ambos dominios (funciones objetivo y variables de decisión) para la creación de nichos. A esta propuesta se le conoce como *nested sharing*.

Belegundu et al. [9] utilizaron NPGA para el diseño de compuestos de cerámica laminada. Por otro lado, Poloni y Pediroda [107] reportan el uso de esta propuesta para el diseño de una lámina aérea.

NPGA es más rápido en su desempeño porque no clasifica a toda la población sino que utiliza un subconjunto de ella (típicamente el 10% de la población total). Produce buenos frentes no dominados que permanecen un número considerable de generaciones [19]. Su desventaja es que requiere, además del parámetro de distribución de aptitud, de escoger el tamaño del torneo (subconjunto de la población) para el proceso de selección.

Capítulo 4

Técnicas multiobjetivo para manejo de restricciones

Un problema con un solo objetivo y restricciones puede transformarse de manera sencilla en un problema multiobjetivo de manera que se tienen $m+1$ objetivos, donde m es el número de restricciones. Entonces se tiene el vector $\bar{v} = (f(\mathbf{X}), f_1(\mathbf{X}), \ldots, f_m(\mathbf{X}))$, donde $f_1(\mathbf{X}), \ldots, f_m(\mathbf{X})$ son las restricciones del problema y $f(\mathbf{X})$ es la función objetivo original. Con ello se pretende encontrar aquella solución que satisfaga las m restricciones y minimice (o maximize según sea el caso) la función objetivo: Encontrar $\mathbf{X}$ tal que $f_i(\mathbf{X}) = 0$ para $1 \leq i \leq m$ y $f(\mathbf{X}) \leq f(\mathbf{Y})$ para toda $\mathbf{Y}$ factible (para el caso de minimización).

A continuación se describen algunas propuestas que utilizan esta idea, el proceso de jerarquización de Pareto y nichos para evitar que el algoritmo evolutivo quede atrapado en un óptimo local. Diferentes mecanismos son incorporados en algunas técnicas para alcanzar el óptimo global ya que el enfoque multiobjetivo por sí solo se ha reportado que no lo logra [105, 64, 133].

4.1 Constrained Optimisation Multi-Objective Genetic Algorithms (COMOGA)

Surry et al. [133] utilizan una combinación de jerarquías de Pareto y VEGA [122], donde los individuos se jerarquizan dependiendo de su cantidad de violación de restricciones (cantidad de individuos dominados por una solución). Sin embargo, la selección se realiza no tan sólo por jerarquías, sino que también se basa en la aptitud y en parámetros adicionales que el usuario debe establecer.

Esta propuesta fue probada en un problema de diseño de una red de gas en el cual mostró un desempeño semejante al de una función de penalización. Su principal problema es la necesidad de establecer el valor de al menos un parámetro adicional, lo que puede asemejarse al problema original de establecer factores en las funciones de penalización.

4.2 Direct Genetic Search

Parmee y Purchase [105] proponen el uso de VEGA [122] para dirigir la búsqueda de un algoritmo evolutivo hacia la zona factible del espacio de búsqueda de un problema de diseño de una turbina de gas. Una vez que se tiene un punto factible se realiza una búsqueda local, variando los valores de la solución arrojados por VEGA para formar un hipercubo cuyos puntos presumiblemente se encontrarán también dentro de la zona factible.

En este enfoque no se utiliza la técnica multiobjetivo para recorrer la zona factible sino que sólo para llegar a ella. Además, no existen comparaciones del desempeño de esta técnica con respecto a otras.

4.3 Reducción de objetivos, no dominancia y búsqueda lineal

Una combinación de técnicas después de reducir un problema de un objetivo y restricciones a un problema con dos objetivos es sugerida por Camponogara y Taludkar [15]. La redefinición del problema consiste en optimizar la función objetivo original y minimizar:

$$\Phi(\mathbf{X}) = \sum_{i=1}^{n} \max(0, g_i(\mathbf{X})) \tag{4.1}$$

La expresión (4.1) pretende minimizar la cantidad de violación de un individuo (volverlo factible). En cada generación del algoritmo se genera una lista de conjuntos de frentes de Pareto a partir de los individuos de la población. El operador de recombinación (que sustituye a la cruza) toma dos conjuntos de Pareto S_i y S_j donde $i < j$ y dos soluciones $x_i \in S_i$ y $x_j \in S_j$ de manera que x_i domine a x_j. A partir de estos puntos se define una dirección de búsqueda

$$d = \frac{(x_i - x_j)}{|x_i - x_j|} \tag{4.2}$$

Esta dirección indica el camino más prometedor para minimizar todos los objetivos. En este momento se inicia una búsqueda lineal proyectando a d sobre el eje de una variable j en el espacio de soluciones y siguiéndola con el objetivo de encontrar una solución x que domine tanto a x_i y a x_j. Después de generar algunos descendientes se inicia una nueva generación descartando aquellos individuos menos aptos.

En intervalos establecidos se reemplaza la mitad de la población con soluciones aleatorias para mantener la diversidad. Esto es necesario debido a la alta presión de selección y denota los problemas que tiene esta técnica para mantener diversidad. Más aún, el costo computacional se incrementa al implementar una búsqueda lineal dentro del algoritmo genético.

4.4 Selección por torneo con reglas especiales

Jiménez y Verdegay [64] propusieron un algoritmo evolutivo con selección por torneo entre un subconjunto de individuos de una población no traslapable y el uso de elitismo. Siguiendo un proceso similar al algoritmo genético común, se establecen una serie de reglas para determinar al ganador en el proceso de selección:

- Un individuo factible es preferible a otro también factible pero con menor aptitud.
- Un individuo factible es preferible a un individuo no factible, sin importar su aptitud.
- Entre dos individuos no factibles, es preferible aquel que viole en menor cantidad las restricciones del problema.

Gracias a estas reglas, el algoritmo se centra primordialmente en la satisfacción de restricciones, lo que ocasiona que se muestree aleatoriamente la zona factible del espacio de búsqueda. Por ello este enfoque es recomendable para alcanzar la zona factible de espacios altamente restringidos, pero puede no ser tan adecuado para encontrar el óptimo global de una función arbitraria.

4.5 VEGA para manejo de restricciones

Coello [21] propuso un esquema poblacional similar a VEGA. En cada generación se divide la población en $m + 1$ subpoblaciones de igual tamaño fijo, donde m es el número de restricciones del problema y la otra subpoblación se refiere a la función objetivo la cual rige su proceso de selección basado en la aptitud del individuo (decodificar las variables del problema y sustituirlas en la función objetivo) sin tomar en cuenta las restricciones. Las restantes m subpoblaciones toman cada una como base para seleccionar individuos una sola restricción del problema. De esta manera se logra que cada subpoblación centre sus esfuerzos en satisfacer una restricción. Los siguientes criterios se toman en cuenta para cada subpoblación $j, j = 1, \ldots, m$:

$$\begin{array}{lll}
\textbf{if } g_j(\mathbf{X}) < 0.0 & \textbf{then} & \text{fitness} = g_j(\mathbf{X}) \\
\textbf{else if } v \neq 0 & \textbf{then} & \text{fitness} = -v \\
\textbf{else} & & \text{fitness} = f(\mathbf{X})
\end{array}$$

Donde $g_j(\mathbf{X})$ se refiere a la restricción j del problema, v significa el número de restricciones violadas ($v \leq m$) y $f(\mathbf{X})$ es el valor del individuo en la función objetivo.

Posteriormente se aplican los operadores genéticos a toda la población permitiendo que individuos de subpoblaciones distintas se crucen, tal y como se propone en VEGA. Con ello se espera que individuos con una buena aptitud se relacionen con aquellos que no violan restricciones o, por lo menos, minimizan la violación de una restricción. Así, se espera que al final se obtengan individuos factibles y con una aptitud al menos cercana a la óptima.

Cabe señalar que las aptitudes (*fitness*) de los individuos integrantes de las subpoblaciones encargadas de satisfacer una restricción se asignan dependiendo de su factibilidad (como se describe en los criterios mostrados arriba y suponiendo que $g_j(\mathbf{X}) \geq 0$ significa que la restricción se satisface):

- Si el individuo es no factible, se asigna como aptitud el valor positivo de la cantidad de violación de la restricción con el objetivo de minimizarla.

- Si el individuo no viola la restricción correspondiente a esa subpoblación pero es no factible, se asigna el valor del número de restricciones que viola para minimizarse (coopera con las demás subpoblaciones que tratan de satisfacer restricciones).

- Si el individuo es factible, se asigna como aptitud el valor de la función objetivo para optimizar ese valor (coopera con la subpoblación que optimiza la función objetivo).

Esta propuesta no utiliza el concepto de no dominancia para crear jerarquías de individuos y no requiere de un factor de penalización ni tampoco de una técnica para mantener diversidad. Sin embargo, el número de subpoblaciones (y de individuos) crece con respecto al número de restricciones que tenga el problema.

4.6 Jerarquías en distintos espacios y restricciones de apareamiento

La propuesta hecha por Ray et al. [136] se basa en el uso del concepto de no dominancia de soluciones tanto en el espacio de los objetivos como en el espacio de los valores de las restricciones. Utiliza además una estrategia de selección que permite eliminar debilidades en ambos espacios y asegurar una mejor satisfacción de restricciones en los descendientes de los individuos que se reproduzcan. Utiliza nichos para preservar la diversidad en la población calculando similitud mediante distancias Euclidianas.

$$\begin{pmatrix} f_{11} & f_{12} & \dots & f_{1k} \\ f_{21} & f_{22} & \dots & f_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ f_{M1} & f_{M2} & \dots & f_{Mk} \end{pmatrix} \tag{4.3}$$

$$\begin{pmatrix} c_{11} & c_{12} & \dots & c_{1s} \\ c_{21} & c_{22} & \dots & c_{2s} \\ \vdots & \vdots & \ddots & \vdots \\ c_{M1} & c_{M2} & \dots & c_{Ms} \end{pmatrix} \tag{4.4}$$

La separación de espacios se implementa generando tres matrices, una llamada *matriz objetivo*(ecuación 4.3) agrupando en cada renglón los valores de uno de los M individuos para cada objetivo. En la segunda, llamada *matriz de restricción*(ecuación 4.4) se agrupan en cada renglón los valores para cada restricción del problema de las M soluciones. La tercera, conocida como *matriz combinada*(ecuación 4.5), se genera uniendo las dos matrices previas.

$$\begin{pmatrix} f_{11} & f_{12} & \dots & f_{1k} & c_{11} & c_{12} & \dots & c_{1s} \\ f_{21} & f_{22} & \dots & f_{2k} & c_{21} & c_{22} & \dots & c_{2s} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ f_{M1} & f_{M2} & \dots & f_{Mk} & c_{M1} & c_{M2} & \dots & c_{Ms} \end{pmatrix} \tag{4.5}$$

Con estas tres matrices se crean jerarquías de Pareto para cada una de ellas de manera similar a como lo hace NSGA [130]. En la selección de individuos se favorece a aquellos con alta jerarquía ($= 1$) en la matriz combinada y que sean factibles. Aquellos que cumplan con estos requisitos se insertan en la nueva población. Los elementos restantes de la población se generan seleccionando un individuo A y dos candidatos a cruza (B y C) teniendo como base su jerarquía y el tipo de problema que se resuelve. Si el problema no tiene restricciones se escogen los tres individuos con respecto al rango en la matriz objetivo. Si el problema es moderadamente restringido se escoge a A basándose en su rango en la matriz objetivo y a B y C con respecto a su rango en la matriz de restricción. Si el problema es altamente restringido se escogen los tres dependiendo de su rango en la matriz de restricción. Existen tres tipos de condiciones para escoger a uno de los dos candidatos (B y C) para cruzarse con A:

- Si ambos son factibles, se escoge al que tenga mayor jerarquía en la matriz objetivo.
- Si ambos son no factibles se escoge al que tenga mayor jerarquía en la matriz de restricción.
- Si uno es factible y el otro es no factible siempre se preferirá al factible.

Mediante cruza uniforme entre los individuos seleccionados se genera un hijo. Otros dos hijos se generan mediante un proceso llamado *random mix* el cual asegura que se generen valores factibles para las variables del problema que no existan en los padres.

Esta técnica ha demostrado que puede disminuir significativamente el número de evaluaciones de la función de aptitud a comparación de otras propuestas. El trabajo cooperativo que realizan los espacios definidos por las tres matrices permite una convergencia más rápida, aún en espacios altamente restringidos.

La principal desventaja de esta técnica es que puede resultar bastante difícil de implementarse, y dicha tarea pudiese no justificarse en algunos casos.

4.7 Manejo de restricciones basado en no dominancia y MOGA

Coello [24] propone el uso del concepto de no dominancia para jerarquizar a los individuos de toda la población, de manera similar a la utilizada por MOGA [39], pero con ajustes para manejar restricciones y buscar no sólo soluciones compromiso. Para otorgar un rango a un individuo en particular debe compararse contra todos los individuos de la población. Basa su criterio de selección en tres reglas:

- Todo individuo factible se prefiere a cualquiera no factible.
- Si ambos individuos son no factibles, se prefiere aquel que viole el menor número de restricciones del problema.
- Si ambos individuos son no factibles y violan el mismo número de restricciones, ganará aquel que viole en menor cantidad acumulada las restricciones del problema.

El algoritmos de asignación de jerarquías se describe a continuación:

Sea el vector $\vec{x}_i$ $(i = 1, \ldots, pop\_size)$ un individuo de la población actual cuyo tamaño es pop_size.

- Para calcular el rango de un individuo $\vec{x}_i$ se realiza lo siguiente:

$$\text{rank}(\vec{x}_i) = \text{count}(\vec{x}_i) + 1 \tag{4.6}$$

  donde count($\vec{x}_i$) se calcula de acuerdo a los siguientes criterios:

  1. Comparar $\vec{x}_i$ con los demás individuos de la población, uno por uno. Sea $\vec{x}_j$ ($j = 1, \ldots, pop\_size$ y $j \neq i$) el individuo contra el cual se comparará $\vec{x}_i$.
  2. Inicializar count($\vec{x}_i$)($i = 1, \ldots, pop\_size$) con un valor de cero.
  3. Si $\vec{x}_i$ y $\vec{x}_j$ son factibles, a ambos se les otorga un rango de cero y count($\vec{x}_i$) permanece sin cambios.
  4. Si $\vec{x}_i$ es no factible y $\vec{x}_j$ es factible, se incrementa el valor de count($\vec{x}_i$) en uno.
  5. Si $\vec{x}_i$ y $\vec{x}_j$ son no factibles, pero $\vec{x}_i$ viola más restricciones que $\vec{x}_j$, entonces se incrementa el valor de count($\vec{x}_i$) en uno.
  6. Si $\vec{x}_i$ y $\vec{x}_j$ son ambos no factibles y violan el mismo número de restricciones, pero $\vec{x}_i$ tiene una cantidad de violación total mayor que la de $\vec{x}_j$, se incrementa en uno el valor de count($\vec{x}_i$).

     La cantidad total de violación de un individuo $\vec{x}_i$ en las p restricciones del problema (coef($\vec{x}_i$)) se calcula de la siguiente manera (asumiendo que si $g_i(\vec{x}) \leq 0$ entonces se considera una restricción satisfecha):

$$\text{coef}(\vec{x}_i) = \sum_{k=1}^{p} g_k(\vec{x}_i) \quad \text{para toda } g_k(\vec{x}_i) > 0 \tag{4.7}$$

- La aptitud de un individuo, la cual debe normalizarse, se calcula de acuerdo con las siguientes reglas:

  1. Si $\vec{x}_i$ es factible, entonces rank($\vec{x}_i$) = $fitness(\vec{x}_i)$, sino
  2. rank($\vec{x}_i$) = $\frac{1}{\text{rank}(\vec{x}_i)}$

- La selección de individuos se realiza mediante muestreo universal estocástico de acuerdo a su rango rank($\vec{x}_i$).

Esta propuesta además utiliza auto adaptación en los parámetros de cruza y mutación. Ha demostrado un buen desempeño con respecto al número de evaluaciones de la función de aptitud y a la calidad de las soluciones arrojadas. Además, no requiere de una técnica adicional para mantener diversidad en la población [24].

Su principal desventaja es que el costo computacional crece con poblaciones grandes, ya que se requiere una comparación de cada individuo con el resto de la población para asignar rangos y poder llevar a cabo el proceso de selección.

Capítulo 5

NPGA para manejo de restricciones

La idea de Goldberg de utilizar al algoritmo genético para optimizar un vector de funciones en lugar de una sola función objetivo [45] fue interpretado de diversas maneras. Fonseca y Fleming [39] implementaron MOGA donde se verifica no dominancia para jerarquizar a cada individuo de la población; Srinivas y Deb [130] propusieron NSGA donde se generan grupos de individuos no dominados para obtener niveles de clasificación. Horn y Nafpliotis [57] plantean en su propuesta el uso de un torneo basado en no dominancia y un mecanismo de nichos.

En este capítulo se describe la versión original de NPGA para optimización multiobjetivo y posteriormente se detalla su adaptación para el manejo de restricciones.

5.1 NPGA para optimización multiobjetivo

NPGA modifica el proceso de selección de un algoritmo genético (AG) de manera que proporcione múltiples soluciones a problemas con múltiples funciones objetivo. Para lograrlo, utiliza una de las técnicas de selección más populares, el torneo.

En un torneo binario, dos individuos compiten entre sí tomando como referencia su aptitud. En su versión determinística gana aquel con aptitud mayor. En su versión probabilística, se decide con igual probabilidad la selección del mejor o del peor. Para aumentar la presión de selección se agregan más individuos al torneo.

Para evitar que el torneo dirija la búsqueda del AG hacia una única solución NPGA modifica la selección por torneo de dos maneras:

- Torneo basado en no dominancia.
- Uso de nichos para resolver empates y mantener diversidad.

5.1.1 Torneo basado en no dominancia

La idea de este tipo de torneo se basa en hacer competir a los individuos de acuerdo a criterios de no dominancia y ya no con respecto a su aptitud. Esto es, entre dos individuos se prefire aquel que domine al otro.

En las primeras pruebas de esta propuesta se utilizó el torneo binario, pero se observó que producía muchos individuos dominados en las útimas estapas del proceso evolutivo; es decir, no había suficiente presión de dominancia. Para corregir este comportamiento se propuso lo siguiente:

- Seleccionar al azar dos individuos candidatos a ser seleccionados.
- Seleccionar un subconjunto de individuos de la población actual (normalmente el 10% del total).
- Verificar el criterio de no dominancia de cada candidato con respecto al subconjunto de individuos de la población actual.
- Si uno de los candidatos es no dominado por el subconjunto de individuos y el otro es dominado por el mismo grupo, el primero gana el torneo.
- Si ambos individuos son no dominados o dominados se utiliza un mecanismo de nichos (descrito a continuación) para determinar al ganador.

Para lograr esto se agrega el parámetro conocido como t_{dom}, el cual indica la cardinalidad del subconjunto de individuos de la población actual que se utilizan para determinar la no dominancia de un individuo candidato a seleccionarse. Los autores mencionan que este parámetro ayuda a controlar la presión de selección en el proceso.

5.1.2 Uso de nichos para resolver empates y mantener diversidad

En el capítulo 3 se comentó que el objetivo de utilizar nichos en un algoritmo evolutivo era el mantener diversidad en la población y distribuir a los individuos a lo largo del frente de Pareto en el caso de optimización multiobjetivo (o de los máximos o mínimos de una función para el caso de optimización de un solo objetivo) [45].

Esta distribución de individuos es posible mediante la degradación de la aptitud de aquellas soluciones que se encuentren dentro del mismo nicho. Para determinar la aptitud de un individuo ubicado dentro de un nicho se divide ésta entre un contador de nicho m_i. Este valor estima qué tan poblado se encuentra el nicho i. Se calcula mediante la siguiente expresión [57]:

$$m_i = \sum_{j \in Pop} Sh\left[d[i,j]\right] \tag{5.1}$$

Donde $d[i,j]$ es una medida de distancia entre los individuos i y j, y $Sh[d[i,j]]$ es la función de distribución de aptitud (*Sharing Function*), la cual es una función decreciente de $d[i,j]$ de manera que $Sh[0] = 1$ y $Sh[d \geq \sigma_{share}] = 0$. σ_{share} es el radio del nicho, escogido por el usuario y que determina el estimado de la separación mínima esperada entre los picos (máximos o mínimos) de la función. Los individuos dentro del radio σ_{share} degradarán su aptitud entre sí. La función más utilizada es la función triangular *Triangular Sharing Function* donde $Sh[d] = 1 - d/\sigma_{share}$ para $d \leq \sigma_{share}$ y $Sh[d] = 0$ para $d \geq \sigma_{share}$.

La medida de similitud para conformar los nichos se puede llevar a cabo a nivel del genotipo (codificación de las variables del problema) y a nivel del fenotipo (valores de las variables del problema). Deb y Goldberg [34] sugieren que esta última proporciona mejores resultados. Sin embargo, para el caso de optimización multiobjetivo se agrega una posibilidad más: Calcular la similitud a nivel de los valores de las funciones objetivo, la cual parece más adecuada ya que el propósito principal es generar una distribución de individuos a lo largo de todo el frente de Pareto, que precisamente se forma en el espacio de los valores de las funciones objetivo.

La técnica de nichos utilizada en NPGA para seleccionar un individuo

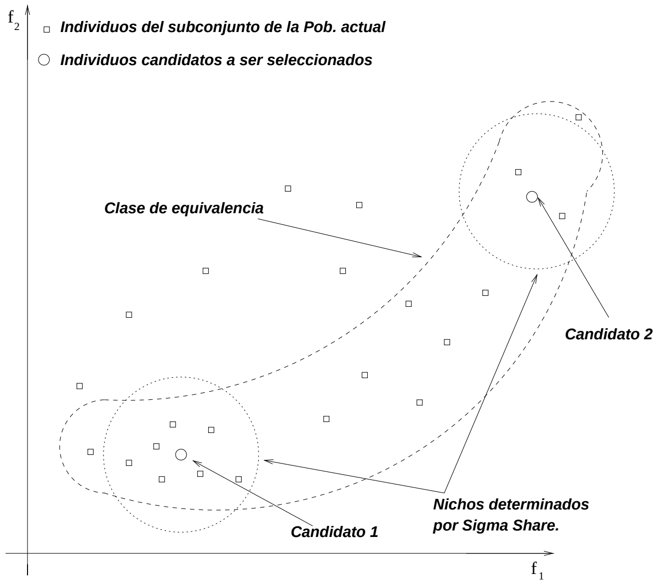

Figura 5.1: Equivalence Class Sharing

en caso de empate en el torneo basado en no dominancia trabaja a nivel de los valores de las funciones objetivo. Se calcula para cada individuo i su correspondiente contador de nicho m_i, pero no se utiliza para degradar su aptitud, sino que simplemente se escogerá aquel con menor valor de su contador m_i ya que indica que existen menos individuos en su nicho. Así, al poblar cada nicho de manera equitativa y seleccionar un valor adecuado de σ_{share} se cubrirá todo el frente de Pareto. A esta propuesta se le conoce como distribución por clases de equivalencia (*equivalence class sharing*). En la figura 5.1, donde se intenta maximizar f_1 y minimizar f_2, se muestra el caso en el que el candidato 2 será seleccionado ya que su nicho se encuentra menos poblado, aunque desde el punto de vista de no dominancia ninguna de las dos soluciones se prefiere. De esta manera se promueve la diversidad. El algoritmo de NPGA se muestra a continuación [57](*tournlist* es una lista con los índices de los individuos de la población actual, identificada por *oldpop*):

```
function select
  begin
    barajear(tournlist);
    candidato_1 = tournlist[0];
    candidato_2 = tournlist[1];
    candidato_1_dominado = FALSE;
    candidato_2_dominado = FALSE;
    for (i = 2 to tdom + 2) /* torneo basado en no dominancia */
      begin
        comparado=tournlist[i];
        if (oldpop[comparado] domina_a oldpop[candidato_1])
          candidato_1_dominado=TRUE;
        if (oldpop[comparado] domina_a oldpop[candidato_2])
          candidato_2_dominado=TRUE;
      end
    if (candidato_1_dominado=TRUE AND candidato_2_dominado=FALSE)
      winner=candidato_2;
    else
      if (candidato_1_dominado=FALSE AND candidato_2_dominado=TRUE)
        winner=candidato_1;
      else /* equivalence class sharing */
        if (oldpop[candidato_1].nichecount > oldpop[candidato_2].nichecount)
          winner=candidato_2;
        else
          winner=candidato_1;
    return(winner);
  end
```

5.1.3 Pruebas

Horn y Nafpliotis [57] aplicaron su propuesta a tres problemas: Uno muy sencillo con cadenas binarias, la función $F2$ de Schaffer y uno sobre monitoreo de aguas contaminadas.

En los tres problemas la técnica mostró resultados buenos. En el primer problema llamado *unitation versus pairs*, donde se trata de maximizar el número de unos y a la vez maximizar el número de posiciones adyacentes con cambio de valor (10 ó 01), la técnica pudo conformar de manera estable el frente de pareto; además, muestran que desactivando el mecanismo de nichos, el algoritmo converge de manera prematura. En el segundo problema, se comparan los resultados de NPGA contra los reportados por Schaffer con VEGA [121]. Los resultados del problema en la generación 200 muestran un frente de Pareto mejor definido que el formado por VEGA en la generación 3. Para el problema 3, la técnica logra mantener diversidad a lo largo de 2000 generaciones y forma un aparente frente de Pareto.

5.1.4 Discusión

Con base en los resultados discutidos previamente los autores remarcan la sensibilidad de la técnica a sus parámetros. Establecen la necesidad de una población numerosa para muestrear de manera adecuada el frente de Pareto.

El tamaño del subconjunto de la población actual t_{dom} determina el comportamiento de la búsqueda (N es el tamaño de la población):

- $t_{dom} \approx 1\%$ de N resulta en demasiadas soluciones dominadas, originando un frente de Pareto difuso.
- $t_{dom} \approx 10\%$ de N produce una distribución completa y unida.
- $t_{dom} \gg 10\%$ de N causa convergencia prematura hacia una pequeña parte del frente de Pareto.

Horn y Nafpliotis [57] intuyen que la diversidad en la frontera no dominada es la que ayuda a la búsqueda a extender la frontera, por lo que individuos de lugares distantes de la frontera deben cruzarse para generar hijos que dominen los sectores que quedan entre los padres. Por último, suponen que el *equivalence class sharing* puede ocuparse, no tan sólo en problemas basados en no dominancia, sino para cualquier clase de equivalencia en un orden parcial.

5.2 Modificaciones para manejo de restricciones

Dadas las diferencias entre los problemas multiobjetivo y los problemas con una sola función objetivo, no es posible utilizar de manera directa el algoritmo del NPGA para manejar restricciones. Los cambios requeridos se describen a continuación.

5.2.1 Torneo basado en aptitud, factibilidad y no dominancia

El proceso de selección se enfoca primero a detectar individuos factibles dentro de la población sin importar su valor de aptitud. Con ello se pretende dirigir al AG hacia la zona factible del espacio de búsqueda. Si se da el caso de que ambos candidatos sean factibles (normalmente hacia la mitad o el final del proceso evolutivo), se escoge aquel con mayor aptitud. Esto se hace para buscar, dentro de la zona factible, al óptimo global. Si ambos individuos son no factibles, entonces se escoge un subcojunto de individuos de la población actual al azar y se realiza el torneo basado en no dominancia, como se propone en el NPGA original. En este caso, el criterio es escoger al candidato que sea no dominado por el subconjunto de individuos, siempre y cuando el otro candidato sea dominado por ese mismo grupo. Si ambos son dominados o ambos son no dominados se procede a un desempate que se explica más adelante.

5.2.2 Selección probabilística

Con el objeto de mantener la diversidad requerida para explorar la zona factible del espacio de búsqueda sin utilizar una técnica de nichos se propone el uso de un parámetro que controle la cantidad y forma del proceso de selección. El llamado porcentaje de selección (S_r) indica la proporción de selecciones mediante aptitud, factibilidad y no dominancia, explicada anteriormente. El restante $(1 - S_r)$ serán selecciones probabilísticas con 50% de probabilidad de ser seleccionado para cada candidato.

Con ello se busca tener en todo momento del proceso, individuos dominados que puedan reproducirse y, aunque sean no factibles, permitan al proceso de búsqueda llegar a partes no visitadas de la zona factible.

5.2.3 Diferencias en la revisión de no dominancia

Existe una diferencia sustancial en la forma de verificar la no dominancia entre dos individuos de la población. Basado en la definición proporcionada en las expresiones 3.7 y 3.8, se deben comparar todos los valores de las evaluaciones de las funciones objetivo para determinar la no dominancia entre individuos de una población. Sin embargo, para que este criterio sea útil en la localización de la zona factible en un problema de un solo objetivo con restricciones, se comparan únicamente los valores de las restricciones y se excluye el valor de la función de aptitud, que en esta propuesta es la misma que la función objetivo.

5.2.4 Desempates

Cuando dos individuos candidatos son no factibles y ambos son dominados o no dominados se tiene una situación de empate. Para decidir cuál de ellos es preferible para orientar la búsqueda hacia la zona factible se elige aquel que viole en menor cantidad acumulada las restricciones del problema. Es decir, se escoge aquel que se encuentre más cerca de la zona factible dentro del espacio de búsqueda.

Este criterio coadyuva a adentrarse en la zona de soluciones que no violan restricciones y por medio de los individuos seleccionados de manera estocástica se pretende recorrerla para llegar al óptimo global.

5.2.5 Algoritmo

El algoritmo de selección ya adaptado para el manejo de restricciones, donde *tournlist* es una lista con los índices de los individuos de la población actual denotada por *oldpop*, *fitness* es la aptitud del individuo y $flip(P)$ es una función que regresa TRUE con probabilidad P y regresa FALSE en caso contrario, se muestra a continuación:

```
function select
  begin
    barajear(tournlist);
    candidato_1 = tournlist[0];
    candidato_2 = tournlist[1];
    if (flip(Sr))
      begin /* torneo basado en aptitud, factibilidad y no dominancia */
        candidato_1_dominado = FALSE;
        candidato_2_dominado = FALSE;
```

```
    if (oldpop[candidato_1]=factible AND oldpop[candidato_2]=factible)
      if (oldpop[candidato_1].fitness ≥ oldpop[candidato_2].fitness) /* aptitud */
        winner=candidato_1;
      else
        winner=candidato_2;
    else /* factibilidad */
      if (oldpop[candidato_1]=factible AND oldpop[candidato_2]=NOfactible)
        winner=candidato_1;
      else
        if (oldpop[candidato_1]=NOfactible AND oldpop[candidato_2]=factible)
          winner=candidato_2;
        else
          begin /* no dominancia */
            for (i = 2 to tdom + 2)
              begin
                comparado=tournlist[i];
                if (oldpop[comparado] domina_a oldpop[candidato_1])
                  candidato_1_dominado=TRUE;
                if (oldpop[comparado] domina_a oldpop[candidato_2])
                  candidato_2_dominado=TRUE;
              end
            if (candidato_1_dominado=TRUE AND candidato_2_dominado=FALSE)
              winner=candidato_2;
            else
              if (candidato_1_dominado=FALSE AND candidato_2_dominado=TRUE)
                winner=candidato_1;
              else /* desempates */
                if (oldpop[candidato_1].sumviol < oldpop[candidato_2].sumviol)
                  winner=candidato_1;
                else
                  winner=candidato_2;
          end
  end
else /* Seleccion probabilistica */
  if (flip(0.5))
    winner=candidato_1;
  else
    winner=candidato_2;
```

```
  return(winner);
end
```

Capítulo 6

Pruebas y resultados

6.1 Representación y Operadores Genéticos

Se han utilizado dos tipos de representación para implementar el algoritmo propuesto en esta tesis:

- **Real** (un valor real por cada posición del cromosoma). Se trabaja a nivel del fenotipo.

- **Binaria** con Código de Gray.

La diferencia en la longitud de las cadenas generadas es grande. Para las 3 funciones de ingeniería se habla, por ejemplo, de longitudes de 4, 4 y 3 para la representación real y de 90, 110 y 65 para la representación binaria.

Para la representación binaria se ocuparon la **Cruza de Dos puntos** y **Mutación Uniforme**, mientras que para la representación real se utilizaron **Simulated Binary CrossOver** y **Parameter Based Mutation** [35].

6.2 Funciones de prueba

Con el objeto de comprobar la eficiencia del algoritmo se escogieron funciones propuestas en la literatura para probar técnicas de manejo de restricciones [25, 119]. Todas se refieren a problemas de optimización numérica global previamente resueltas con técnicas evolutivas y técnicas clásicas. Se seleccionaron 4 funciones que plantean problemas de ingeniería y 5 funciones

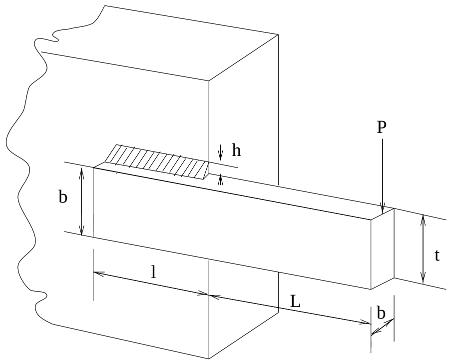

Figura 6.1: Viga soldada usada para el problema 1.

artificiales creadas específicamente para probar técnicas de manejos de restricciones [119].

Entre todas las funciones de prueba se tienen restricciones de desigualdad, espacios de búsqueda de alta y baja dimensionalidad, espacios convexos y no convexos, además de zonas factibles disjuntas y/o muy pequeñas.

Aunque los problemas aquí presentados no contienene restricciones de igualdad, la técnica propuesta puede aplicarse a este tipo de problemas transformándolas a desigualdades de acuerdo a las indicaciones expuestas en la literatura [89, 22]

6.2.1 Problema 1: Diseño de una viga soldada [25]

Una viga soldada es diseñada para minimizar el costo sujeto a restricciones de esfuerzo cortante (τ),esfuerzo de pandeo (σ), carga en la barra (P_c), deflexión en la viga (δ) y restricciones de límite. Hay cuatro variables mostradas en la figura 6.1: h (x_1), l (x_2), t (x_3) and b (x_4). El problema se plantea de la siguiente manera:

Minimizar:

$$F(\mathbf{X}) = 1.10471x_1^2x_2 + 0.04811x_3x_4(14.0 + x_2) \tag{6.1}$$

Sujeta a:

$$g_1(\mathbf{X}) = \tau(\mathbf{X}) - \tau_{max} \leq 0 \tag{6.2}$$

$$g_2(\mathbf{X}) = \sigma(\mathbf{X}) - \sigma_{max} \leq 0 \tag{6.3}$$

$$g_3(\mathbf{X}) = x_1 - x_4 \leq 0 \tag{6.4}$$

$$g_4(\mathbf{X}) = 0.10471x_1^2 + 0.04811x_3x_4(14.0 + x_2) - 5.0 \leq 0 \tag{6.5}$$

$$g_5(\mathbf{X}) = 0.125 - x_1 \leq 0 \tag{6.6}$$

$$g_6(\mathbf{X}) = \delta(\mathbf{X}) - \delta_{max} \leq 0 \tag{6.7}$$

$$g_7(\mathbf{X}) = P - P_c(\mathbf{X}) \leq 0 \tag{6.8}$$

donde:

$$\tau(\mathbf{X}) = \sqrt{(\tau')^2 + 2\tau'\tau''\frac{x_2}{2R} + (\tau'')^2} \tag{6.9}$$

$$\tau' = \frac{P}{\sqrt{2}x_1x_2}, \tau'' = \frac{MR}{J}, M = P\left(L + \frac{x_2}{2}\right) \tag{6.10}$$

$$R = \sqrt{\frac{x_2^2}{4} + \left(\frac{x_1 + x_3}{2}\right)^2} \tag{6.11}$$

$$J = 2\left\{\sqrt{2}x_1x_2\left[\frac{x_2^2}{12} + \left(\frac{x_1 + x_3}{2}\right)^2\right]\right\} \tag{6.12}$$

$$\sigma(\mathbf{X}) = \frac{6PL}{x_4x_3^2}, \delta(\mathbf{X}) = \frac{4PL^3}{Ex_3^3x_4} \tag{6.13}$$

$$P_c(\mathbf{X}) = \frac{4.013E\sqrt{\frac{x_3^2x_4^6}{36}}}{L^2}\left(1 - \frac{x_3}{2L}\sqrt{\frac{E}{4G}}\right) \tag{6.14}$$

$$P = 6000\ lb, \quad L = 14\ in, \quad E = 30 \times 10^6\ psi, \quad G = 12 \times 10^6\ psi \tag{6.15}$$

$$\tau_{max} = 13,600\ psi, \quad \sigma_{max} = 30,000\ psi, \quad \delta_{max} = 0.25\ in \tag{6.16}$$

Donde $0.1 \leq x_1 \leq 2.0$, $0.1 \leq x_2 \leq 10.0$, $0.1 \leq x_3 \leq 10.0$ y $0.1 \leq x_4 \leq 2.0$.

6.2.2 Problema 2: Diseño de un recipiente de presión [25]

Un recipiente cilíndrico está tapado por ambos lados con cabezas hemisféricas como se muestra en la figura 6.2. El objetivo es minimizar el costo total, incluyendo el costo del material, forma y soldadura. Existen cuatro variables: T_s (ancho de la estructura), T_h (ancho de la cabeza), R (radio interno) y L (longitud de la sección cilíndrica del recipiente, sin incluir la cabeza). T_s

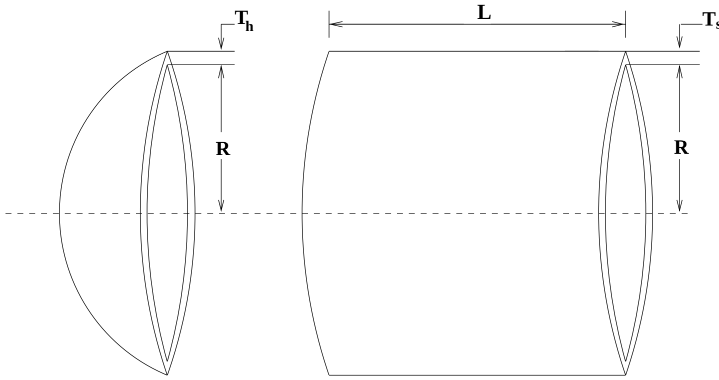

Figura 6.2: Vasija de presión usada para el problema 2.

y T_h son enteros múltiplos de 0.0625 pulgadas, los cuales son los anchos disponibles de los platos de acero, y R y L son continuas. El problema se define como sigue:

Minimizar:

$$F(\mathbf{X}) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3 \quad (6.17)$$

Sujeta a:

$$g_1(\mathbf{X}) = -x_1 + 0.0193x_3 \leq 0 \quad (6.18)$$

$$g_2(\mathbf{X}) = -x_2 + 0.00954x_3 \leq 0 \quad (6.19)$$

$$g_3(\mathbf{X}) = -\pi x_3^2x_4 - \frac{4}{3}\pi x_3^3 + 1,296,000 \leq 0 \quad (6.20)$$

$$g_4(\mathbf{X}) = x_4 - 240 \leq 0 \quad (6.21)$$

Donde $1 \leq x_1 \leq 99$, $1 \leq x_2 \leq 99$, $10 \leq x_3 \leq 200$ y $10 \leq x_4 \leq 200$.

6.2.3 Problema 3: Minimización del peso de un resorte [25]

Consiste en minimizar el peso de un resorte de tensión/compresión sujeto a restricciones de deflección mínima, esfuerzo cortante, frecuencia de agitación, límites de los diámetros externos y de variables de diseño (ver figura 6.3). Las variables de diseño son: D (el diámetro medio del resorte), d (diámetro

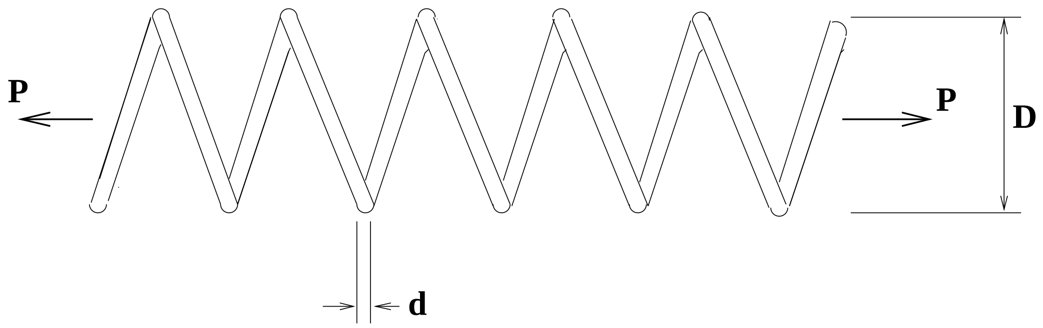

Figura 6.3: Resorte de tensión/compresión usado para el problema 3.

del material) y N (el número de bobinas activas). El problema se expresa formalmente como sigue:

Minimizar
$$F(\mathbf{X}) = (N+2)Dd^2 \tag{6.22}$$

Sujeta a:

$$g_1(\mathbf{X}) = 1 - \frac{D^3 N}{71785 d^4} \leq 0 \tag{6.23}$$

$$g_2(\mathbf{X}) = \frac{4D^2 - dD}{12566(Dd^3 - d^4)} + \frac{1}{5108 d^2} - 1 \leq 0 \tag{6.24}$$

$$g_3(\mathbf{X}) = 1 - \frac{140.45 d}{D^2 N} \leq 0 \tag{6.25}$$

$$g_4(\mathbf{X}) = \frac{D+d}{1.5} - 1 \leq 0 \tag{6.26}$$

Donde $0.05 \leq x_1 \leq 2$, $0.25 \leq x_2 \leq 1.3$ y $2 \leq x_3 \leq 15$.

6.2.4 Problema 4: Diseño de una armadura de 10 barras [25]

Considere la armadura mostrada en la figura 6.4. El problema consiste en encontrar el momento de inercia de cada miembro de la armadura de manera que minimice su peso, sujeto a restricciones de esfuerzo y desplazamiento. El peso de la armadura esta dado por:

$$f(x) = \sum_{j=1}^{10} \rho\, A_j\, L_j \tag{6.27}$$

Donde x es una solución candidata, A_j es el área de la sección transversal del miembro $j$ésimo. L_j es la longitud del miembro j y ρ es la densidad del peso del material.

Los datos asumidos son: módulo de elasticidad, $E = 1.0 \times 10^4$ ksi (68965.5 MPa), $\rho = 0.10$ lb/in^3 (2768.096 kg/m^3), y una carga de 100 kips (45351.47 Kg) en la dirección negativa del eje y es aplicada en los nodos 2 y 4. El esfuerzo máximo permitido de cada miembro es conocido como σ_a, y se asume como ± 25 ksi (172.41 MPa).

El desplazamiento máximo permitido de cada nodo (vertical y horizontal) se representa como u_a, y se asume como 2 pulgadas (5.08 cm).

Existen 10 restricciones de esfuerzo y 12 de desplazamiento (realmente se pueden asumir sólo 8 restricciones de desplazamiento ya que existen dos nodos con desplazamiento cero. Sin embargo, se toman todas como restricciones del problema). El momento de inercia de cada elemento puede ser diferente, de ahí que el problema tenga 10 variables de diseño.

6.2.5 Problema 5: [119]

Maximizar

$$F(\mathbf{X}) = \left| \frac{\sum_{i=1}^{n} \cos^4(x_i) - 2\prod_{i=1}^{n} \cos^2(x_i)}{\sqrt{\sum_{i=1}^{n} i x_i^2}} \right| \tag{6.28}$$

Sujeta a:

$$g_1(\mathbf{X}) = 0.75 - \prod_{i=1}^{n} x_i \leq 0 \tag{6.29}$$

$$g_2(\mathbf{X}) = \sum_{i=1}^{n} x_i - 7.5n \leq 0 \tag{6.30}$$

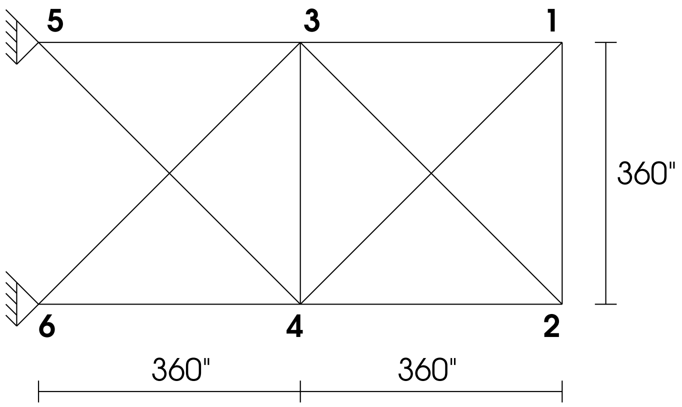

Figura 6.4: Armadura plana de 10 barras usada para el problema 4.

Donde $n = 20$ y $0 \leq x_i \leq 10$ $(i = 1, \ldots, n)$. El óptimo global no se conoce; la mejor solución reportada en la literatura es [119] $f(x^*) = 0.803619$. La restricción g_1 está cerca de ser activa($g_1 = -10^{-8}$).

6.2.6 Problema 6 [119]

Minimizar

$$F(\mathbf{X}) = 5.3578547x_3^2 + 0.8356891x_1x_5 + 37.293239x_1 - 40792.141 \quad (6.31)$$

Sujeta a:

$$g_1(\mathbf{X}) = 85.334407 + 0.0056858x_2x_5 + 0.0006262x_1x_4 - 0.0022053x_3x_5 - 92 \leq 0 \quad (6.32)$$

$$g_2(\mathbf{X}) = -85.334407 - 0.0056858x_2x_5 - 0.0006262x_1x_4 + 0.0022053x_3x_5 \leq 0 \quad (6.33)$$

$$g_3(\mathbf{X}) = 80.51249 + 0.0071317x_2x_5 + 0.0029955x_1x_2 + 0.0021813x_3^2 - 110 \leq 0 \quad (6.34)$$

$$g_4(\mathbf{X}) = -80.51249 - 0.0071317x_2x_5 - 0.0029955x_1x_2 - 0.0021813x_3^2 + 90 \leq 0 \quad (6.35)$$

$$g_5(\mathbf{X}) = 9.300961 + 0.0047026x_3x_5 + 0.0012547x_1x_3 + 0.0019085x_3x_4 - 25 \leq 0 \quad (6.36)$$

$$g_6(\mathbf{X}) = -9.300961 - 0.0047026x_3x_5 - 0.0012547x_1x_3 - 0.0019085x_3x_4 + 20 \leq 0 \quad (6.37)$$

Donde $78 \leq x_1 \leq 102$, $33 \leq x_2 \leq 45$, $27 \leq x_i \leq 45$, $(i = 3, 4, 5)$. El óptimo global está en $x^* = (78, 33, 29.995256025682, 45, 36.775812905788)$ donde $f(x^*) = -30665.539$. Las restricciones g_1 y g_6 están activas.

6.2.7 Problema 7 [119]

Minimizar

$$\begin{aligned} F(\mathbf{X}) = \; & x_1^2 + x_2^2 + x_1x_2 - 14x_1 - 16x_2 + (x_3 - 10)^2 + 4(x_4 - 5)^2 + (x_5 - 3)^2 \\ & +2(x_6 - 1)^2 + 5x_7^2 + 7(x_8 - 11)^2 + 2(x_9 - 10)^2 \\ & +(x_{10} - 7)^2 + 45 \end{aligned} \quad (6.38)$$

Sujeta a:

$$g_1(\mathbf{X}) = -105 + 4x_1 + 5x_2 - 3x_7 + 9x_8 \leq 0 \quad (6.39)$$

$$g_2(\mathbf{X}) = 10x_1 - 8x_2 - 17x_7 + 2x_8 \leq 0 \quad (6.40)$$

$$g_3(\mathbf{X}) = -8x_1 + 2x_2 + 5x_9 - 2x_{10} - 12 \leq 0 \quad (6.41)$$

$$g_4(\mathbf{X}) = 3(x_1 - 2)^2 + 4(x_2 - 3)^2 + 2x_3^2 - 7x_4 - 120 \leq 0 \quad (6.42)$$

$$g_5(\mathbf{X}) = 5x_1^2 + 8x_2 + (x_3 - 6)^2 - 2x_4 - 40 \leq 0 \quad (6.43)$$

$$g_6(\mathbf{X}) = x_1^2 + 2(x_2 - 2)^2 - 2x_1x_2 + 14x_5 - 6x_6 \leq 0 \quad (6.44)$$

$$g_7(\mathbf{X}) = 0.5(x_1 - 8)^2 + 2(x_2 - 4)^2 + 3x_5^2 - x_6 - 30 \leq 0 \quad (6.45)$$

$$g_8(\mathbf{X}) = -3x_1 + 6x_2 + 12(x_9 - 8)^2 - 7x_{10} \leq 0 \quad (6.46)$$

Donde $-10 \leq x_i \leq 10$ $(i = 1, \ldots, 10)$. El óptimo global se encuentra en $x^* = (2.171996, 2.363683, 8.773926, 5.095984, 0.9906548, 1.430574, 1.321644, 9.828726, 8.280092, 8.375927)$ donde $f(x^*) = 24.3062091$. Las restricciones g_1, g_2, g_3, g_4, g_5 y g_6 están activas.

6.2.8 Problema 8 [119]

Minimizar

$$F(\mathbf{X}) = x_1 + x_2 + x_3 \tag{6.47}$$

Sujeta a:

$$g_1(\mathbf{X}) = -1 + 0.0025(x_4 + x_6) \leq 0 \tag{6.48}$$

$$g_2(\mathbf{X}) = -1 + 0.0025(x_5 + x_7 - x_4) \leq 0 \tag{6.49}$$

$$g_3(\mathbf{X}) = -1 + 0.01(x_8 - x_5) \leq 0 \tag{6.50}$$

$$g_4(\mathbf{X}) = -x_1 x_6 + 833.33252 x_4 + 100 x_1 - 83333.333 \leq 0 \tag{6.51}$$

$$g_5(\mathbf{X}) = -x_2 x_7 + 1250 x_5 + x_2 x_4 - 1250 x_4 \leq 0 \tag{6.52}$$

$$g_6(\mathbf{X}) = -x_3 x_8 + 1250000 + x_3 x_5 - 2500 x_5 \leq 0 \tag{6.53}$$

Donde $100 \leq x_1 \leq 10000$, $1000 \leq x_i \leq 10000$, $(i = 2, 3)$, $10 \leq x_i \leq 1000$, $(i = 4, \ldots, 8)$. La solución óptima es $x^* = (579.3167, 1359.943, 5110.071, 182.0174, 295.5985, 217.9799, 286.4162, 395.5979)$, donde $f(x^*) = 7049.3307$. g_1, g_2 y g_3 están activas.

6.2.9 Problema 9 [119]

Maximizar

$$F(\mathbf{X}) = \frac{100 - (x_1 - 5)^2 - (x_2 - 5)^2 - (x_3 - 5)^2}{100} \tag{6.54}$$

Sujeta a:

$$g_1(\mathbf{X}) = (x_1 - p)^2 + (x_2 - q)^2 + (x_3 - r)^2 - 0.0625 \leq 0 \tag{6.55}$$

Donde $0 \leq x_i \leq 10$ $(i = 1, 2, 3)$ y $p, q, r = 1, 2, \ldots, 9$. La zona factible del espacio de búsqueda consiste de 9^3 esferas disjuntas. Un punto (x_1, x_2, x_3) es factible si y sólo si existen p, q, r que satisfagan las restricciones descritas anteriormente (ecuación 6.55). El óptimo global se localiza en $x^* = (5, 5, 5)$ donde $f(x^*) = 1$. La solución se localiza en la zona factible.

6.3 Estimación de la complejidad de cada problema

Con el objeto de tener una medida aproximada de la dificultad para encontrar la zona factible en un problema determinado se utiliza el valor de ρ sugerido por Koziel y Michalewicz [76]. La medida se calcula de la siguiente manera:

$$\rho = |F|/|S| \tag{6.56}$$

Para efectos de este trabajo se generaron $1,000,000$ de individuos de manera aleatoria y se verificó su pertenencia o no a la zona factible.

Problema	**n**	**Tipo de función**	ρ	**DL**	**DN**
1	4	cuadrática	2.6859%	6	1
2	4	cuadrática	39.6762%	3	1
3	3	cuadrática	0.7537%	1	3
4	10	no lineal	46.8070%	0	22
5	20	no lineal	99.9973%	2	0
6	5	cuadrática	27.0079%	4	2
7	10	cuadrática	0.0000%	3	5
8	8	lineal	0.0020%	6	0
9	3	cuadrática	4.7697%	0	9^3

Donde n es el número de variables, DL es el número de desigualdades lineales y DN, el número de las desigualdades no lineales.

6.4 Resultados

La implementación del algoritmo fue realizada en lenguaje C, compilador egcs versión 1.1.2 bajo la plataforma SUN, sistema operativo Solaris 2.7 en una máquina ULTRA 10. Los parámetros utilizados fueron los siguientes:

Representación binaria.

Parámetro	**Valor**
Número máximo de generaciones	400
Tamaño de la población	200
Porcentaje de cruza	0.6
Porcentaje de mutación	0.03
Tamaño del torneo	10
Porcentaje de selección	0.99

Representación real.

Parámetro	**Valor**
Número máximo de generaciones	$10,000$
Tamaño de la población	50
Porcentaje de cruza	0.8
Porcentaje de mutación	0.6
Tamaño del torneo	10
Porcentaje de selección	0.8

Los resultados por problema se presentan de la siguiente manera:

1. El mejor valor reportado en la literatura, incluyendo el valor de la función objetivo, variables y restricciones.

2. El mejor resultado encontrado por el algoritmo para la representación binaria, incluyendo el valor de la función objetivo, variables y restricciones.

3. Estadísticas detalladas de 30 corridas de prueba para la representación binaria, incluyendo el valor de la función objetivo, variables, restricciones, media, desviación estándar, varianza, mediana, mejor y peor solución encontrada.

4. Gráfica de convergencia (aptitud contra generación) para la representación binaria de la corrida ubicada en el valor de la mediana con respecto al valor de la función objetivo, indicada con un &.

5. El mejor resultado encontrado por el algoritmo para la representación real, incluyendo el valor de la función de objetivo, variables y restricciones.

6. Estadísticas detalladas de 30 corridas de prueba para la representación real, incluyendo el valor de la función objetivo, variables, restricciones, media, desviación estándar, varianza, mediana, mejor y peor solución encontrada.

7. Gráfica de convergencia (aptitud contra generación) para la representación real de la corrida ubicada en el valor de la mediana con respecto al valor de la función objetivo, indicada con un &.

6.4.1 Problema 1

El mejor valor reportado previamente en la literatura es [29]: $\mathbf{1.725255}$, con $x = (0.205700, 3.471800, 9.035100, 0.205800)$ y las restricciones con un valor de $g_1 = -0.270305$,$g_2 = -0.139490$, $g_3 = -0.000100$, $g_4 = -3.432597$, $g_5 = -0.080700$, $g_6 = -0.235538$, $g_7 = -5.492326$.

Representación binaria

El mejor valor encontrado por el algoritmo es: $\mathbf{f(x) = 1.728226}$, con $x = (0.205986, 3.471328, 9.020224, 0.206480)$ y las restricciones con un valor de $g_1 = -0.074092$, $g_2 = -0.266227$, $g_3 = -0.000495$, $g_4 = -3.430043$, $g_5 = -0.080986$, $g_6 = -0.235514$ y $g_7 = -58.666440$.

Representación real

El mejor valor encontrado por el algoritmo es: $\mathbf{f(x) = 1.724896}$, con $x = (0.205708, 3.470923, 9.03674, 0.205729)$ y las restricciones con un valor de $g_1 = -0.062627$, $g_2 = -0.730609$, $g_3 = -0.000021$, $g_4 = -3.432928$, $g_5 = -0.080708$, $g_6 = -0.235541$ y $g_7 = -0.025913$.

6.4.2 Problema 2

El mejor valor reportado previamente en la literatura es [29]: $\mathbf{f(x) = 6062.253733}$, con $x = (0.812500, 0.437500, 42.077800, 176.893600)$ y las restricciones con un valor de $g_1 = -0.000398$, $g_2 = -0.036078$, $g_3 = -5.604816$ y $g_4 = -63.106400$.

Diseño de una viga soldada

Representacion Binaria con Código Gray Cruza de 2 puntos y Mutación Uniforme

Prueba	X1	X2	X3	X4	g1(X)	g2(X)	g3(X)	g4(X)	g5(X)	g6(X)	g7(X)	f(X)	Factible
1	0,220608	3,288233	8,726897	0,220608	-0,131568	-2,223607	0,000000	-3,393623	-0,095608	-0,235028	-1228,508419	1,778070	**SI**
2	0,206770	3,425760	9,111661	0,206858	-18,596727	-653,044384	-0,000088	-3,415378	-0,081770	-0,235972	-132,421932	1,741946	**SI**
3	0,206421	3,476653	9,045672	0,208345	-76,409959	-435,787185	-0,001923	-3,410948	-0,081421	-0,235765	-235,825863	1,748241	**SI**
4	0,200661	3,586233	9,037907	0,205728	-11,229263	-8,317692	-0,005067	-3,422634	-0,075661	-0,235546	-0,438461	1,732669	**SI**
5	0,244788	3,036029	8,285433	0,244859	-0,368589	-16,368165	-0,000071	-3,330946	-0,119788	-0,234238	-3539,179536	1,863752	**SI**
6	0,205986	3,471328	9,020224	0,206480	-0,074092	-0,266227	-0,000495	-3,430043	-0,080986	-0,235514	-58,666440	1,728226	**SI**
7	0,228100	3,240858	8,488332	0,233169	-1,100921	-0,419383	-0,005069	-3,352877	-0,103100	-0,234607	-2375,725124	1,827952	**SI**
8	0,213870	3,377781	8,851173	0,214470	-17,386921	-4,092034	-0,000600	-3,408136	-0,088870	-0,235239	-704,990557	1,757754	**SI**
9	0,205775	3,486180	8,995372	0,207632	-2,184216	-1,566654	-0,001856	-3,424324	-0,080775	-0,235475	-149,408449	1,734316	**SI**
10	0,201296	3,584729	9,000603	0,207403	-5,262254	-3,442538	-0,006107	-3,416479	-0,076296	-0,235484	-131,495404	1,739741	**Si**
11	0,256467	2,735634	8,687467	0,256569	-3,413655	-3972,031685	-0,000101	-3,198484	-0,131467	-0,236951	-5336,255618	1,993408	**SI**
12	0,189668	3,855402	9,031761	0,205952	-0,254854	-0,115858	-0,016284	-3,398356	-0,064668	-0,235533	-17,348291	1,751094	**SI**
13	0,179330	4,136699	9,056199	0,205639	-1,147373	-116,319606	-0,026308	-3,371667	-0,054330	-0,235628	-0,558238	1,771929	**SI**
14	0,210303	3,396353	8,979565	0,210308	-0,876260	-278,955296	-0,000005	-3,414832	-0,085303	-0,235584	-382,864760	1,746477	**SI**
15	0,256447	2,931054	8,092579	0,256536	-0,398272	-0,915859	-0,000089	-3,302069	-0,131447	-0,233854	-4790,887489	1,903990	**SI**
16	0,203633	3,512166	9,047144	0,205678	-0,243166	-62,139007	-0,002045	-3,427920	-0,078633	-0,235587	-0,026149	1,728624	**SI**
17	0,243019	3,161053	8,121219	0,255200	-120,142588	-56,151083	-0,012181	-3,282694	-0,118019	-0,233941	-4649,552536	1,917356	**SI**
18	0,201806	3,543519	9,078810	0,206234	-9,763784	-350,807570	-0,004428	-3,415431	-0,076806	-0,235776	-62,724017	1,739728	**SI**
&19	0,206458	3,362405	9,295125	0,206985	-8,846281	-1817,370365	-0,000527	-3,388451	-0,081458	-0,236794	-223,702762	1,765414	**SI**
20	0,187049	4,117086	8,641089	0,225014	-10,802402	-2,579749	-0,037966	-3,301601	-0,062049	-0,234880	-1619,301183	1,853863	**SI**
21	0,218968	3,307752	8,762043	0,219042	-5,176866	-29,584811	-0,000073	-3,396863	-0,093968	-0,235102	-1094,759397	1,773320	**SI**
22	0,188414	3,888171	9,033419	0,205878	-0,655990	-0,302742	-0,017464	-3,395757	-0,063414	-0,235535	-11,564352	1,753009	**SI**
23	0,204904	3,495076	9,026274	0,206223	-8,398621	-3,085319	-0,001319	-3,428865	-0,079904	-0,235525	-38,708915	1,728847	**SI**
24	0,215761	3,434707	8,598738	0,227274	-3,167509	-7,543350	-0,011512	-3,355919	-0,090761	-0,234808	-1825,025689	1,815845	**SI**
25	0,204104	3,511967	9,041452	0,205744	-24,749086	-34,068686	-0,001640	-3,428400	-0,079104	-0,235564	-3,333079	1,728860	**SI**
26	0,276511	2,763972	7,804812	0,276534	-0,196324	-80,325355	-0,000024	-3,251296	-0,151511	-0,233303	-7172,940492	1,974155	**SI**
27	0,222368	3,267007	8,695379	0,222368	-0,036700	-23,437230	0,000000	-3,388574	-0,097368	-0,234985	-1384,815715	1,784708	**SI**
28	0,220183	3,303086	8,748869	0,220187	-49,994702	-95,599728	-0,000004	-3,391300	-0,095183	-0,235112	-1199,339414	1,780527	**SI**
29	0,236301	3,122563	8,423247	0,236783	-0,102924	-0,007223	-0,000482	-3,351167	-0,111301	-0,234487	-2725,033145	1,835602	**SI**
30	0,182765	4,044560	9,044876	0,205852	-11,911150	-72,571931	-0,023087	-3,380135	-0,057765	-0,235588	-14,351533	1,765615	**SI**

MEDIA	**1,792654**	
DESV. EST.	**0,074713**	
MEJOR	**1,728226**	**SI**
PEOR	**1,993408**	**SI**
MEDIANA	**1,765414**	**SI**
VARIANZA	**0,005582**	

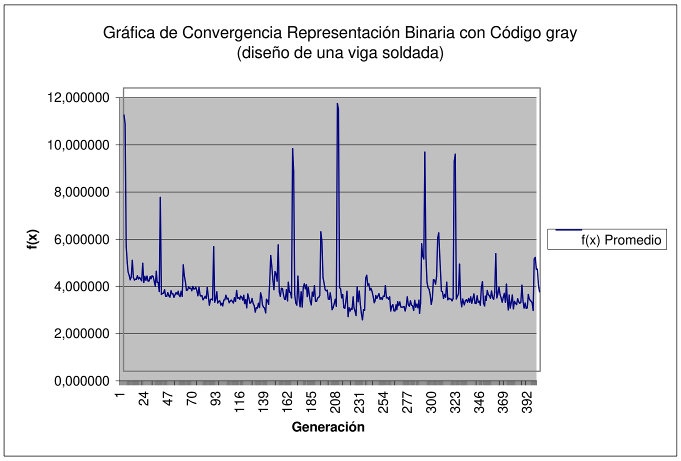

Figura 6.5: Resultados de 30 corridas y gráfica de convergencia para el problema 1 (representación binaria).

Diseño de una viga soldada

Representacion Real SBX y Parameter Based Mutation

Prueba	X1	X2	X3	X4	g1(X)	g2(X)	g3(X)	g4(X)	g5(X)	g6(X)	g7(X)	f(X)	Factible
1	0,306862	2,558301	7,397627	0,306993	-0,012872	-0,26028	-0,000131	-3,181002	-0,181862	-0,232337	-11334,38578	2,075263	**SI**
2	0,332337	2,408379	7,110039	0,332338	-0,431016	-1,028288	-0,000001	-3,123116	-0,207337	-0,231623	-15353,96935	2,159172	**SI**
3	0,331592	2,412256	7,118419	0,331598	-0,110811	-4,79788	-0,000006	-3,124689	-0,206592	-0,231647	-15230,29175	2,156806	**SI**
4	0,278127	2,755306	7,771977	0,278141	-0,007325	-1,345942	-0,000015	-3,249351	-0,153127	-0,233188	-7363,454763	1,978002	**SI**
5	0,372137	2,211268	6,719005	0,372149	-0,003289	-1,213985	-0,000012	-3,035321	-0,247137	-0,230553	-22726,32065	2,288473	**SI**
6	0,475695	1,835322	5,942553	0,475741	-0,318857	-0,516859	-0,000046	-2,822503	-0,350695	-0,228012	-48513,37266	2,612597	**SI**
7	0,205712	3,470681	9,037187	0,205728	-0,070272	-3,524417	-0,000016	-3,432881	-0,080712	-0,235543	-0,116334	1,724936	**SI**
8	0,319547	2,480757	7,25071	0,319557	-0,01366	-0,011154	-0,00001	-3,152172	-0,194547	-0,231979	-13263,1874	2,116971	**SI**
9	0,302400	2,586412	7,453806	0,302405	-1,167939	-2,481192	-0,000005	-3,191740	-0,177400	-0,232471	-10661,14739	2,059966	**SI**
10	0,273719	2,788712	7,834849	0,273719	-0,049519	-3,964629	-0,000001	-3,259992	-0,148719	-0,233325	-6809,96643	1,962976	**Si**
11	0,568898	2,088445	4,125032	0,987327	-0,092999	-0,433188	-0,418429	-1,813738	-0,443898	-0,218324	-353099,8668	3,899063	**SI**
12	0,323852	2,623028	6,745959	0,369172	-0,079824	-0,457084	-0,04532	-2,997347	-0,198852	-0,230631	-22128,42396	2,295581	**SI**
13	0,228667	3,199525	8,571474	0,228681	-0,011278	-9,391151	-0,000009	-2,388781	-0,590519	-0,223046	-150879,3867	3,310248	**SI**
14	0,57412	1,608303	5,331615	0,591024	-0,34965	-0,943269	-0,016904	-2,599266	-0,44912	-0,225493	-89720,07766	2,951849	**SI**
15	0,380474	2,8357	5,14115	0,635633	-0,002945	-1,248395	-0,255159	-2,337966	-0,255474	-0,224585	-109544,4939	3,100356	**SI**
16	0,715519	1,330748	4,846292	0,715527	-0,011278	-9,391151	-0,000009	-2,388781	-0,590519	-0,223046	-150879,3867	3,310248	**SI**
17	0,50614	1,750174	5,760093	0,506356	-0,058975	-0,409516	-0,000216	-2,763104	-0,38114	-0,227315	-58105,66996	2,705375	**SI**
&18	0,311383	2,530953	7,345264	0,311387	-5,731397	-0,380586	-0,000004	-3,170814	-0,186383	-0,232211	-11994,90849	2,090128	**SI**
19	0,224263	3,247664	8,654339	0,22431	-1,107912	-0,558454	-0,000048	-3,383906	-0,099263	-0,234902	-1555,882118	1,791268	**SI**
20	0,232542	3,158128	8,499828	0,232546	-0,939643	-1,370961	-0,000004	-3,362698	-0,107542	-0,234628	-2316,477318	1,820300	**SI**
21	0,205708	3,470923	9,03674	0,205729	-0,062627	-0,730609	-0,000021	-3,432928	-0,080708	-0,235541	-0,025913	1,724896	**SI**
22	0,288962	2,676506	7,625393	0,288965	-0,013443	-4,20733	-0,000003	-3,223395	-0,163962	-0,232867	-8780,400712	2,014750	**SI**
23	0,232973	3,152896	8,49334	0,232981	-0,080686	-11,633536	-0,000008	-3,36137	-0,107973	-0,234621	-2358,880906	1,821993	**SI**
24	0,322775	2,4625	7,212969	0,322916	-0,141394	-0,620828	-0,000141	-3,14435	-0,197775	-0,231885	-13800,15052	2,128157	**SI**
25	0,206372	3,462684	9,022836	0,206377	-2,125268	-2,563841	-0,000004	-3,431134	-0,081372	-0,235519	-50,695082	1,727323	**SI**
26	0,39883	2,099559	6,490079	0,398923	-5,692093	-5,537621	-0,000093	-2,978001	-0,27383	-0,22987	-28450,31847	2,374280	**SI**
27	0,215281	3,351203	8,833557	0,215299	-0,75346	-0,209213	-0,000018	-3,407545	-0,090281	-0,235208	-773,972932	1,759179	**SI**
28	0,351282	2,309747	6,915315	0,351308	-0,043913	-0,186329	-0,000026	-3,080817	-0,226282	-0,231105	-18701,27801	2,221128	**SI**
29	0,255884	2,935437	8,102682	0,255892	-0,008332	-0,304266	-0,000008	-3,303806	-0,130884	-0,233874	-4719,155973	1,901666	**SI**
30	0,205735	3,470623	9,036466	0,205739	-0,581967	-0,362997	-0,000004	-3,432925	-0,080735	-0,23554	-0,77959	1,724925	**SI**

MEDIA	**2,260263**	
DESV. EST.	**0,555012**	
MEJOR	**1,724896**	**SI**
PEOR	**3,899063**	**SI**
MEDIANA	**2,103550**	**SI**
VARIANZA	**0,308038**	

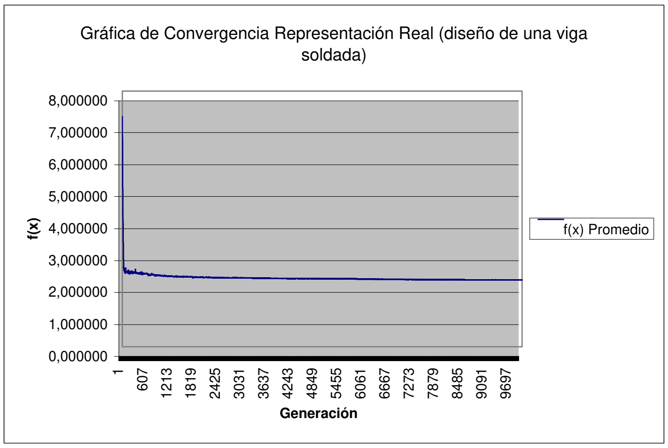

Figura 6.6: Resultados de 30 corridas y gráfica de convergencia para el problema 1 (representación real).

Representación binaria

El mejor valor encontrado por el algoritmo es: $\mathbf{f(x) = 6059.946341}$, con $x = (0.812500, 0.437500, 42.097398, 176.654047)$ y las restricciones con un valor de $g_1 = -0.000020$, $g_2 = -0.035891$, $g_3 = -27.886075$ y $g_4 = -63.345953$.

Representación real

El mejor valor encontrado por el algoritmo es: $\mathbf{f(x) = 6092.012458}$, con $x = (0.875000, 0.437500, 45.325684, 140.375071)$ y las restricciones con un valor de $g_1 = -0.000214$, $g_2 = -0.005093$, $g_3 = -55.121733$ y $g_4 = -99.624929$.

6.4.3 Problema 3

El mejor valor reportado previamente en la literatura es [27]: $\mathbf{f(x) = 0.0127047834}$, con $x = (0.051480, 0.351661, 11.632201)$ y las restricciones con un valor de $g_1 = -0.003337$, $g_2 = -0.000110$, $g_3 = -4.026318$ y $g_4 = -0.731239$.

Representación binaria

El mejor valor encontrado por el algoritmo es: $\mathbf{f(x) = 0.012681}$, con $x = (0.051989, 0.363965, 10.890522)$ y las restricciones con un valor de $g_1 = -0.000013$, $g_2 = -0.000021$, $g_3 = -4.061338$ y $g_4 = -0.722698$.

Representación real

El mejor valor encontrado por el algoritmo es: $\mathbf{f(x) = 0.012710}$, con $x = (0.052289, 0.052289, 10.527731)$ y las restricciones con un valor de $g_1 = -0.001078$, $g_2 = -0.000562$, $g_3 = -4.066350$ y $g_4 = -0.717763$.

6.4.4 Problema 4

El mejor valor reportado previamente en la literatura es [24]: $\mathbf{f(x) = 5153.61}$, con $x = (979.7582, 0.1039, 539.4210, 208.7898, 0.1087, 0.1243, 69.6764, 483.3090, 44.7585, 0.1036)$, la cual es factible.

Diseño de un recipiente de presión

Representacion Binaria con Código Gray. Curza de 2 puntos y Mutación Uniforme

Prueba	X1	X2	X3	X4	g1(X)	g2(X)	g3(X)	g4(X)	f(X)	Factible
1	0,875000	0,437500	45,188151	141,774305	-0,002869	-0,006405	-1,563606	-98,225695	6107,551462	**SI**
2	0,875000	0,500000	45,197827	141,686710	-0,002682	-0,068813	-77,251003	-98,313290	6333,781119	**SI**
3	0,812500	0,437500	42,024501	177,554525	-0,001427	-0,036586	-0,118478	-62,445475	6068,727923	**SI**
4	0,937500	0,500000	47,840980	116,877788	-0,014169	-0,043597	-3052,998751	-123,122212	6456,953164	**SI**
5	0,875000	0,437500	44,605976	147,873115	-0,014105	-0,011959	-93,416521	-92,126885	6176,030459	**SI**
6	0,875000	0,437500	44,789144	145,921751	-0,010570	-0,010212	-0,868573	-94,078249	6153,973314	**SI**
7	0,812500	0,437500	42,051365	177,221615	-0,000909	-0,036330	-7,132946	-62,778385	6065,473626	**SI**
8	0,875000	0,437500	45,238989	141,290479	-0,001888	-0,005920	-244,301490	-98,709521	6102,732080	**SI**
9	0,875000	0,500000	45,197827	141,686710	-0,002682	-0,068813	-77,251003	-98,313290	6333,781119	**SI**
10	1,000000	0,500000	51,567057	86,381727	-0,004756	-0,008050	-24,520032	-153,618273	6433,161591	**Si**
11	0,812500	0,437500	41,159816	188,627462	-0,018116	-0,044835	-13,664149	-51,372538	6177,430630	**SI**
12	0,812500	0,437500	42,097398	176,654047	-0,000020	-0,035891	-27,886075	-63,345953	6059,946341	**SI**
13	0,812500	0,437500	41,680804	181,889306	-0,008060	-0,039865	-45,697910	-58,110694	6111,414573	**SI**
14	0,875000	0,437500	45,257432	141,825941	-0,001532	-0,005744	-4905,232603	-98,174059	6120,225403	**SI**
15	0,875000	0,437500	43,202246	163,422222	-0,041197	-0,025351	-1,086579	-76,577778	6349,307800	**SI**
16	0,812500	0,437500	41,868650	179,979646	-0,004435	-0,038073	-2616,487839	-60,020354	6098,937591	**SI**
17	0,812500	0,437500	41,895123	179,176604	-0,003924	-0,037821	-25,775612	-60,823396	6084,726865	**SI**
18	0,875000	0,437500	44,890105	144,908038	-0,008621	-0,009248	-284,893583	-95,091962	6143,333896	**SI**
&19	0,875000	0,437500	45,306857	141,271491	-0,000578	-0,005273	-4597,599327	-98,728509	6113,251114	**SI**
20	0,812500	0,437500	41,880095	179,361926	-0,004214	-0,037964	-6,899673	-60,638074	6086,501186	**SI**
21	0,875000	0,437500	44,824928	145,569362	-0,009879	-0,009870	-149,320285	-94,430638	6150,398535	**SI**
22	0,812500	0,437500	42,087406	176,773452	-0,000213	-0,035986	-3,049954	-63,226548	6061,059397	**SI**
23	0,875000	0,437500	44,991608	143,809572	-0,006662	-0,008280	-28,791697	-96,190428	6130,405333	**SI**
24	0,812500	0,437500	42,032096	177,463940	-0,001281	-0,036514	-22,071452	-62,536060	6067,891267	**SI**
25	0,812500	0,437500	42,025257	177,624536	-0,001413	-0,036579	-440,789340	-62,375464	6070,489345	**SI**
26	0,875000	0,437500	43,740201	157,304360	-0,030814	-0,020218	-17,325897	-82,695640	6281,178332	**SI**
27	0,875000	0,437500	42,985464	165,946266	-0,045381	-0,027419	-1,381760	-74,053734	6377,392663	**SI**
28	0,875000	0,437500	45,143559	142,232957	-0,003729	-0,006830	-0,795912	-97,767043	6112,685385	**SI**
29	1,000000	0,500000	51,187883	89,199916	-0,012074	-0,011668	-71,817666	-150,800084	6469,322010	**SI**
30	0,875000	0,437500	45,198480	141,668105	-0,002669	-0,006306	-0,900726	-98,331895	6106,360563	**SI**

MEDIA	**6177,253268**	
DESV. EST.	**130,929702**	
MEJOR	**6059,946341**	**SI**
PEOR	**6469,322010**	**SI**
MEDIANA	**6116,738259**	**SI**
VARIANZA	**17142,586857**	

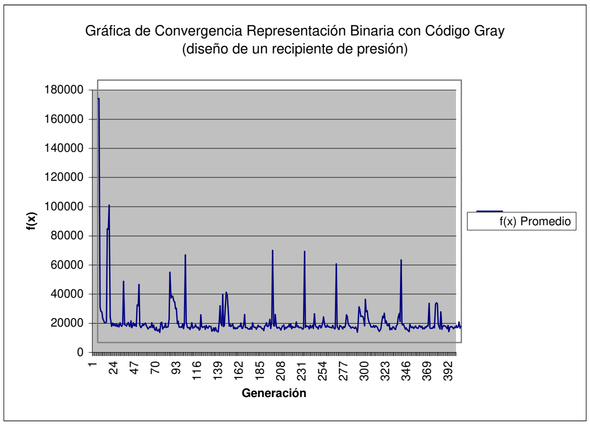

Figura 6.7: Resultados de 30 corridas y gráfica de convergencia para el problema 2 (representación binaria).

Diseño de un recipiente de presión
Representacion Real SBX y Parameter Based Mutation

Prueba	X1	X2	X3	X4	g1(X)	g2(X)	g3(X)	g4(X)	f(X)	Factible
1	0,937500	0,500000	48,404997	111,537227	-0,003284	-0,038216	-87,889400	-128,462773	6387,812944	**SI**
2	1,250000	0,625000	64,517876	13,094343	-0,004805	-0,009499	-177,049330	-226,905657	7348,002560	**SI**
3	1,000000	0,500000	51,415413	87,500991	-0,007683	-0,009497	-29,470646	-152,499009	6447,477131	**SI**
4	1,125000	0,562500	58,230755	44,187767	-0,001146	-0,006979	-1793,513454	-195,812233	6832,349774	**SI**
5	0,875000	0,437500	42,696770	169,361956	-0,050952	-0,030173	-9,500504	-70,638044	6415,378069	**SI**
6	1,062500	0,562500	54,989444	63,115225	-0,001204	-0,037901	-85,609580	-176,884775	6776,756181	**SI**
7	1,187500	0,625000	61,463030	27,252705	-0,001264	-0,038643	-29,948398	-212,747295	7277,484292	**SI**
8	0,875000	0,437500	45,221712	141,499732	-0,002221	-0,006085	-450,821232	-98,500268	6105,585219	**SI**
9	1,250000	0,625000	64,579427	12,820300	-0,003617	-0,008912	-136,104674	-227,179700	7344,243865	**SI**
10	0,875000	0,437500	45,154680	142,141866	-0,003515	-0,006724	-150,902133	-97,858134	6112,036089	**SI**
11	0,937500	0,500000	47,492870	119,659534	-0,020888	-0,046918	-639,684857	-120,340466	6482,466693	**SI**
12	0,875000	0,437500	45,325684	140,375071	-0,000214	-0,005093	-55,121733	-99,624929	6092,012458	**SI**
13	1,062500	0,562500	55,010437	63,059392	-0,000799	-0,037700	-810,738567	-176,940608	6778,181654	**SI**
14	1,125000	0,562500	58,289977	43,694259	-0,000003	-0,006414	-9,642302	-196,305741	6820,451107	**SI**
15	1,125000	0,562500	56,484955	54,025436	-0,034840	-0,023634	-417,796519	-185,974564	6962,698056	**SI**
16	1,125000	0,562500	57,954535	45,625103	-0,006477	-0,009614	-790,775021	-194,374897	6848,855437	**SI**
17	1,000000	0,500000	51,505910	86,838681	-0,005936	-0,008634	-81,615795	-153,161319	6439,153046	**SI**
18	1,000000	0,500000	50,310687	95,903870	-0,029004	-0,020036	-40,158317	-144,096130	6555,211301	**SI**
19	0,937500	0,500000	48,451164	111,154357	-0,002393	-0,037776	-191,774967	-128,845643	6383,708585	**SI**
&20	1,000000	0,500000	51,052622	90,404720	-0,014684	-0,012958	-1619,731455	-149,595280	6488,931642	**SI**
21	1,000000	0,500000	50,929597	91,187380	-0,017059	-0,014132	-414,351155	-148,812620	6495,701241	**SI**
22	0,875000	0,437500	44,489851	149,182852	-0,016346	-0,013067	-535,372192	-90,817148	6191,774911	**SI**
23	1,125000	0,562500	56,448724	54,210515	-0,035540	-0,023979	-124,367915	-185,789485	6964,382229	**SI**
24	1,000000	0,500000	51,799959	84,683835	-0,000261	-0,005828	-62,955780	-155,316165	6411,591120	**SI**
25	0,875000	0,437500	45,219183	141,458805	-0,002270	-0,006109	-21,211915	-98,541195	6104,066813	**SI**
26	0,812500	0,437500	41,837039	179,958212	-0,005045	-0,038375	-306,677995	-60,041788	6093,089798	**SI**
27	1,062500	0,562500	54,948131	63,367293	-0,002001	-0,038295	-7,386373	-176,632707	6779,624380	**SI**
28	1,000000	0,500000	50,949291	91,007652	-0,016679	-0,013944	-165,611958	-148,992348	6492,725118	**SI**
29	1,000000	0,500000	51,725861	85,217345	-0,001691	-0,006535	-11,697399	-154,782655	6418,260715	**SI**
30	0,937500	0,437500	45,849668	135,130526	-0,052601	-0,000094	-171,976482	-104,869474	6426,051525	**SI**

MEDIA	**6582,353483**	
DESV. EST.	**367,275062**	
MEJOR	**6092,012458**	**SI**
PEOR	**7348,002560**	**SI**
MEDIANA	**6488,931642**	**SI**
VARIANZA	**134890,971267**	

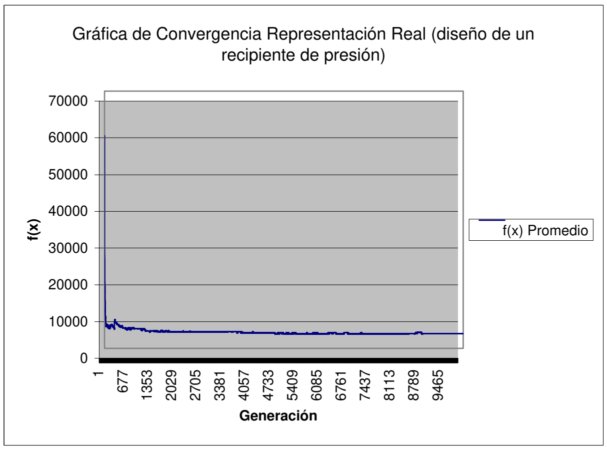

Figura 6.8: Resultados de 30 corridas y gráfica de convergencia para el problema 2 (representación real).

Minimización del peso de un resorte

Representacion Binaria con Código Gray. Cruza de 2 puntos y Mutación Uniforme

Prueba	X1	X2	X3	g1(X)	g2(X)	g3(X)	g4(X)	f(X)	Factible
1	0,051989	0,363965	10,890522	-0,000013	-0,000021	-4,061338	-0,722698	0,012681	**SI**
2	0,052679	0,380621	10,040517	-0,000258	-0,000809	-4,086463	-0,711133	0,012718	**SI**
3	0,050003	0,316783	14,134992	-0,000057	-0,001749	-3,951055	-0,755476	0,012780	**SI**
4	0,051558	0,353562	11,643618	-0,013228	-0,000048	-3,975116	-0,729920	0,012823	**SI**
5	0,053946	0,409276	8,891723	-0,001414	-0,008395	-4,087025	-0,691185	0,012973	**SI**
6	0,052148	0,367832	10,685075	-0,000461	-0,000056	-4,066200	-0,720013	0,012689	**SI**
7	0,051896	0,361514	11,039572	-0,000502	-0,000447	-4,051865	-0,724393	0,012696	**SI**
8	0,052825	0,384620	9,860151	-0,002415	-0,000089	-4,086433	-0,708370	0,012729	**SI**
9	0,052131	0,367025	10,739962	-0,000279	-0,000930	-4,060876	-0,720563	0,012707	**SI**
10	0,050139	0,319464	13,934456	-0,000212	-0,002668	-3,951752	-0,753598	0,012797	**Si**
11	0,051498	0,351740	11,674406	-0,004991	-0,000907	-4,007642	-0,731175	0,012756	**SI**
12	0,050342	0,325019	13,445638	-0,000002	-0,000370	-3,978005	-0,749759	0,012723	**SI**
13	0,051454	0,350859	11,668753	-0,000365	-0,000540	-4,030993	-0,731791	0,012697	**SI**
14	0,050092	0,319486	13,877292	-0,000019	-0,000030	-3,966854	-0,753614	0,012728	**SI**
15	0,050418	0,326907	13,297253	-0,000233	-0,000010	-3,983101	-0,748449	0,012712	**SI**
16	0,050475	0,327945	13,259427	-0,002395	-0,000640	-3,971336	-0,747720	0,012750	**SI**
17	0,052430	0,374521	10,347221	-0,000844	-0,000598	-4,073675	-0,715366	0,012712	**SI**
18	0,052270	0,370823	10,522873	-0,000128	-0,000055	-4,073461	-0,717938	0,012687	**SI**
19	0,052597	0,378928	10,113933	-0,000386	-0,000061	-4,086857	-0,712317	0,012699	**SI**
20	0,053078	0,391038	9,566579	-0,002737	-0,000034	-4,096110	-0,703923	0,012742	**SI**
21	0,052329	0,372177	10,469564	-0,001431	-0,000301	-4,068027	-0,716996	0,012708	**SI**
22	0,051264	0,346484	12,018284	-0,007108	-0,000197	-3,990253	-0,734835	0,012764	**SI**
&23	0,052264	0,370511	10,574960	-0,002968	-0,000438	-4,056450	-0,718150	0,012727	**SI**
24	0,050001	0,317424	14,076076	-0,002094	-0,000057	-3,951538	-0,755050	0,012758	**SI**
25	0,050883	0,336425	12,679503	-0,002046	-0,002870	-3,979869	-0,741794	0,012786	**SI**
26	0,051197	0,344021	12,233942	-0,008728	-0,002242	-3,966248	-0,736522	0,012835	**SI**
27	0,052306	0,371510	10,520718	-0,002705	-0,000505	-4,059260	-0,717456	0,012726	**SI**
28	0,051938	0,362308	11,039522	-0,003863	-0,000939	-4,033823	-0,723836	0,012744	**SI**
29	0,051878	0,361215	11,049592	-0,000285	-0,000160	-4,053934	-0,724605	0,012686	**SI**
30	0,050601	0,330919	13,015930	-0,001006	-0,000420	-3,986088	-0,745654	0,012723	**SI**

MEDIA	**0,012742**	
DESV. EST.	**0,000059**	
MEJOR	**0,012681**	**SI**
PEOR	**0,012973**	**SI**
MEDIANA	**0,012727**	**SI**
VARIANZA	**0,000000**	

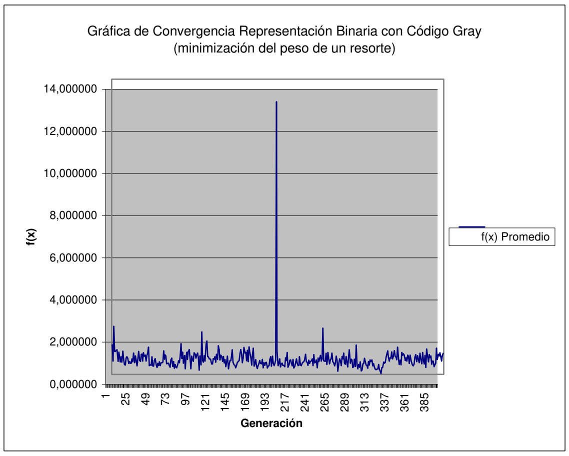

Figura 6.9: Resultados de 30 corridas y gráfica de convergencia para el problema 3 (representación binaria).

Minimización del peso de un resorte
Representacion Real SBX y Parameter Based Mutation

Prueba	X1	X2	X3	g1(X)	g2(X)	g3(X)	g4(X)	f(X)	Factible
1	0,066587	0,831644	2,459809	-0,001336	-0,000642	-4,497113	-0,401180	0,016445	**SI**
2	0,054260	0,421369	8,335779	-0,000995	-0,000813	-4,149096	-0,682914	0,012822	**SI**
3	0,052289	0,371066	10,527731	-0,001078	-0,000562	-4,066350	-0,717763	0,012710	**SI**
4	0,062675	0,682140	3,499642	-0,001569	-0,001281	-4,405641	-0,503456	0,014737	**SI**
5	0,052950	0,387809	9,689367	-0,000232	-0,000030	-4,103380	-0,706160	0,012710	**SI**
6	0,061008	0,624056	4,148099	-0,012520	-0,001458	-4,304083	-0,543291	0,014280	**SI**
7	0,050000	0,317425	14,045374	0,000000	0,000000	-3,962211	-0,755050	0,012733	**SI**
8	-	-	-	-	-	-	-	-	**NO**
9	0,053234	0,395012	9,387096	-0,002347	-0,000064	-4,104602	-0,701169	0,012747	**SI**
10	0,069870	0,955770	2,056088	-0,047987	-0,015188	-4,224750	-0,316240	0,018925	**Si**
11	0,054205	0,415338	9,294287	-0,073182	-0,009808	-3,748377	-0,686971	0,013783	**SI**
12	0,122127	1,031052	14,711597	-0,008497	-0,788537	-0,096764	-0,231214	0,256994	**SI**
13	0,051428	0,349103	11,821139	-0,000337	-0,003123	-4,013654	-0,732979	0,012761	**SI**
&14	0,057143	0,500663	6,114812	-0,001340	-0,003404	-4,236159	-0,628129	0,013266	**SI**
15	0,052878	0,385005	9,849350	-0,000305	-0,002096	-4,086926	-0,708078	0,012756	**SI**
16	0,070422	0,981667	2,078718	-0,112417	-0,013928	-3,937517	-0,298607	0,019857	**SI**
17	0,061699	0,646866	3,848123	-0,000023	-0,002549	-4,381704	-0,527623	0,014401	**SI**
18	0,053100	0,389685	9,657596	-0,000139	-0,004026	-4,085328	-0,704810	0,012809	**SI**
19	0,067057	0,849499	2,503302	-0,055935	-0,002041	-4,213503	-0,388963	0,017202	**SI**
20	-	-	-	-	-	-	-	-	**NO**
21	0,059678	0,579410	4,692384	-0,001187	-0,002530	-4,320721	-0,573941	0,013810	**SI**
22	0,053932	0,411334	8,774126	-0,004237	-0,003534	-4,102366	-0,689823	0,012890	**SI**
23	0,050000	0,317425	14,045384	-0,000001	0,000000	-3,962208	-0,755050	0,012733	**SI**
24	0,056562	0,482765	6,539599	-0,000154	-0,005075	-4,212283	-0,640449	0,013190	**SI**
25	0,067599	0,873303	2,256816	-0,001512	-0,000596	-4,516136	-0,372732	0,016987	**SI**
26	-	-	-	-	-	-	-	-	**NO**
27	0,050447	0,327438	13,268872	-0,000675	-0,000321	-3,980437	-0,748077	0,012724	**SI**
28	0,066966	0,826222	2,564968	-0,000894	-0,022610	-4,371527	-0,404542	0,016914	**SI**
29	0,058098	0,529138	5,566210	-0,007001	-0,003658	-4,235872	-0,608509	0,013514	**SI**
30	-	-	-	-	-	-	-	-	**NO**

MEDIA	**0,023642**	
DESV. EST.	**0,047640**	
MEJOR	**0,012710**	**SI**
PEOR	**0,256994**	**SI**
MEDIANA	**0,013390**	**SI**
VARIANZA	**0,002270**	

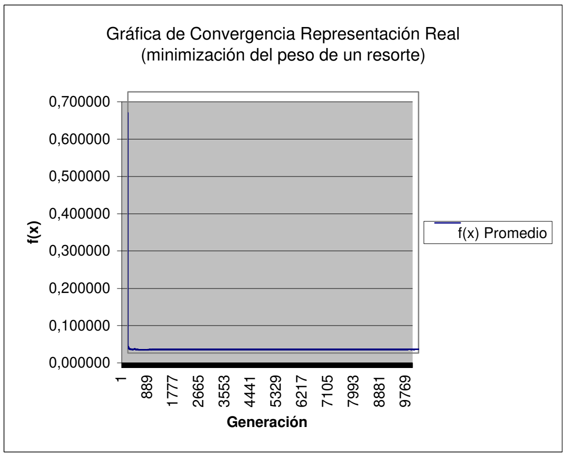

Figura 6.10: Resultados de 30 corridas y gráfica de convergencia para el problema 3 (representación real).

Representación binaria

El mejor valor encontrado por el algoritmo es: $\mathbf{f(x) = 5157.685516}$, con $x = (985.808351, 0.105877, 519.966658, 188.576078, 0.102124, 0.137725, 690.171450, 495.366009, 467.438050, 0.135133)$, la cual es factible.

Representación real

El mejor valor encontrado por el algoritmo es: $\mathbf{f(x) = 5152.636136}$, con $x = (856.013342, 0.100000, 601.548206, 172.969615, 0.100000, 0.000081, 65.334218, 539.767385, 474.838781, 0.100000)$, la cual es factible.

6.4.5 Problema 5

El mejor valor reportado previamente en la literatura es [119]:
$\mathbf{f(x) = 0.80361903586843}$, con $x = (3.16237443645701, 3.12819975856112, 3.09481384891456, 3.06140284777302, 3.02793443337239, 2.99385691314995, 2.95870651588255, 2.92182183591092, 0.49455118612682, 0.48849305858571, 0.48250798063845, 0.47695629293225, 0.47108462715587, 0.46594074852233, 0.46157984137635, 0.45721400967989, 0.45237696886802, 0.44805875597713, 0.44435772435707, 0.44019839654132)$, y las restricciones con un valor de $g_1 = -0.000000011181$ y $g_2 = -1.20067569819217$.

Representación binaria

El mejor valor encontrado por el algoritmo es: $\mathbf{f(x) = 0.787933}$, con $x = (3.088605, 3.034861, 3.112081, 3.079128, 3.090021, 2.949984, 2.929546, 2.905113, 0.485143, 1.372498, 0.421162, 0.479062, 0.406244, 0.502223, 0.489684, 0.378411, 0.325555, 0.345997, 0.519377, 0.364493)$, y las restricciones con un valor de $g_1 = -0.004704$, $g_2 = -119.720812$.

Representación real

El mejor valor encontrado por el algoritmo es: $\mathbf{f(x) = 0.577987}$, con $x = (6.256161, 3.138469, 0.221302, 3.107984, 3.067729, 0.405324, 3.067875, 0.372948, 1.585459, 0.380775, 0.469965, 3.276084, 0.388189, 0.292191, 1.289642, 0.367375, 2.899132, 0.242646, 0.392249, 2.834611)$, y las restricciones con un valor de $g_1 = -0.000626$, $g_2 = -115.943890$.

Diseño de una armadura de 10 barras
Representacion Binaria con Código Gray Cruza de 2 puntos y Mutación Uniforme

Prueba	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	f(X)	Factible
1	745,783852	0.221669	579,478677	368,481268	0.230879	0.163417	647,270980	505,002728	446,273931	0.182063	5.240,526554	**SI**
2	702,009249	0.345167	725,008150	246,551559	0.135388	0.177541	593,594700	509,895341	490,143387	0.105666	5.223,579567	**SI**
3	928,879756	0.536538	760,959898	224,691696	0.105628	0.124506	637,986270	396,432723	450,116515	0.984668	5.231,666629	**SI**
4	792,937518	0.113842	502,202620	148,422929	0.103442	0.148046	726,406990	630,175834	553,366502	0.113738	5.223,418629	**SI**
5	971,530676	0.205408	514,402256	270,322495	0.100024	0.506560	705,572240	441,827144	446,638460	0.198456	5.180,247302	**SI**
6	857,127937	0.270728	414,141444	222,124388	0.103272	0.222779	802,276060	623,676472	461,959557	0.186973	5.214,140848	**SI**
7	851,849818	0.224277	402,770728	323,788947	0.132483	1,110655	579,356010	568,194928	491,794798	0.102277	5.235,495381	**SI**
8	849,172886	0.101141	893,459878	145,551180	0.102165	0.253060	549,914640	496,394874	470,533292	0.104703	5.233,204699	**SI**
9	985,808351	0.105877	519,966658	188,576078	0.102124	0.137725	690,171450	495,366009	467,438050	0.135133	5.157,685516	**SI**
10	819,592334	0.164389	663,417092	185,296895	0.129416	0.349358	597,214070	519,560291	487,457020	0.115593	5.192,023046	**Si**
11	972,994557	0.340009	525,106929	223,987879	0.131116	0.153015	680,978110	432,954102	500,223986	0.125826	5.170,996297	**SI**
12	942,417345	0.193118	567,184435	242,544407	0.166100	1,351268	614,377720	470,389262	449,207560	0.128867	5.196,096742	**SI**
&13	925,798859	0.222363	475,557021	243,532225	0.106199	0.666105	609,654080	544,336122	455,922515	0.170938	5.193,794057	**SI**
14	862,494556	0.149127	535,343675	190,941565	0.100000	0.121412	659,779160	478,140090	564,504211	0.105891	5.178,479598	**SI**
15	931,197876	0.195309	519,386003	262,909733	0.103456	0.112745	784,476650	401,081584	513,542929	0.124146	5.184,509974	**SI**
16	907,566978	0.316939	623,410896	307,290599	0.100900	0.112315	650,630320	380,170212	470,123000	0.109054	5.182,254554	**SI**
17	866,955615	0.192838	587,883964	277,273636	0.100070	0.536246	773,886700	370,638614	533,410263	2,075572	5.263,082609	**SI**
18	873,837859	0.111898	605,263098	300,303389	0.109676	0.134195	676,046540	439,987203	429,774643	0.161377	5.173,294393	**SI**
19	882,291117	0.145370	575,160236	213,207171	0.100914	0.104455	696,796780	506,350902	445,174535	0.134882	5.158,647897	**SI**
20	940,726896	0.231477	534,614806	270,206941	0.138821	0.618957	787,141540	426,958438	462,832839	0.109836	5.203,215910	**SI**
21	898,218173	0.224737	551,021885	222,639697	0.107776	0.193944	721,388240	519,757730	431,159405	0.100014	5.172,249518	**SI**
22	991,616778	0.182372	579,958271	287,777147	0.112788	0.325754	626,601640	464,426787	386,572505	0.147664	5.180,061015	**SI**
23	804,684335	0.104034	552,929910	210,219264	0.104913	0.114880	745,093380	565,763968	455,303120	0.102627	5.178,260180	**SI**
24	719,748652	0.795729	627,539560	246,078070	0.115270	0.143614	714,491490	626,844157	423,290801	0.392016	5.274,693439	**SI**
25	887,609326	0.121527	447,240446	301,052631	0.103684	0.101924	769,151250	531,949734	424,465441	0.100779	5.179,940623	**SI**
26	934,264153	0.108630	604,826163	219,567404	0.103776	0.110244	670,774440	463,457177	440,064168	0.239974	5.160,529914	**SI**
27	946,952615	0.267480	523,333678	259,288604	0.108788	0.116317	770,219530	370,499051	555,507626	0.137542	5.199,432983	**SI**
28	850,514105	0.432172	616,376982	175,595297	0.106665	0.987939	579,425260	526,716784	502,903973	0.101326	5.205,815116	**SI**
29	840,320752	0.105208	666,423496	245,815210	0.101259	0.169845	606,470910	370,556213	585,319095	0.120950	5.201,273570	**SI**
30	890,886857	0.125472	627,550179	264,822377	0.108090	2,118536	534,714660	504,452173	405,595981	0.100320	5.196,401627	**SI**

MEDIA	**5198,085918**	
DESV. EST.	**29,496938**	
MEJOR	**5157,685516**	**SI**
PEOR	**5274,693439**	**SI**
MEDIANA	**5193,794057**	**SI**
VARIANZA	**870,069381**	

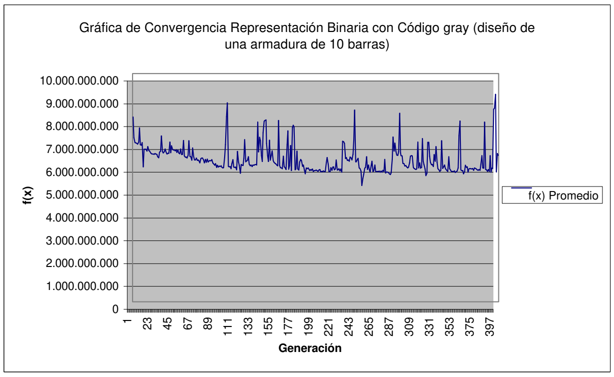

Figura 6.11: Resultados de 30 corridas y gráfica de convergencia para el problema 4 (representación binaria).

Diseño de una armadura de 10 barras
Representacion Real SBX y Parameter Based Mutation

Prueba	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	f(X)	Factible
1	771,251601	0,100000	308,730432	397,792533	0,000062	0,100000	112,497930	771,654841	392,874333	0,100000	5352,856480	**SI**
2	759,332580	20,250932	541,249331	426,163379	0,001169	14,097188	631,280721	585,213750	156,522530	22,775574	6261,828577	**SI**
3	723,508721	32,595710	586,478474	176,883898	0,100000	33,199407	295,509528	463,159802	214,367215	95,610024	5957,391992	**SI**
4	708,503715	41,942212	549,811337	133,016074	0,100000	41,516816	579,866143	260,661155	263,027021	94,346903	6062,208910	**SI**
5	728,140570	0,100000	744,053198	362,359459	0,100000	0,100000	62,757299	380,883990	502,099358	0,100000	5226,682535	**SI**
6	653,037652	71,870434	988,194949	96,446685	0,100000	80,462096	568,184257	221,385446	100,492374	157,840475	6165,768366	**SI**
7	820,278757	0,100000	490,679605	405,962384	0,100000	0,100000	75,260441	389,929548	524,158376	0,100000	5216,708727	**SI**
8	651,673019	0,100000	804,923210	458,078841	25,132153	0,100000	415,885390	154,457182	841,012296	0,100000	6077,649195	**SI**
9	696,282677	342,860037	709,057401	205,136849	0,000081	96,505839	299,922242	220,951461	171,293966	159,683347	6392,889754	**SI**
10	753,994445	40,130555	561,246062	136,325197	0,100000	38,238975	275,679336	523,795978	217,083783	112,999815	6025,601453	**Si**
11	486,616830	0,100000	560,029570	455,151721	0,100000	0,100000	80,414834	679,408988	506,264113	0,100000	5393,476077	**SI**
12	613,680459	9,745362	967,590092	262,551645	0,100000	9,268283	419,929463	198,194315	463,858108	18,821586	5905,734043	**SI**
13	925,110239	0,100000	606,936848	281,154598	0,100000	0,100000	66,028795	395,013598	465,989901	0,100000	5160,336241	**SI**
14	513,112262	25,689785	881,841324	161,009642	0,100000	25,100477	403,120321	273,443106	344,950446	51,483535	5890,490482	**SI**
15	890,217384	58,325030	410,857218	349,480176	0,001939	17,359768	440,486191	156,363363	561,581701	100,025185	6324,153929	**SI**
16	851,287722	48,817213	557,348845	160,387050	0,100000	50,655851	600,516232	222,913139	192,184242	109,858164	6122,469163	**SI**
&17	745,588845	32,708114	786,866035	198,636233	0,100000	33,535124	379,868446	161,967300	333,989002	76,853701	5942,929690	**SI**
18	748,727121	14,129719	466,561650	263,685019	0,100000	14,927180	316,876257	441,926518	304,330744	35,241127	5799,981337	**SI**
19	870,685029	1,376805	418,019993	322,386856	0,100000	1,232449	209,089576	358,156221	528,166978	3,632059	5505,038642	**SI**
20	787,685754	33,478370	795,612051	272,986019	0,100000	32,937346	342,615069	194,103412	235,830741	78,193510	5930,608412	**SI**
21	319,114691	54,614535	819,171777	440,225908	0,001986	68,984109	358,591961	271,307388	359,197014	159,475923	6405,962359	**SI**
22	533,840337	0,100000	438,280469	211,107666	0,100000	0,100000	90,172754	816,812715	646,222113	0,100000	5391,488394	**SI**
23	794,787333	40,675097	987,912738	133,776850	0,100000	47,698028	450,339424	494,872185	90,024482	92,545981	6238,229446	**SI**
24	794,787333	40,675097	987,912738	133,776850	0,100000	47,698028	450,339424	494,872185	90,024482	92,545981	6238,229446	**SI**
25	399,358259	0,100000	507,244373	402,271080	0,100000	0,100000	75,640312	708,926397	788,221460	0,000404	5515,138297	**SI**
26	672,338666	72,361797	631,760443	37,165496	0,100000	73,349233	662,844138	317,527408	236,044191	156,507066	6320,808189	**SI**
27	856,013342	0,100000	601,548206	172,969615	0,100000	0,000081	65,334218	539,767385	474,838781	0,100000	5152,636136	**SI**
28	655,387640	20,584626	535,608086	226,634217	0,100000	20,176663	417,090337	335,415312	329,097149	46,884937	5877,525240	**SI**
29	703,886797	11,636538	781,469432	300,025214	0,100000	11,305964	273,478092	297,672866	309,321478	24,691443	5709,002641	**SI**
30	564,039157	15,664086	782,480524	148,908855	0,001146	77,154744	510,527478	254,610780	274,671091	74,084710	6005,917148	**SI**

MEDIA	**5869,547752**	
DESV. EST.	**391,521216**	
MEJOR	**5152,636136**	**SI**
PEOR	**6405,962359**	**SI**
MEDIANA	**5942,929690**	**SI**
VARIANZA	**153288,862356**	

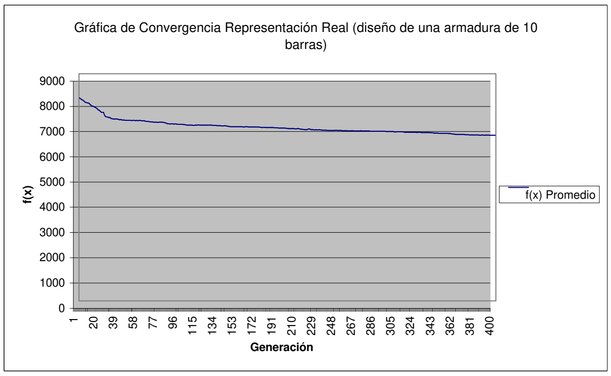

Figura 6.12: Resultados de 30 corridas y gráfica de convergencia para el problema 4 (representación real).

Problema 5

Representacion Binaria con Código Gray Cruza de 2 puntos y Mutación Uniforme

Prueba	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11
1	3,054536	3,203219	3,121261	3,030338	3,143456	3,031976	2,924928	1,041008	2,991275	0,448434	0,519561
2	6,226722	3,155379	3,114982	2,997501	3,165633	3,095120	0,965073	2,827206	0,533201	0,517695	0,625109
3	3,170140	3,447412	3,238984	3,033157	3,132490	2,967858	3,089732	0,409868	0,345963	2,924532	1,123713
4	3,182330	3,122111	3,074973	3,019912	3,008669	0,510412	2,971562	2,931525	2,976774	2,988866	0,276489
5	6,094214	3,172168	3,131023	2,976169	3,101546	2,887213	2,950790	0,373939	2,986885	0,213119	1,227341
6	3,211492	3,075988	3,010481	3,059000	2,934995	2,892515	2,941487	2,926300	2,838597	2,736287	0,548511
7	3,081342	3,125012	3,076513	3,046423	3,066759	3,003420	3,006623	2,926552	0,353516	0,678855	0,655082
8	3,184093	3,085355	3,040821	3,017850	3,055657	3,000426	2,868660	2,912807	2,796207	0,535047	0,312689
&9	6,220391	3,075506	3,090293	3,090820	2,987818	2,982820	2,937957	2,959903	0,314425	0,485197	2,779479
10	3,191471	3,136880	3,122730	3,045546	3,039032	3,017857	3,008854	3,039241	0,422139	0,900442	0,413507
11	6,205093	3,042838	2,998497	3,065352	3,079414	3,050156	2,907135	2,898987	1,157532	0,965583	0,261480
12	6,236288	3,137403	3,074824	2,989465	3,130014	3,089144	2,931726	0,830382	0,438243	0,503410	0,390614
13	3,145806	3,171700	3,107050	3,112863	2,964484	3,000729	3,126856	2,930394	2,855858	1,506257	0,583708
14	3,100199	3,117020	3,107506	3,070352	2,979464	2,952134	0,468208	2,925035	0,378568	2,811662	3,101250
15	3,088217	3,089385	3,031036	3,083603	3,037552	2,979810	0,363585	2,940766	2,910585	0,426216	1,174101
16	3,152762	3,079396	3,138426	3,083627	2,870684	3,101004	0,420611	2,879222	3,166533	0,411483	2,855840
17	6,209150	3,132020	2,971437	3,041189	3,041505	2,987819	2,960963	1,309034	2,979825	0,335510	2,938359
18	6,167961	3,158771	3,094350	3,115459	3,065029	2,915037	2,976940	0,419678	0,221016	2,895904	0,280457
19	3,146359	3,191070	3,074750	3,070736	2,888665	2,920660	2,952039	2,918059	1,206370	0,668163	0,368660
20	6,161591	3,137261	2,965211	3,085959	2,986760	2,947657	2,935352	3,135789	1,930208	2,817281	0,258586
21	6,212432	0,293245	3,293185	3,096481	3,120563	3,037741	3,037075	2,912810	0,534554	2,907638	0,300154
22	3,046389	3,188885	3,186826	3,007496	3,009996	3,090412	3,067101	0,303590	2,939763	0,353630	0,546265
23	3,147555	3,103994	3,210940	3,125540	2,995513	3,067569	2,986755	0,422030	2,925987	0,258313	2,890856
24	3,088605	3,034861	3,112081	3,079128	3,090021	2,949984	2,929546	2,905113	0,485143	1,372498	0,421162
25	3,155315	3,293755	0,412161	3,010997	3,045526	3,021344	3,025038	3,017427	2,897402	2,943727	0,291991
26	6,210464	3,108944	3,090199	2,974310	3,016223	2,932582	2,950842	0,401537	0,434723	0,752633	1,302024
27	6,296906	3,093147	3,125575	3,105163	3,018480	2,972812	2,937930	2,952076	0,234257	0,421622	0,646160
28	6,220015	3,045681	3,084943	3,045705	2,984170	2,990371	0,439191	2,929475	0,293998	0,321713	2,881748
29	3,180487	3,112546	3,087863	3,163566	3,127920	2,932208	2,917065	2,925140	0,514120	0,436243	0,302571
30	3,165873	2,935171	3,177132	3,022239	3,061333	3,040691	2,920207	2,989161	2,959076	0,419604	0,500553

Prueba	X12	X13	X14	X15	X16	X17	X18	X19	X20	g1(X)	g2(X)	f(X)	Factible
1	0,646501	0,408953	0,369074	0,380544	0,362630	0,502278	0,345941	0,377238	0,459076	-0,008433	-119,637774	0,777734	SI
2	0,626536	0,452188	0,438243	0,451664	0,372057	0,412921	0,735580	0,343287	0,412872	-0,010111	-118,531031	0,757102	SI
3	0,440128	0,387997	0,343971	0,672780	0,409830	0,429202	0,674149	0,324242	0,343649	-0,000450	-119,090204	0,748760	SI
4	0,618380	0,419990	0,260475	0,611033	0,307203	0,287819	0,362660	0,373224	0,498753	0,000871	-118,196842	0,746686	SI
5	2,981860	0,454764	0,292661	0,260632	0,279518	0,258501	0,249879	0,720040	0,401212	-0,000270	-114,986526	0,720113	SI
6	0,400854	0,328189	0,363272	0,401456	0,402075	0,167754	0,331764	0,178328	0,364844	-0,041123	-116,885811	0,750558	SI
7	0,444329	0,316078	0,413141	0,352667	0,431675	0,376145	0,520993	0,384815	0,978090	0,000815	-119,761971	0,775139	SI
8	0,608175	0,348824	0,399617	0,322189	0,398307	0,514379	0,347158	0,376830	0,317678	-0,001419	-118,557231	0,783059	SI
&9	0,625756	0,452740	0,332084	0,441787	0,447439	0,292687	0,165662	0,346195	0,430225	-0,056496	-115,540815	0,750560	SI
10	0,432411	2,630560	0,372748	0,327977	0,430578	0,256016	0,369310	0,292369	0,369572	-0,017765	-118,180759	0,752452	SI
11	0,410280	0,343436	0,283387	0,657016	0,338241	0,426581	0,381855	0,337622	0,389779	-0,013714	-116,799736	0,759659	SI
12	0,389156	1,463693	0,444656	0,228693	0,368005	0,469987	0,414112	0,640976	0,791385	-0,017749	-118,037823	0,739681	SI
13	0,420483	0,333964	0,375676	0,327333	0,365275	0,296894	0,299827	0,243630	0,278646	-0,002836	-117,552566	0,772845	SI
14	0,369302	0,432490	0,464434	0,247245	0,336031	0,472240	0,409424	0,403014	0,425795	0,000344	-118,428629	0,745678	SI
15	2,815105	0,376681	0,368938	0,429578	0,298408	0,312517	0,267510	0,433172	0,330486	-0,004438	-118,242750	0,744149	SI
16	0,395568	0,369875	0,340500	0,290480	0,293398	0,371081	0,389783	0,536413	0,605487	-0,001808	-118,247825	0,745780	SI
17	0,147728	0,358908	0,389753	0,335409	0,296991	0,337788	0,248033	0,497304	0,499126	-0,028459	-114,982149	0,737968	SI
18	2,933613	0,416047	1,542249	0,224068	0,343110	0,286764	0,358406	0,349180	0,385885	-0,004238	-114,850077	0,725365	SI
19	0,348879	0,326668	0,417615	0,459632	0,534002	0,492240	0,447146	0,481163	0,356198	-0,152193	-119,730927	0,777676	SI
20	0,489174	0,267956	0,308772	0,265560	0,211551	0,420886	0,375422	0,381159	0,274868	-0,008701	-114,642998	0,744862	SI
21	0,284825	2,732685	0,491259	0,441312	0,615987	0,356027	0,323957	0,333710	0,259097	-0,003355	-115,415263	0,689747	SI
22	1,057062	1,454085	0,365319	0,298122	0,372340	0,361085	0,440397	0,424349	0,392492	-0,006059	-119,094395	0,754375	SI
23	0,342510	1,533700	0,280599	0,292105	0,283464	0,359430	0,249890	0,610279	0,453703	-0,003190	-117,459269	0,742010	SI
24	0,479062	0,406244	0,502223	0,489684	0,378411	0,325555	0,345997	0,519377	0,364493	-0,004704	-119,720812	0,787933	SI
25	2,862904	0,341265	0,303510	0,322275	1,254960	0,204374	0,312405	0,220875	0,164555	-0,000835	-115,898195	0,692784	SI
26	0,496914	0,667995	0,512208	0,379894	0,525172	0,298791	0,403882	0,513003	0,462433	-0,019421	-118,565229	0,761694	SI
27	0,556437	0,409926	0,341733	0,554454	0,519852	0,737617	0,335882	0,378452	0,380564	-0,002718	-116,980954	0,771496	SI
28	2,643142	0,450833	0,411358	0,560101	0,295101	0,783634	0,236084	0,301289	0,308645	-0,026403	-115,772803	0,716140	SI
29	0,807760	0,381873	0,492991	0,398045	0,531611	0,362709	0,540010	0,578389	0,400327	0,000501	-119,806554	0,785204	SI
30	0,358160	0,616736	0,315881	0,354325	0,341781	0,378029	0,424530	0,362541	0,341427	-0,006347	-118,315549	0,781825	SI

MEDIA	0,751301	
DESV. EST.	0,025028229	
MEJOR	0,787933	SI
PEOR	0,689747	SI
MEDIANA	0,750559	SI
VARIANZA	0,000626412	

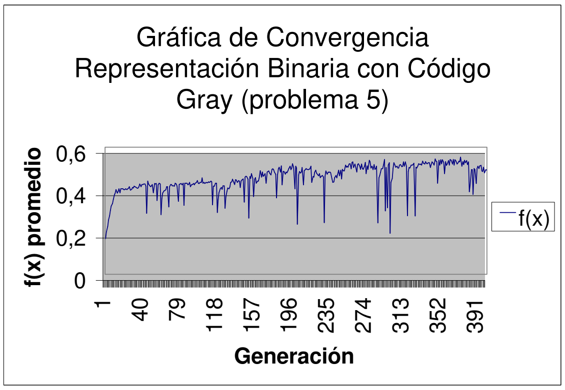

Figura 6.13: Resultados de 30 corridas y gráfica de convergencia para el problema 5 (representación binaria).

Problema 5

Representacion Real SBX y Parameter Based Mutation

Prueba	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10
1	9,417835	6,273650	3,134385	6,264076	3,129756	3,127692	0,007342	4,682103	3,120398	6,235304
2	6,256161	3,138469	0,221302	3,107984	3,067729	0,405324	3,067875	0,372948	1,585459	0,380775
3	6,279371	6,275562	6,271756	3,133978	6,264153	6,260354	6,256557	3,126398	3,124506	6,245167
4	0,412828	3,165422	0,556812	0,296096	3,133799	3,137474	3,132322	0,332900	0,406103	3,119416
&5	3,137714	3,134073	4,693694	0,012383	6,244856	0,012820	0,013859	0,015192	3,107326	6,206426
6	1,569579	9,410832	9,403878	6,264689	3,130195	6,255386	0,001218	6,246160	3,120700	3,118550
7	3,138250	3,134897	3,131494	3,128172	3,124799	9,364054	7,791094	0,010479	0,010797	6,216041
8	0,000280	3,134690	9,393704	0,000320	3,124383	3,120969	6,234939	3,114118	0,000256	6,214207
9	3,137176	6,265093	3,127978	3,123680	3,119131	0,004329	7,767373	3,105685	3,101304	0,003673
10	3,139394	6,274411	9,405042	6,265658	3,130652	9,385311	1,562816	3,124117	9,365512	6,239431
11	6,278932	3,137385	3,135188	6,266100	9,392746	9,386348	9,379923	6,249104	6,244772	9,360637
12	3,132651	3,123789	3,114710	6,211560	0,032324	3,088444	3,079585	3,070739	3,061881	6,098316
13	0,025903	3,128791	1,560472	3,117498	3,110132	3,103810	3,097447	6,180514	0,023270	6,155089
14	6,267116	4,683874	0,006985	3,109495	6,202363	0,006896	3,085727	3,077705	3,069761	0,007201
15	0,012382	0,012462	0,010988	3,128851	3,125210	6,244842	3,119365	4,672096	6,225744	3,110076
16	3,138011	0,003106	6,261732	6,254577	3,123728	3,120206	0,003044	1,555553	3,109602	9,316334
17	3,138874	3,136419	0,021259	6,262147	6,256873	6,251471	0,019574	0,017741	7,791623	6,230572
18	3,136986	6,259221	3,126553	3,119869	0,140260	0,153192	0,187423	6,187773	0,194466	0,148016
19	6,271488	1,574596	6,246755	6,234705	0,101759	6,206815	0,116165	0,129082	0,116731	3,078903
20	0,209516	6,234143	6,209474	3,095215	0,166965	3,067575	3,065205	0,173011	0,200263	3,030472
21	0,201397	9,397530	0,227634	6,226076	1,682999	0,221814	3,102403	0,209116	1,364416	3,078155
22	3,137988	9,403120	3,130757	6,254364	3,123609	0,002907	0,002581	3,112892	9,326603	1,551623
23	0,151103	0,434967	6,158759	4,438400	3,118045	6,141692	3,103019	6,165442	3,139525	0,413912
24	4,708202	0,019183	3,132471	6,257593	9,376902	0,027502	3,119171	6,231549	6,225302	3,110470
25	9,417890	9,411013	7,836089	0,000715	9,390381	6,255811	9,376611	3,123306	9,362756	3,118765
26	6,272346	6,261543	3,125575	0,041873	3,114306	6,217686	1,548015	3,099817	0,033877	6,173246
27	1,568475	0,030042	6,260663	0,025626	9,371000	3,124282	6,236062	9,341377	9,328735	0,029410
28	3,137826	3,134097	1,564522	0,042090	3,122969	0,041685	0,044437	3,111655	3,107929	3,104318
29	3,137926	6,263592	3,119058	0,212266	0,210461	6,195712	3,095603	0,191063	6,145956	3,073094
30	1,567463	6,259047	3,123428	3,117916	6,223037	3,105915	0,050185	3,093886	0,049754	0,049138

Prueba	X11	X12	X13	X14	X15	X16	X17	X18	X19	X20	g1(X)	g2(X)	f(X)	Factible
1	6,230601	0,006706	0,006713	0,005610	0,006108	9,308197	6,201345	6,196753	3,096393	9,277536	-0,000094	-64,271497	0,240300	SI
2	0,469965	3,276084	0,388189	0,292191	1,289642	0,367375	2,899132	0,242646	0,392249	2,834611	-0,000626	-115,943890	0,577987	SI
3	3,120730	3,118845	1,557904	3,115077	3,113195	3,111315	9,326935	6,214717	9,315114	9,309156	-2666176685804 5,500000	-45,459209	0,214291	SI
4	0,350428	0,666946	0,320124	0,538820	3,105141	3,083496	0,309985	0,624968	8,918444	0,402384	-0,000907	-113,986093	0,336140	SI
&5	6,198690	3,096214	6,183292	9,258509	0,013513	3,080726	3,077443	3,073538	3,069558	3,065801	-0,006422	-83,304374	0,303764	SI
6	9,348003	6,227600	0,001254	9,326691	9,319512	0,001355	0,001164	6,199744	3,097930	3,095700	-0,000641	-54,859860	0,236462	SI
7	3,104872	0,010275	6,195583	6,188764	0,010068	0,010967	3,084977	6,160985	3,078314	6,146973	-0,000599	-76,958146	0,284089	SI
8	3,103901	3,100503	3,097104	3,093718	3,090320	6,172156	4,618987	6,157844	6,150626	3,073368	-0,001702	-74,003607	0,286949	SI
9	3,092379	3,087950	6,165019	3,079119	3,074650	6,136583	3,065851	0,004424	0,003732	6,096939	-0,002042	-83,437933	0,328023	SI
10	3,117597	6,230682	6,226305	0,000000	9,325373	0,000002	9,311739	3,102454	3,100298	3,098144	-0,001302	-48,595062	0,229966	SI
11	0,000915	3,116093	6,227794	0,000845	6,219254	0,000926	1,552079	6,206363	9,301361	0,000839	-0,009777	-48,542397	0,226936	SI
12	0,032110	3,035277	3,026213	0,028654	0,024004	0,028436	1,484193	2,981299	2,971942	2,962560	-0,000043	-99,411314	0,460647	SI
13	3,072480	3,066329	0,024920	6,100152	3,047529	0,022841	3,033951	0,026073	3,022453	3,016042	-0,000187	-96,064305	0,389796	SI
14	3,053837	3,045915	3,037907	3,029907	0,006327	3,013810	3,005666	2,997110	2,989391	2,981115	-0,000342	-93,321892	0,438673	SI
15	0,012095	6,207281	3,100666	3,097558	9,278762	6,180929	3,087721	6,168283	3,081881	0,010584	-0,000635	-80,112224	0,276247	SI
16	3,102546	6,197227	3,095508	6,182717	6,175388	6,168012	0,003113	3,077904	0,003569	3,070912	-0,001035	-77,037212	0,293418	SI
17	9,337374	0,020506	6,214707	0,020405	6,204157	3,099796	3,096914	9,277667	3,092793	0,020979	-0,002307	-70,488148	0,251535	SI
18	6,144591	9,194332	3,068605	3,059322	0,156571	3,050011	1,534037	0,191327	0,154494	0,475512	-0,000227	-100,317438	0,371695	SI
19	3,074257	0,109272	6,115285	0,109987	6,085086	0,129730	3,038270	3,031585	3,023246	0,115346	-0,000003	-95,090935	0,387925	SI
20	0,142684	2,992194	2,988888	0,184148	2,957648	2,942056	1,431827	0,120920	2,896674	0,176831	-0,000396	-107,714291	0,536843	SI
21	3,070043	0,225450	3,052875	6,054709	3,044091	0,204833	0,234855	3,017854	0,264099	0,184246	-0,000638	-104,935405	0,428160	SI
22	0,003088	6,196568	6,189221	6,181849	3,088018	3,084492	6,159556	3,077409	0,003047	3,070296	-0,004167	-73,900013	0,294608	SI
23	0,226250	0,202233	0,278197	3,026530	3,036501	0,238386	0,139324	0,385219	2,959479	0,218688	-0,078526	-106,024329	0,465978	SI
24	6,213600	6,206280	0,019850	6,194519	3,095494	9,265751	0,021271	0,018748	3,083091	0,026307	-0,005869	-73,646746	0,278473	SI
25	0,000796	0,000831	9,334825	3,109700	6,214479	9,313531	3,102917	3,100662	0,000918	3,096144	-0,008896	-45,431857	0,235213	SI
26	3,083094	3,076461	0,043822	0,032403	6,115231	0,036577	3,049955	6,078020	0,044834	3,033787	-0,008709	-89,517531	0,361610	SI
27	3,104081	6,198344	3,096041	0,025157	3,093332	0,027944	6,162135	0,028820	3,082296	3,073878	-0,021019	-76,792300	0,290517	SI
28	0,045061	0,040904	0,043351	6,177409	3,085974	6,161907	6,154188	3,074643	6,138177	6,130143	-0,001577	-92,536717	0,300299	SI
29	6,116516	6,095071	0,215057	1,379089	0,180217	0,174814	1,229243	0,202105	0,203483	0,224867	-0,000564	-102,334807	0,407349	SI
30	3,076928	0,047036	6,123333	0,055967	3,052386	0,042023	3,040560	3,034557	3,028652	6,023398	-0,000681	-95,835389	0,378913	SI

MEDIA	0,337094	
DESVIACIÓN ESTÁNDAR	0,094511735	
MEJOR	0,577987	SI
PEOR	0,214291	SI
MEDIANA	0,302032	SI
VARIANZA	0,008932468	

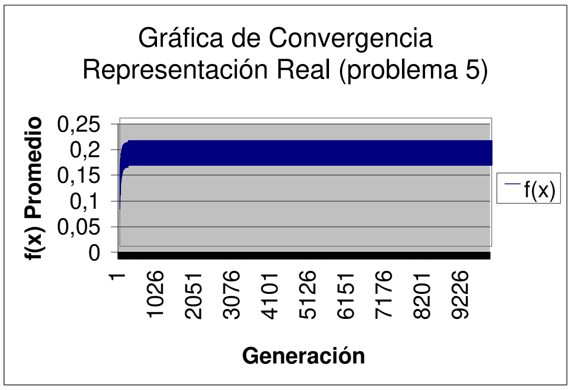

Figura 6.14: Resultados de 30 corridas y gráfica de convergencia para el problema 5 (representación real).

6.4.6 Problema 6

El mínimo global se encuentra en: [119]: $\mathbf{f(x) = -30665.539}$, con $x = (78, 33, 29.995256025682, 45, 36.775812905788)$, con g_1 y g_6 activas.

Representación binaria

El mejor valor encontrado por el algoritmo es: $\mathbf{f(x) = -30659.996878}$, con $x = (78.003942, 33.019521, 30.020488, 44.990450, 36.732249)$, y con $g_1 = -0.003615$, $g_2 = -91.996385$, $g_3 = -11.156384$, $g_4 = -8.843616$, $g_5 = -4.997549$ y $g_6 = -0.002451$.

Representación real

El mejor valor encontrado por el algoritmo es: $\mathbf{f(x) = -30557.850481}$, con $x = (77.999999, 33.000000, 30.001641, 40.967347, 38.396399)$, y con $g_1 = -0.000636$, $g_2 = -91.999364$, $g_3 = -10.777266$, $g_4 = -9.222734$, $g_5 = -4.999983$ y $g_6 = -0.000017$.

6.4.7 Problema 7

La solución óptima se encuentra en: [119]: $\mathbf{f(x) = 24.3062091}$, con $x = (2.171996, 2.363683, 8.773926, 5.095984, 0.9906548, 1.430574, 1.321644, 9.828726, 8.280092, 8.375927)$, con g_1, g_2, g_3, g_4, g_5, y g_6 activas.

Representación binaria

El mejor valor encontrado por el algoritmo es: $\mathbf{f(x) = 25.473916}$, con $x = (2.088123, 2.402701, 8.719868, 4.964167, 0.921841, 1.303988, 1.282828, 9.830786, 8.049844, 8.203815)$, y con $g_1 = -0.005412$, $g_2 = -0.486879$, $g_3 = -0.057991$, $g_4 = -1.226610$, $g_5 = -1.507760$, $g_6 = -0.267830$, $g_7 = -6.176742$ y $g_8 = -49.245055$.

Representación real

El mejor valor encontrado por el algoritmo es: $\mathbf{f(x) = 24.381843}$, con $x = (2.136325, 2.453076, 8.785347, 5.100696, 1.009871, 1.438481, 1.251897, 9.771741, 8.206078, 8.422531)$, y con $g_1 = 0.000655$, $g_2 = -0.000127$, $g_3 = 0.000874$, $g_4 = -0.087974$, $g_5 = 0.000805$, $g_6 = 0.000611$, $g_7 = -6.401676$ y $g_8 = -50.138621$.

Problema 6

Representacion Binaria con Código Gray Cruza de 2 puntos y Mutación Uniforme

Prueba	X1	X2	X3	X4	X5	g1(X)	g2(X)	g3(X)	g4(X)	g5(X)	g6(X)	f(X)	Factible
1	78,124795	33,644262	30,361258	44,983370	35,838227	-0,008844	-91,991156	-11,004197	-8,995803	-4,999526	-0,000474	-30599,904403	SI
2	78,098401	33,089732	30,152296	44,987064	36,475021	-0,028423	-91,971577	-11,155625	-8,844375	-4,983658	-0,016342	-30627,868350	SI
3	78,008375	33,007929	30,024029	44,937721	36,752075	-0,006365	-91,993635	-11,156561	-8,843439	-4,996342	-0,003658	-30657,263866	SI
4	78,022533	33,146509	30,082706	44,995358	36,559001	-0,002527	-91,997473	-11,124397	-8,875603	-4,998895	-0,001105	-30649,993451	SI
5	78,057145	33,079527	30,263147	44,950593	36,117287	-0,085810	-91,914190	-11,234541	-8,765459	-4,998848	-0,001152	-30618,117696	SI
6	78,004739	33,107989	30,058059	44,995982	36,618495	-0,001762	-91,998238	-11,134394	-8,865606	-4,999888	-0,000112	-30655,264164	SI
7	78,007600	33,617852	30,375291	44,992413	35,828796	-0,019365	-91,980635	-11,029332	-8,970668	-4,999870	-0,000130	-30603,832841	SI
8	78,031549	33,387547	30,357711	44,927624	35,901928	-0,058396	-91,941604	-11,124515	-8,875485	-4,998463	-0,001537	-30603,174319	SI
9	78,003682	33,168780	30,222216	44,776977	36,300622	-0,051851	-91,948149	-11,158027	-8,841973	-4,999305	-0,000695	-30623,038935	SI
10	78,003942	33,019521	30,020488	44,990450	36,732249	-0,003615	-91,996385	-11,156384	-8,843616	-4,997549	-0,002451	-30659,996878	SI
11	78,006269	33,120273	30,165949	44,962483	36,362837	-0,040647	-91,959353	-11,174378	-8,825622	-4,999622	-0,000378	-30637,013489	SI
12	78,284321	33,065294	30,080940	44,970717	36,510467	-0,019003	-91,980997	-11,150301	-8,849699	-4,997923	-0,002077	-30635,976734	SI
13	78,011896	33,196927	30,114817	44,982393	36,482513	-0,004926	-91,995074	-11,114418	-8,885582	-4,999453	-0,000547	-30645,344887	SI
14	78,016902	33,177328	30,121554	44,951061	36,474609	-0,011889	-91,988111	-11,124585	-8,875415	-4,999790	-0,000210	-30643,346375	SI
15	78,003128	33,220227	30,133188	44,991483	36,447100	-0,005709	-91,994291	-11,109759	-8,890241	-4,997753	-0,002247	-30642,317482	SI
16	78,000336	33,012260	30,006465	44,573954	36,924524	-0,001073	-91,998927	-11,116908	-8,883092	-4,999403	-0,000597	-30652,218717	SI
17	78,095828	33,094347	30,205771	44,998731	36,262621	-0,057088	-91,942912	-11,196690	-8,803310	-4,994249	-0,005751	-30624,611290	SI
18	78,000441	33,169832	30,133028	44,977770	36,438042	-0,017992	-91,982008	-11,137064	-8,862936	-4,999983	-0,000017	-30643,141452	SI
19	78,008553	33,011516	30,036680	44,984753	36,675544	-0,013634	-91,986366	-11,171125	-8,828875	-4,999939	-0,000061	-30658,169891	SI
20	78,015103	33,216707	30,156220	44,917893	36,400982	-0,017191	-91,982809	-11,118202	-8,881798	-4,999895	-0,000105	-30637,072868	SI
21	78,003204	33,174439	30,112767	44,439564	36,735426	-0,005276	-91,994724	-11,066821	-8,933179	-4,995891	-0,004109	-30630,108965	SI
22	78,020849	33,002731	30,041165	44,981146	36,666111	-0,016803	-91,983197	-11,175892	-8,824108	-4,999418	-0,000582	-30656,505770	SI
& 23	78,006269	33,120273	30,165949	44,962483	36,362837	-0,040647	-91,959353	-11,174378	-8,825622	-4,999622	-0,000378	-30637,013489	SI
24	78,029428	33,200190	30,122328	44,856547	36,506868	-0,007529	-91,992471	-11,104310	-8,895690	-4,999906	-0,000094	-30640,144002	SI
25	78,009321	33,813571	30,555177	44,999805	35,404570	-0,046272	-91,953728	-11,011806	-8,988194	-4,996967	-0,003033	-30572,648043	SI
26	78,006916	33,373418	30,306168	44,951423	36,046817	-0,038920	-91,961080	-11,106233	-8,893767	-4,995542	-0,004458	-30612,139657	SI
27	78,141343	33,653322	30,621823	44,975242	35,258800	-0,099265	-91,900735	-11,102498	-8,897502	-4,990979	-0,009021	-30551,492556	SI
28	78,000441	33,169832	30,133028	44,977770	36,438042	-0,017992	-91,982008	-11,137064	-8,862936	-4,999983	-0,000017	-30643,141452	SI
29	78,094279	33,270071	30,140996	44,774755	36,479728	-0,000035	-91,999965	-11,067305	-8,932695	-4,999375	-0,000625	-30631,490851	SI
30	78,046953	33,217467	30,205744	44,971703	36,265201	-0,034083	-91,965917	-11,140303	-8,859697	-4,997303	-0,002697	-30627,755705	SI

MEDIA	-30630,670286	
DESV. EST.	25,02923148	
MEJOR	-30659,996878	SI
PEOR	-30551,492556	SI
MEDIANA	-30637,013489	SI
VARIANZA	626,4624285	

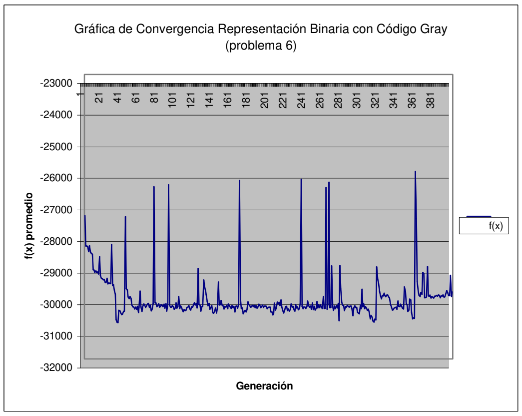

Figura 6.15: Resultados de 30 corridas y gráfica de convergencia para el problema 6 (representación binaria).

Problema 6

Representacion Real SBX y Parameter Based Mutation

Prueba	X1	X2	X3	X4	X5	g1(X)	g2(X)	g3(X)	g4(X)	g5(X)	g6(X)	f(X)	Factible
1	78,000000	37,275514	33,243778	35,908880	33,055486	-0,329238	-91,670762	-9,580065	-10,419935	-4,999681	-0,000319	-29807,361144	SI
2	78,690951	35,372465	31,224256	45,000000	33,605996	-0,003367	-91,996633	-10,545245	-9,454755	-4,999998	-0,000002	-30423,868138	SI
3	78,000000	34,188566	30,724783	36,844001	38,285568	-0,017822	-91,982178	-10,105325	-9,894675	-4,999899	-0,000101	-30329,790765	SI
4	78,000000	33,166170	30,143942	40,662163	38,166289	-0,019411	-91,980589	-10,728696	-9,271304	-4,999413	-0,000587	-30526,993143	SI
5	78,000000	37,910901	33,434714	39,541009	31,188586	-0,311095	-91,688905	-9,758802	-10,241198	-4,999995	-0,000005	-29860,840033	SI
6	77,999999	33,000000	30,001641	40,967347	38,396399	-0,000636	-91,999364	-10,777266	-9,222734	-4,999983	-0,000017	-30557,850481	SI
7	78,000000	34,210762	30,863748	30,389388	40,572035	-0,050862	-91,949138	-9,517555	-10,482445	-4,999849	-0,000151	-30134,894306	SI
8	78,000000	35,832549	31,707332	28,941102	39,200170	-0,006532	-91,993468	-8,904805	-11,095195	-4,999609	-0,000391	-29941,508942	SI
9	85,671020	36,121207	31,437912	45,000000	31,248455	-0,000182	-91,999818	-10,012165	-9,987835	-4,999997	-0,000003	-30064,584180	SI
10	91,016787	34,432625	31,069984	45,000000	30,679653	-0,196588	-91,803412	-10,460275	-9,539725	-4,999922	-0,000078	-29892,110195	SI
11	78,000000	33,142686	30,106242	36,816445	39,825882	-0,006621	-91,993379	-10,353263	-9,646737	-4,998800	-0,001200	-30430,984707	SI
12	74,382480	39,292160	33,332423	44,999968	30,148962	-0,050260	-91,949740	-9,860844	-10,139156	-4,999710	-0,000290	-30191,252326	SI
13	78,000000	34,581554	33,319995	35,256495	33,161808	-0,859887	-91,140113	-10,807297	-9,192703	-4,999977	-0,000023	-29773,248929	SI
14	78,000000	38,717264	33,032556	38,163761	32,580335	-0,002717	-91,997283	-9,065056	-10,934944	-4,999318	-0,000682	-29913,338154	SI
15	78,000000	36,260885	33,444979	36,248251	32,512090	-0,589979	-91,410021	-10,167577	-9,832423	-4,998728	-0,001272	-29770,891039	SI
16	78,000000	41,619273	34,144781	44,999997	27,559807	-0,021149	-91,978851	-9,039908	-10,960092	-4,999709	-0,000291	-29840,275622	SI
17	78,000000	33,000000	30,011007	35,615621	40,544635	-0,001928	-91,998072	-10,270461	-9,729539	-4,999989	-0,000011	-30414,808788	SI
18	84,779236	34,428254	32,576246	45,000000	28,957911	-0,688373	-91,311627	-11,319327	-8,680673	-4,999942	-0,000058	-29892,988487	SI
19	78,000000	35,400382	32,452633	34,941026	35,114752	-0,404153	-91,595847	-10,053716	-9,946284	-4,999984	-0,000016	-29951,607053	SI
20	78,000000	37,548570	32,464239	40,171376	32,966925	-0,025463	-91,974537	-9,587341	-10,412659	-4,999997	-0,000003	-30087,573669	SI
21	78,000000	34,277714	30,987978	42,037137	35,553597	-0,112740	-91,887260	-10,692578	-9,307422	-4,999237	-0,000763	-30420,845994	SI
22	78,000000	36,268464	32,477930	36,781260	34,313304	-0,250780	-91,749220	-9,837215	-10,162785	-4,999985	-0,000015	-29995,047949	SI
23	78,000000	34,509362	31,907225	42,615213	33,198572	-0,406128	-91,593872	-11,033199	-8,966801	-4,999993	-0,000007	-30264,584471	SI
24	78,000000	34,370668	30,745262	41,572734	36,316681	-0,000216	-91,999784	-10,492926	-9,507074	-4,999965	-0,000035	-30451,385432	SI
25	78,000000	35,783713	32,916790	42,237783	31,166569	-0,523852	-91,476148	-10,809542	-9,190458	-4,999722	-0,000278	-30046,398319	SI
& 26	84,729889	36,359700	31,582248	45,000000	31,168928	-0,005177	-91,994823	-10,001096	-9,998904	-4,999981	-0,000019	-30081,154469	SI
27	78,000000	39,411946	34,390769	35,103291	31,098130	-0,340834	-91,659166	-8,958189	-11,041811	-4,999957	-0,000043	-29519,306976	SI
28	78,000000	34,084416	30,978622	35,729750	38,139045	-0,134735	-91,865265	-10,159549	-9,840451	-4,998729	-0,001271	-30255,423207	SI
29	78,000000	37,484467	32,440049	36,951013	34,326611	-0,000497	-91,999503	-9,257333	-10,742667	-4,999924	-0,000076	-30007,356294	SI
30	85,120566	34,601319	30,722308	44,909018	33,117894	-0,000143	-91,999857	-10,433692	-9,566308	-5,000000	0,000000	-30204,834201	SI

MEDIA	-30101,770247	
DESV. EST.	263,3513097	
MEJOR	-30557,850481	SI
PEOR	-29519,306976	SI
MEDIANA	-30072,869325	SI
VARIANZA	69353,91231	

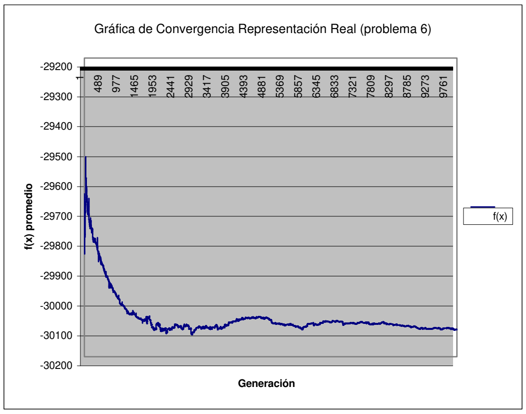

Figura 6.16: Resultados de 30 corridas y gráfica de convergencia para el problema 6 (representación real).

Problema 7

Representacion Binaria con Código Gray Cruza de 2 puntos y Mutación Uniforme

Prueba	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	g1(X)	g2(X)	g3(X)	g4(X)	g5(X)	g6(X)	g7(X)	g8(X)	f(X)	Factible
1	2,086057	2,865977	8,501583	5,487254	1,098755	1,584183	1,275236	9,564833	7,771776	8,101907	-0,068095	-4,616584	-0,301439	-13,762892	-0,030597	-0,228243	-7,903019	-45,150628	29,669028	SI
2	2,028188	2,694636	8,762596	4,996212	1,242886	1,961382	1,178736	9,659755	7,980033	8,759462	-0,012482	-1,994216	-0,454992	-1,031947	-0,235673	-0,219759	-6,087864	-51,228196	27,575353	SI
3	2,028451	2,636565	8,601797	5,003462	0,702739	1,002097	1,091245	9,662064	7,613651	7,658302	-0,018528	-0,035039	-0,202828	-6,511660	-1,571982	-1,945477	-7,972965	-42,082887	28,370580	SI
4	2,101364	2,754120	8,459292	5,081862	1,247014	2,047454	1,142005	9,565452	7,984390	8,352862	-0,160894	-1,302504	-0,086444	-12,181140	-0,003996	-0,848221	-6,880934	-48,246481	27,853149	SI
5	2,169427	2,295623	8,586232	4,694216	0,728421	0,887836	1,360897	9,879279	8,172989	8,113054	-0,013355	-0,047409	-0,125334	-3,342049	-0,802782	-0,208297	-6,488454	-49,166818	26,379666	SI
6	2,074557	2,176098	8,709270	5,018414	1,057466	1,697976	1,378314	9,992803	8,169796	8,404599	-0,020999	-0,108947	-0,204482	-0,694206	-3,768962	-0,046405	-4,134601	-51,653312	26,379451	SI
7	2,151236	2,340074	8,720295	5,090834	0,855183	1,144512	1,397734	9,866811	8,329332	8,834922	-0,086588	-1,236085	-0,552927	-1,738127	-0,921984	-0,103500	-6,335772	-52,956204	26,598306	SI
8	2,088123	2,402701	8,719868	4,964167	0,921841	1,303988	1,282828	9,830786	8,049844	8,203815	-0,005412	-0,486879	-0,057991	-1,226610	-1,507760	-0,267830	-6,176742	-49,245055	25,473916	SI
9	2,257423	2,141675	8,679385	5,212983	0,889864	1,385948	1,561914	9,976557	8,472793	8,344929	-0,158659	-1,158585	-0,101928	-2,681761	-0,633672	-0,390819	-5,615032	-49,654323	26,326207	SI
10	2,135749	2,311350	8,335128	4,720933	0,946418	1,355480	1,361659	9,884152	8,357829	8,700430	-0,027864	-0,513215	-0,075002	-12,145583	-2,691129	-0,000660	-5,770564	-51,905653	26,782692	SI
11	2,107398	2,761131	8,303146	5,235601	0,763923	0,895576	1,221060	9,460072	7,872847	8,016270	-1,287285	-2,852952	-0,005231	-18,501893	-0,872031	-0,716366	-8,713871	-45,675284	31,462253	SI
12	2,241159	2,354565	8,341199	4,999984	1,050486	1,584038	1,372574	9,817351	8,328500	8,301680	-0,024100	-0,123987	-0,181007	-14,007862	-0,568256	-0,077110	-6,276443	-49,412903	25,521489	SI
13	2,096198	2,465431	8,744072	4,986968	1,196730	1,897741	1,283958	9,791495	7,909132	7,872492	-0,016467	-1,005763	-0,038047	-0,820363	-0,750323	-0,140996	-5,464014	-46,504364	26,259279	SI
14	2,166301	2,642784	8,297644	4,946202	0,960655	1,386603	1,252763	9,652690	8,145039	8,357540	-0,004949	-1,470853	-0,034728	-16,328258	-0,006655	-0,801381	-7,917942	-48,892544	26,435371	SI
15	2,335686	2,024074	8,639186	5,246879	0,599996	0,872721	1,587452	9,890899	8,259785	7,344904	-1,281151	-0,040620	-0,028221	-3,309301	-0,058721	-0,834996	-5,941940	-45,467081	29,469557	SI
16	1,915136	2,770268	8,700210	4,931741	0,789022	0,940535	1,018695	9,602309	7,788785	8,592435	-0,123414	-1,123977	-0,021499	-2,902149	-2,071464	-0,353408	-7,535603	-48,735505	28,293223	SI
17	2,141183	2,362166	8,748054	5,110388	0,971868	1,411413	1,329469	9,843315	7,964844	7,755660	-0,023009	-0,399848	-0,092231	-1,028693	-0,848330	-0,130988	-6,049958	-46,525345	25,576679	SI
18	2,024200	2,619570	8,657936	5,062686	1,016894	1,467737	1,096448	9,676934	7,770076	8,042698	-0,002287	-0,000308	-0,189479	-4,938433	-1,617252	-0,309852	-6,699252	-46,019682	26,494123	SI
& 19	2,352000	2,121154	8,240078	5,003757	0,932284	1,526687	1,568981	9,958990	8,392219	7,698389	-0,062261	-0,203938	-0,009369	-15,767336	-0,360817	-0,524794	-5,909150	-46,371763	26,633399	SI
20	2,073194	2,610063	8,844462	5,318102	1,047284	1,710825	1,261028	9,715514	7,956905	8,214899	-0,000367	-2,155025	-0,010696	-0,153420	-0,174075	-1,382826	-6,993050	-48,041205	26,624419	SI
21	1,987575	2,652472	8,631898	4,990156	1,280454	2,053345	1,077969	9,666967	7,890043	8,557813	-0,018545	-0,335562	-0,261070	-5,428197	-2,081375	-0,135798	-5,428371	-49,807497	27,619995	SI
22	2,269660	2,152318	8,701784	5,183691	1,064215	1,751138	1,552193	9,979627	8,243021	7,691864	0,000296	-0,949970	-0,021268	-1,751342	-0,092418	-0,180122	-5,107223	-47,029408	26,139813	SI
23	2,159847	2,073721	8,782920	5,482030	0,807826	1,726456	1,487409	9,979138	8,451485	8,575155	-0,641993	-0,318976	-0,024222	-0,586214	-3,304944	-3,331209	-5,293917	-51,617237	28,932542	SI
24	2,184824	2,278037	8,691832	4,949787	0,935322	1,633683	1,441406	9,897963	8,237865	8,348049	-0,113068	-1,084031	-0,429291	-1,365206	-0,562030	-1,733739	-6,170753	-50,643633	26,608405	SI
25	2,124237	2,692804	8,507770	5,255516	1,027072	1,429570	1,188301	9,620766	8,001398	8,209151	-0,017044	-1,259640	-0,019599	-11,600540	-0,117790	-0,166376	-7,585114	-47,679922	26,236128	SI
26	1,960800	2,267449	8,956693	6,094277	0,832202	1,197549	1,384710	9,983965	7,631678	7,612916	-0,118004	-2,103742	-0,218943	-0,064114	-6,083243	-0,438701	-4,880436	-43,940180	33,764320	SI
27	2,229332	2,541662	8,005699	5,026931	0,964228	1,416366	1,299794	9,653075	7,808365	7,361305	-0,396075	-0,830328	-0,432116	-26,007995	-0,848144	-0,774699	-7,723349	-42,526472	29,574015	SI
28	2,155344	2,816540	8,078659	5,071393	1,239439	1,877122	1,262471	9,562922	8,281274	9,027428	-0,017047	-3,315053	-0,258157	-24,763260	-0,062113	-0,072829	-7,387335	-51,809409	29,203803	SI
29	1,933490	2,907375	8,695268	5,344556	0,965824	1,291764	1,060118	9,540890	7,925107	9,018766	-0,041508	-2,864333	-0,065167	-6,148936	-1,473730	-0,086757	-7,704381	-51,420276	28,919101	SI
30	2,086747	2,366372	8,371363	4,778260	0,801375	1,259765	1,275637	9,846168	8,274996	8,851072	-0,032554	-0,056998	-0,288395	-11,659862	-3,229613	-1,592413	-6,512401	-53,112038	27,467176	SI

MEDIA	27,621448	
DESV. EST.	1,877870237	
MEJOR	25,473916	SI
PEOR	33,764320	SI
MEDIANA	26,708046	SI
VARIANZA	3,526396627	

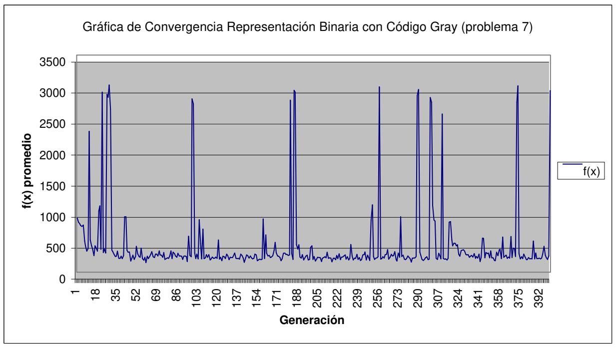

Figura 6.17: Resultados de 30 corridas y gráfica de convergencia para el problema 7 (representación binaria).

Problema 7

Representacion Real SBX y Parameter Based Mutation

Prueba	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	g1(X)	g2(X)	g3(X)	g4(X)	g5(X)	g6(X)	g7(X)	g8(X)	f(X)	Factible
1	2,118842	2,565427	8,665749	5,039088	0,941451	1,239497	1,179561	9,692962	7,980950	8,042471	0,000477	-0,001610	-0,000075	-4,285406	-0,001081	0,000767	-7,170499	-47,256910	25,089834	SI
2	2,090719	2,772495	8,476994	5,086778	0,991999	1,310549	1,053138	9,548227	7,965368	8,323312	-0,000025	-0,079662	-0,000541	-11,656871	-0,002579	-0,003720	-7,885021	-47,885978	26,128601	SI
3	2,436742	2,040965	8,062096	5,135436	0,933634	1,512588	1,650592	10,000000	8,582742	7,750983	0,000015	-0,020358	-0,000262	-21,702060	-0,002363	-0,010186	-5,747008	-45,246262	27,134418	SI
4	2,136325	2,453076	8,785347	5,100696	1,009871	1,438481	1,251897	9,771741	8,206078	8,422531	0,000655	-0,000127	0,000874	-0,087974	0,000805	0,000611	-6,401676	-50,138621	24,381843	SI
5	2,184375	2,574314	8,395829	5,096144	1,292524	2,048937	1,211699	9,669638	8,113105	8,119578	0,000714	-0,010377	-0,000002	-13,866294	-0,000310	-0,013653	-6,061173	-47,790774	26,082248	SI
6	2,072973	2,831775	8,459479	5,094444	0,986328	1,291479	1,005679	9,507435	8,066599	8,705944	0,000651	-0,006135	0,000874	-12,406358	0,000437	0,000250	-8,078626	-50,116650	26,527524	SI
7	2,138097	2,445800	8,767951	5,043828	0,968025	1,344446	1,257006	9,776698	8,101097	8,145751	0,000655	-0,001126	0,000807	-0,267106	-0,002415	-0,004104	-6,521197	-48,637103	24,508429	SI
8	2,168312	2,410409	8,704168	5,051723	1,118587	1,707632	1,293527	9,795072	8,199456	8,235449	0,000362	0,000026	0,000704	-2,361522	0,000234	-0,000170	-5,896033	-49,213234	24,585612	SI
9	1,990493	2,805703	8,802582	5,054792	0,998398	1,344654	0,974919	9,548334	7,777529	8,287258	0,000739	-0,017642	0,000589	-0,261352	0,000818	0,000556	-7,444481	-46,554147	26,389872	SI
10	2,457155	1,968040	8,102033	5,175325	0,751120	1,151765	1,697117	10,000000	8,468407	7,325371	-0,422532	-0,023769	-0,029861	-20,054660	0,000259	-0,026814	-5,839932	-44,207955	28,884972	SI
11	2,099382	2,541665	8,787092	5,068717	0,958828	1,290855	1,182266	9,715758	8,056743	8,285726	0,000880	-0,006508	0,000537	-0,185126	0,000792	0,000820	-6,870672	-49,009599	24,710303	SI
12	2,178719	2,450707	8,615541	5,089946	0,974729	1,353306	1,277081	9,762566	8,123476	8,044174	0,000258	-0,003710	0,000692	-5,871823	0,000898	0,000655	-6,758741	-47,958179	24,685641	SI
13	2,479496	1,895810	8,119782	5,199330	0,994024	1,782074	1,744616	10,000000	8,596909	7,471171	-0,836812	-0,029988	-0,002143	-18,966892	0,000801	-0,007798	-4,724608	-44,086214	29,715859	SI
14	1,995100	2,791539	8,805751	5,053029	0,935177	1,198042	0,984663	9,557380	7,768410	8,231845	0,000527	-0,005823	0,000637	-0,114792	0,000625	-0,001084	-7,624210	-46,215375	26,419201	SI
15	2,050420	2,654252	8,801446	5,059462	1,008035	1,381630	1,091773	9,644583	7,892930	8,184572	-0,001128	-0,000784	0,000649	0,000449	-0,015698	-0,001643	-7,012402	-47,380179	25,330198	SI
16	2,228562	2,487042	8,352198	5,131125	0,983944	1,355334	1,284550	9,722767	8,176209	8,013402	0,000709	-0,002526	-0,000170	-15,190231	-0,000638	-0,000937	-7,218065	-47,484650	25,151913	SI
17	2,034502	2,926977	8,453836	5,066619	1,272441	1,960322	0,930963	9,446702	7,739786	8,138271	0,000322	-0,003769	0,000325	-12,506749	-0,000115	0,000131	-7,006666	-44,697004	28,319347	SI
18	2,041110	2,677931	8,805528	5,070667	0,986757	1,328526	1,073879	9,629802	7,898918	8,260337	0,000677	-0,008697	0,000898	-0,000045	-0,016249	-0,003149	-7,157540	-47,755491	25,422426	SI
19	2,093154	2,651279	8,649059	5,070721	0,986789	1,325814	1,117657	9,635857	8,047817	8,399494	-0,001248	-0,007142	-0,002583	-5,370133	-0,007221	-0,009290	-7,321046	-49,140814	25,198479	SI
20	2,168312	2,410409	8,704168	5,051723	1,118587	1,707632	1,293527	9,795072	8,199456	8,235449	0,000362	0,000026	0,000704	-2,361522	0,000234	-0,000170	-5,896033	-49,213234	24,585612	SI
21	2,178892	2,360806	8,743843	5,076127	1,070242	1,617123	1,327093	9,829108	8,202718	8,151611	0,000287	0,000102	0,000848	-0,893015	0,000713	0,000698	-5,864305	-48,939981	24,463747	SI
22	2,492095	2,024789	7,827579	5,297390	0,930154	1,525863	1,697145	10,000000	8,681025	7,759172	0,000890	-0,128821	-0,000406	-30,009119	-0,003727	-0,013180	-5,958876	-44,076221	28,438060	SI
23	2,578901	1,993061	7,192206	5,309460	0,866312	1,416588	1,760555	10,000000	8,661536	7,330969	-0,000755	-0,084916	0,000661	-48,649477	0,000568	-0,000144	-6,415332	-41,843556	32,126096	SI
24	2,053882	2,784007	8,593023	5,047338	1,293823	2,020862	1,021434	9,547592	7,935286	8,407772	-0,000410	-0,002420	-0,002154	-7,455961	-0,006701	0,000066	-6,363492	-48,261755	26,863558	SI
25	2,024043	2,724236	8,786549	5,030784	0,977079	1,299578	1,038342	9,599521	7,864896	8,290796	-0,001984	-0,006221	-0,000987	-0,502681	-0,019062	-0,000522	-7,324355	-47,543244	25,759414	SI
26	2,147948	2,510301	8,638050	5,060537	1,173717	1,797087	1,226837	9,726390	8,094637	8,155178	0,000295	-0,006369	-0,000159	-5,167045	-0,010948	0,000021	-6,102589	-48,360815	25,022249	SI
27	2,184375	2,574314	8,395829	5,096144	1,292524	2,048937	1,211699	9,669638	8,113105	8,119578	0,000714	-0,010377	-0,000002	-13,866294	-0,000310	-0,013653	-6,061173	-47,790774	26,082248	SI
28	1,913231	3,025219	8,757866	5,055741	1,361152	2,207191	0,808570	9,405242	7,563970	8,281822	0,000482	-0,004645	0,000795	-1,964639	-0,001641	-0,000307	-6,224214	-43,279670	30,150638	SI
29	2,049356	2,661039	8,793776	5,045966	0,977508	1,308696	1,087351	9,640022	7,834009	8,048280	0,000757	0,000322	0,000716	-0,193867	0,000853	-0,000086	-7,151408	-46,189157	25,551668	SI
& 30	2,268594	2,398155	8,287949	5,076043	1,209874	1,920515	1,356260	9,778265	8,297928	8,068271	0,000759	0,000823	0,000652	-16,486793	0,000463	-0,002156	-5,972806	-47,829620	25,641759	SI

MEDIA	26,311726	
DESV. EST.	1,9197272	
MEJOR	24,381843	SI
PEOR	32,126096	SI
MEDIANA	25,700587	SI
VARIANZA	3,685352521	

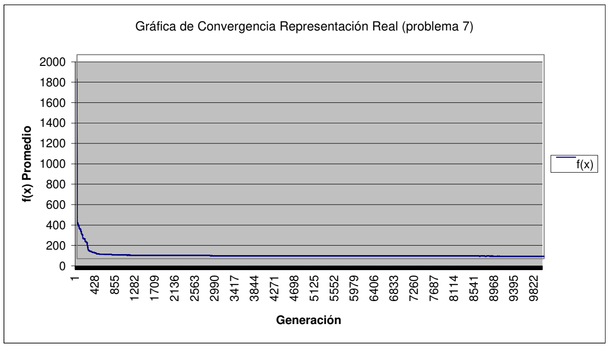

Figura 6.18: Resultados de 30 corridas y gráfica de convergencia para el problema 7 (representación real).

6.4.8 Problema 8

La solución óptima se encuentra en: [119]: $\mathbf{f(x) = 7049.3307}$, con $x = (579.3167, 1359.943, 5110.071, 182.0174, 295.5985, 217.9799, 286.4162, 395.5979)$, con g_1, g_2 y g_3 activas.

Representación binaria

El mejor valor encontrado por el algoritmo es: $\mathbf{f(x) = 7057.377407}$, con $x = (703.958047, 1130.767301, 5222.652058, 204.860812, 306.637296, 227.775050, 328.142785, 415.516117)$, y con $g_1 = 0.081590$, $g_2 = 0.074798$, $g_3 = 0.088788$, $g_4 = -2564.431170$, $g_5 = -12182.618004$ y $g_6 = -85229.439285$.

Representación real

Para la representación real no se muestran los resultados, ya que en ninguna de las 30 corridas de prueba se encontraron individuos factibles.

6.4.9 Problema 9

La solución óptima se encuentra en: [119]: $\mathbf{f(x) = 1}$, con $x = (5, 5, 5)$.

Representación binaria

El mejor valor encontrado por el algoritmo es: $\mathbf{f(x) = 1.000000}$, con $x = (5.000000, 5.000000, 5.000000)$.

Representación real

El mejor valor encontrado por el algoritmo es: $\mathbf{f(x) = 1.000000}$, con $x = (5.000000, 5.000000, 5.000000)$.

6.5 Discusión de resultados

Con base en los resultados obtenidos en los nueve problemas descritos anteriormente y en la complejidad de cada uno de ellos se observa que la técnica ha mejorado los resultados en 4 de las nueve funciones propuestas (1,2,3,4) e iguala el valor de óptimo en una más (9). En los problemas 6, 7 y 8 encontró valores cercanos al óptimo. En el problema 5, la técnica obtuvo resultados

Problema 8

Representacion Binaria con Código Gray Cruza de 2 puntos y Mutación Uniforme

Prueba	X1	X2	X3	X4	X5	X6	X7	X8	g1(X)	g2(X)	g3(X)	g4(X)	g5(X)	g6(X)	f(X)	Factible
1	-	-	-	-	-	-	-	-	-	-	-	-	-	-		NO
2	-	-	-	-	-	-	-	-	-	-	-	-	-	-		NO
3	-	-	-	-	-	-	-	-	-	-	-	-	-	-		NO
4	-	-	-	-	-	-	-	-	-	-	-	-	-	-		NO
5	100,129125	1848,010692	5446,429451	24,927159	267,599893	167,056937	189,165147	376,117739	-0,520040	0,079595	0,085178	-69275,073199	-172,640317	-10034,521110	7394,569268	SI
6	1187,404737	2558,356195	5792,151491	149,007160	257,403477	278,714574	289,704157	362,134898	0,069304	-0,004749	0,047314	-171367,352816	-224457,636903	-128,948508	9537,912423	SI
7	241,910046	1192,457730	7267,165645	152,059479	199,343794	280,766027	216,279319	303,433401	0,082064	-0,341091	0,040896	-346,341799	-17474,050751	-4795,897918	8701,533420	SI
8	-	-	-	-	-	-	-	-	-	-	-	-	-	-		NO
9	172,557922	8013,909189	2831,418582	86,712194	379,755290	186,227818	136,998312	486,355893	-0,317650	0,075104	0,066006	-25952,535039	-36684,512089	-1219,152091	11017,885694	SI
10	-	-	-	-	-	-	-	-	-	-	-	-	-	-		NO
11	-	-	-	-	-	-	-	-	-	-	-	-	-	-		NO
12	-	-	-	-	-	-	-	-	-	-	-	-	-	-		NO
13	450,573033	3179,969297	3571,233111	140,116102	351,570490	197,178449	223,891425	459,621738	-0,156764	0,088365	0,080512	-10356,017150	-2084,970672	-14802,420673	7201,775440	SI
14	2310,498930	5794,633105	2526,492115	257,821617	393,605886	158,683969	292,899560	502,384098	0,041264	0,071710	0,087782	-4071,442543	-33533,470737	-8842,009083	10631,624150	SI
15	-	-	-	-	-	-	-	-	-	-	-	-	-	-		NO
16	174,056601	4960,107256	3402,311359	133,409910	360,354608	274,162893	208,618138	466,000428	0,018932	0,088907	0,056458	-2472,718118	-89360,008232	-10326,491689	8536,475216	SI
17	165,780685	6171,130768	2863,111134	99,545082	377,998542	199,184527	157,441277	484,591205	-0,253176	0,089737	0,065927	-16822,058195	-9218,165165	-182,996752	9200,022587	SI
& 18	114,379181	1093,370803	7925,015397	80,286525	158,525535	173,683931	186,194475	267,406291	-0,365074	-0,338916	0,088808	-24855,868038	-17997,898014	-9195,506376	9132,765381	SI
19	-	-	-	-	-	-	-	-	-	-	-	-	-	-		NO
20	-	-	-	-	-	-	-	-	-	-	-	-	-	-		NO
21	-	-	-	-	-	-	-	-	-	-	-	-	-	-		NO
22	703,958047	1130,767301	5222,652058	204,860812	306,637296	227,775050	328,142785	415,516117	0,081590	0,074798	0,088788	-2564,431170	-12182,618004	-85229,439285	7057,377407	SI
23	-	-	-	-	-	-	-	-	-	-	-	-	-	-		NO
24	113,776455	3641,108093	5434,266822	71,586258	289,475343	137,809358	148,675526	391,925563	-0,476511	-0,083588	0,024502	-27979,990920	-8329,002421	-30430,190536	9189,151371	SI
25	-	-	-	-	-	-	-	-	-	-	-	-	-	-		NO
26	-	-	-	-	-	-	-	-	-	-	-	-	-	-		NO
27	100,171257	1002,842591	6618,562789	101,784975	211,803081	330,627176	323,885010	320,744656	0,081030	0,084758	0,089416	-21614,817291	-85208,741525	-544,356586	7721,576638	SI
28	2131,076441	1039,057958	4169,294454	259,406140	336,026485	173,309082	355,899287	443,658802	0,081788	0,081299	0,076323	-23389,018771	-4486,540381	-38817,035433	7339,428853	SI
29	2525,885111	8209,047175	2929,407258	110,497593	385,177033	233,643849	157,692345	490,956811	-0,139646	0,080929	0,057798	-328821,104419	-44074,649264	-22814,630834	13664,339544	SI
30	481,753185	4908,763225	4058,813166	152,859109	369,904984	216,306931	208,969419	470,299506	-0,077085	0,065038	0,003945	-11982,101396	-4124,885988	-82245,065415	9449,329575	SI

MEDIA	9051,717798	
DESV. EST.	1758,962062	
MEJOR	7057,377407	SI
PEOR	13664,339544	SI
MEDIANA	9132,765381	SI
VARIANZA	3093947,537	

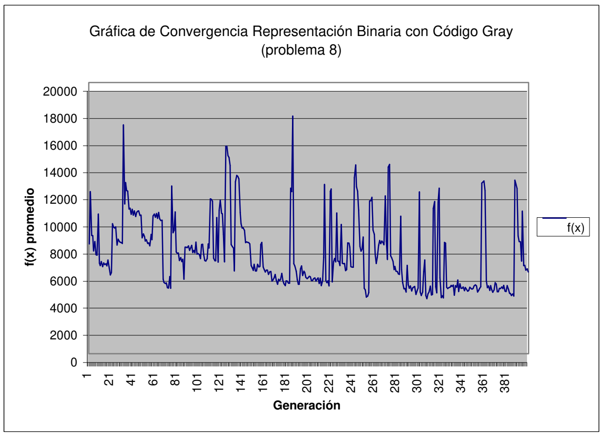

Figura 6.19: Resultados de 30 corridas y gráfica de convergencia para el problema 8 (representación binaria).

Problema 9

Representacion Binaria con Código Gray Cruza de 2 puntos y Mutación Uniforme

Prueba	X1	X2	X3	g(X)	f(X)	Factible
&1	5.000000	5.000000	5.000000	0.000000	1.000000	SI
2	5.000000	5.000000	5.000000	0.000000	1.000000	SI
3	5.000000	5.000000	5.000000	0.000000	1.000000	SI
4	5.000000	5.000000	5.000000	0.000000	1.000000	SI
5	5.000000	5.000000	5.000000	0.000000	1.000000	SI
6	5.000000	5.000000	5.000000	0.000000	1.000000	SI
7	5.000000	5.000000	5.000000	0.000000	1.000000	SI
8	5.000000	5.000000	5.000000	0.000000	1.000000	SI
9	5.000000	5.000000	5.000000	0.000000	1.000000	SI
10	5.000000	5.000000	5.000000	0.000000	1.000000	SI
11	5.000000	5.000000	5.000000	0.000000	1.000000	SI
12	5.000000	5.000000	5.000000	0.000000	1.000000	SI
13	5.000000	5.000000	5.000000	0.000000	1.000000	SI
14	5.000000	5.000000	5.000000	0.000000	1.000000	SI
15	5.000000	5.000000	5.000000	0.000000	1.000000	SI
16	5.000000	5.000000	5.000000	0.000000	1.000000	SI
17	5.000000	5.000000	5.000000	0.000000	1.000000	SI
18	5.000000	5.000000	5.000000	0.000000	1.000000	SI
19	5.000000	5.000000	5.000000	0.000000	1.000000	SI
20	5.000000	5.000000	5.000000	0.000000	1.000000	SI
21	5.000000	5.000000	5.000000	0.000000	1.000000	SI
22	5.000000	5.000000	5.000000	0.000000	1.000000	SI
23	5.000000	5.000000	5.000000	0.000000	1.000000	SI
24	5.000000	5.000000	5.000000	0.000000	1.000000	SI
25	5.000000	5.000000	5.000000	0.000000	1.000000	SI
26	5.000000	5.000000	5.000000	0.000000	1.000000	SI
27	5.000000	5.000000	5.000000	0.000000	1.000000	SI
28	5.000000	5.000000	5.000000	0.000000	1.000000	SI
29	5.000000	5.000000	5.000000	0.000000	1.000000	SI
30	5.000000	5.000000	5.000000	0.000000	1.000000	SI

MEDIA	1.000000	
DESV. EST.	0	
MEJOR	1.000000	SI
PEOR	1.000000	SI
MEDIANA	1.000000	SI
VARIANZA	0	

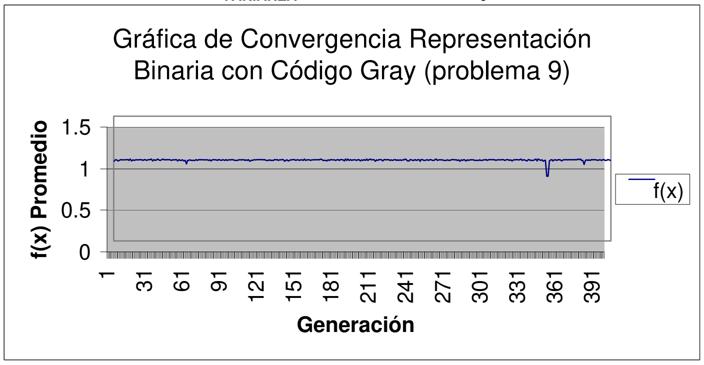

Figura 6.20: Resultados de 30 corridas y gráfica de convergencia para el problema 9 (representación binaria).

Problema 9

Representacion Real SBX y Parameter Based Mutation

Prueba	X1	X2	X3	g(X)	f(X)	Factible
&1	5.000000	5.000000	5.000000	0.000000	1.000000	SI
2	5.000000	5.000000	5.000000	0.000000	1.000000	SI
3	5.000000	5.000000	5.000000	0.000000	1.000000	SI
4	5.000000	5.000000	5.000000	0.000000	1.000000	SI
5	5.000000	5.000000	5.000000	0.000000	1.000000	SI
6	5.000000	5.000000	5.000000	0.000000	1.000000	SI
7	5.000000	5.000000	5.000000	0.000000	1.000000	SI
8	5.000000	5.000000	5.000000	0.000000	1.000000	SI
9	5.000000	5.000000	5.000000	0.000000	1.000000	SI
10	4.185661	5.934453	6.845973	0.000000	0.950560	SI
11	5.000000	5.000000	5.000000	0.000000	1.000000	SI
12	5.820390	5.156173	5.013707	0.000000	0.993024	SI
13	5.000000	5.000000	5.000000	0.000000	1.000000	SI
14	5.000000	5.000000	5.000000	0.000000	1.000000	SI
15	5.000000	5.000000	5.000000	0.000000	1.000000	SI
16	7.813395	3.145391	4.080695	0.000000	0.878001	SI
17	5.000000	5.000000	5.000000	0.000000	1.000000	SI
18	5.000000	5.000000	5.000000	0.000000	1.000000	SI
19	6.856061	3.174209	5.894284	0.000000	0.924218	SI
20	5.000000	5.000000	5.000000	0.000000	1.000000	SI
21	5.000000	5.000000	5.000000	0.000000	1.000000	SI
22	5.000000	5.000000	5.000000	0.000000	1.000000	SI
23	5.000000	5.000000	5.000000	0.000000	1.000000	SI
24	5.000000	5.000000	5.000000	0.000000	1.000000	SI
25	5.000000	5.000000	5.000000	0.000000	1.000000	SI
26	5.000000	5.000000	5.000000	0.000000	1.000000	SI
27	5.000000	5.000000	5.000000	0.000000	1.000000	SI
28	5.000000	5.000000	5.000000	0.000000	1.000000	SI
29	5.000000	5.000000	5.000000	0.000000	1.000000	SI
30	5.000000	5.000000	5.000000	0.000000	1.000000	SI

MEDIA	0.991527	
DESV. EST.	0.02688787	
MEJOR	1.000000	SI
PEOR	0.878001	SI
MEDIANA	1.000000	SI
VARIANZA	0.00072296	

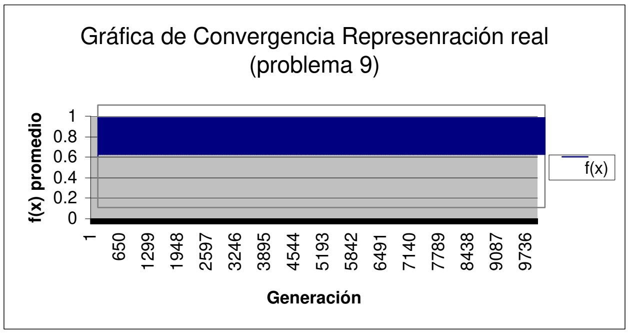

Figura 6.21: Resultados de 30 corridas y gráfica de convergencia para el problema 9 (representación real).

consistentes con una varianza mínima.

Con respecto a la representación, la técnica no parece hacer diferencia en entre ambas (binaria y real) con respecto a los resultados que obtienen para los problemas propuestos. Sin embargo, la manera en la que convergen es distinta. Puede notarse en las gráficas de convergencia de cada problema, donde la representación real tiende a converger de manera más uniforme que su contraparte binaria, la cual mantiene una diversidad mayor en el valor promedio de aptitud de los individuos de la población. Este comportamiento es similar en las nueve funciones de prueba.

Otro punto destacable es el porcentaje de individuos factibles por generación. De manera regular, para cada problema, se presentó alrededor de un 45% de soluciones factibles en cada generación. Esto denota que la técnica pudo mantenerse tanto dentro como fuera de la zona factible y evitar converger hacia una única región de la misma, dentro del espacio de búsqueda del problema. Por ejemplo, para las funciones 3, 7 y 8, con una alta dificultad para generar soluciones que cumplieran con todas las restricciones y para los problemas 2,4, 5 y 6 con menor dificultad para esa misma tarea, se mantuvo la habilidad de la técnica de mantenerse en la frontera de la zona factible.

En espacios altamente restringidos, esta propuesta logró alcanzar la zona factible y recorrerla de manera significativa (problemas 1,3,7,8 y 9). Lo que es más, en problemas con zonas factibles disjuntas (problema 9), el algoritmo logró encontrar el óptimo global de manera consistente.

En los problemas donde esta propuesta no arrojó resultados mejores a los reportados previamente en la literatura fue en aquellos dimensionalmente grandes (funciones 5, 7 y 8). Las técnicas de cruza y mutación utilizadas para ambas representaciones sugieren que la generación de distintos individuos a nivel del genotipo es suficiente. Sin embargo, parece no bastar el proceso de exploración proporcionado por la mutación.

Para evaluar los parámetros del algoritmo se realizó un análisis de sensibilidad con respecto al tamaño de la población, número máximo de generaciones, porcentaje de cruza, porcentaje de mutación y la probabilidad de selección. Se observaron los siguientes comportamientos:

- Disminuir el número de generaciones del algoritmos disminuye también la calidad de la solución final encontrada, generando una varianza mayor entre resultados. Por otro lado, el aumentar el número de generaciones no mejora de manera significativa la calidad de las soluciones encontradas y mantienen una varianza similar a la de los parámetros propuestos para las pruebas previamente presentadas.

- Disminuir el número de individuos de la población no afecta de manera significativa al desempeño final de la técnica. Aumentar la cantidad de individuos parece favorecer el aumento de la calidad de la solución final de problemas con alta dimensionalidad.

- La disminución del porcentaje de cruza no beneficia al desempeño fuera de línea ni al desempeño en línea. El aumento de este porcentaje tampoco beneficia ni afecta de manera significativa a ambos desempeños.

- Con respecto al porcentaje de mutación, el disminuirlo no ocasiona mayores cambios en el comportamiento del algoritmo. Aun más, al aumentarlo se obtienen leves mejorías en los resultados finales para funciones con dimensionalidad baja.

- El disminuir o aumentar el porcentaje de selección ($0.7 \leq S_r \leq 0.99$) parece afectar la distribución favorable (45% en promedio) de individuos factibles y no factibles y por lo tanto, un valor menor a 0.7 parece ocasionar que el algoritmo no sea guiado hacia la zona factible del espacio de búsqueda.

- El tamaño del torneo parece seguir un comportamiento no tan sensible como el reportado por Horn y Nafpliotis [57]. En pruebas donde se disminuyó el tamaño del torneo, se notó una varianza mayor en los resultados finales, aunque en calidad los resultados finales fueron buenos. Al aumentar el tamaño del torneo se observó una varianza mayor también. Sin embargo, los resultados no fueron mejores que aquellos encontrados con un torneo de tamaño 10. Esto indica que la excesiva presión de selección de un torneo mayor a 10 individuos ocasiona que la técnica encuentre la zona factible demasiado rápido y que depende de la zona donde llegue la búsqueda, la calidad de la solución final. Cabe señalar que la técnica no muestra una sensibilidad tan evidente a este parámetro, ya que el proceso de selección no depende únicamente del torneo basado en no dominancia.

Con base en los resultados empíricos obtenidos de este análisis se presentan las siguientes sugerencias con respecto al ajuste de parámetros de la técnica:

Representación binaria.

Parámetro	**Valor**
Número máximo de generaciones	400
Tamaño de la población	200
Porcentaje de cruza	0.6
Porcentaje de mutación	$0.001 \leq P_m \leq 0.05$*
Tamaño del torneo	10
Porcentaje de selección	$0.8 \leq S_r \leq 0.99$*

Representación real.

Parámetro	**Valor**
Número máximo de generaciones	$10,000$
Tamaño de la población	50
Porcentaje de cruza	0.8
Porcentaje de mutación	$0.6 \leq P_m \leq 0.8$*
Tamaño del torneo	10
Porcentaje de selección	$0.8 \leq S_r \leq 0.99$*

* Tanto la probabilidad de mutación como la de selección pueden aumentarse dependiendo de la dificultad del problema.

6.6 Comparación de resultados con otras técnicas

A continuación se muestran los mejores resultados obtenidos por esta propuesta junto con aquellos obtenidos con funciones de penalización. Se muestran resultados de un algoritmo genético que utiliza pena de muerte para manejar restricciones. Además se comparan los resultados de un algoritmo genético con una función de penalización dinámica ($C = 0.5$, $\beta = 2.0$, $\alpha = 2.0$). Para ambas técnicas de comparación se utilizaron tanto representación real como binaria y selección por torneo binario determinístico.

Variables de diseño	**Mejor solución encontrada**		
	Esta propuesta	**Pena de muerte**	**Penalización dinámica**
$x_1(h)$	0.205708	0.199034	0.202594
$x_2(l)$	3.470923	4.148923	3.531309
$x_3(t)$	9.03674	9.922478	9.056843
$x_4(b)$	0.205729	0.220198	0.205668
$g_1(\vec{x})$	-0.062627	-2256.430600	-0.001166
$g_2(\vec{x})$	-0.730609	-6752.450000	-124.915290
$g_3(\vec{x})$	-0.000021	-0.021164	-0.003074
$g_4(\vec{x})$	-3.432928	-3.088111	-3.424637
$g_5(\vec{x})$	-0.080708	-0.074034	-0.077594
$g_6(\vec{x})$	-0.235541	-0.239795	-0.235633
$g_7(\vec{x})$	-0.025913	-1806.922800	-3.456265
$f(\vec{x})$	**1.724896**	**2.089308**	**1.731183**

Tabla 6.1: Comparación de resultados para el problema 1 (Diseño de una viga soldada).

Los parámetros para ambas representaciones fueron los siguientes:

Representación binaria.

Parámetro	**Valor**
Número máximo de generaciones	400
Tamaño de la población	200
Porcentaje de cruza	0.6
Porcentaje de mutación	0.03

Representación real.

Parámetro	**Valor**
Número máximo de generaciones	$10,000$
Tamaño de la población	50
Porcentaje de cruza	0.8
Porcentaje de mutación	0.6

Variables de diseño	**Mejor solución encontrada**		
	Esta propuesta	**Pena de muerte**	**Penalización dinámica**
$x_1(T_s)$	0.812500	0.812500	0.812500
$x_2(T_h)$	0.437500	0.437500	0.437500
$x_3(R)$	42.097398	41.775549	42.089722
$x_4(L)$	176.654047	183.501291	176.744283
$g_1(\vec{x})$	-0.000020	-0.006232	-0.000168
$g_2(\vec{x})$	-0.035891	-0.038961	-0.035964
$g_3(\vec{x})$	-27.886075	-15475.136000	-0.559649
$g_4(\vec{x})$	-63.345953	-56.498709	-63.255717
$f(\vec{x})$	**6059.946341**	**6164.945160**	**6060.766670**

Tabla 6.2: Comparación de resultados para el problema 2 (Diseño de un recipiente de presión).

Variables de diseño	**Mejor solución encontrada**		
	Esta propuesta	**Pena de muerte**	**Penalización dinámica**
$x_1(d)$	0.051989	0.053743	0.052153
$x_2(D)$	0.363965	0.405514	0.367968
$x_3(N)$	10.890522	9.920849	10.672107
$g_1(\vec{x})$	-0.000013	-0.103354	0.000000
$g_2(\vec{x})$	-0.000021	-0.005334	-0.000005
$g_3(\vec{x})$	-4.061338	-3.626788	-4.069062
$g_4(\vec{x})$	-0.722698	-0.693829	-0.719920
$f(\vec{x})$	**0.012681**	**0.013962**	**0.012683**

Tabla 6.3: Comparación de resultados para el problema 3 (Minimización del peso de un resorte).

Variables de diseño	**Mejor solución encontrada**		
	Esta propuesta	**Pena de muerte**	**Penalización dinámica**
x_1	856.013342	924.174963	677.164914
x_2	0.100000	1.086153	0.100000
x_3	601.548206	443.898253	567.528224
x_4	172.969615	82.466387	182.277405
x_5	0.100000	0.952399	0.100000
x_6	0.000081	1.317064	0.100000
x_7	65.334218	46.758902	71.767986
x_8	539.767385	845.476949	744.768935
x_9	474.838781	942.097669	464.225660
x_{10}	0.100000	1.269604	0.100000
$f(\vec{x})$	**5152.636136**	**5742.304130**	**5248.362090**

Tabla 6.4: Comparación de resultados para el problema 4 (Diseño de una armadura de 10 barras).Todas las soluciones son factibles

Puede observarse que NPGA para manejo de restricciones proporciona mejores resultados consistentemente en las nueve funciones de prueba. Sólo en la función 9 iguala el valor del óptimo global.

Por otro lado, el comportamiento de esta propuesta es mucho más consistente, ya que tanto para la pena de muerte como para la función de penalización dinámica las varianzas en los resultados de las 30 corridas fueron mucho más altas. Además, el porcentaje de soluciones factibles fue, o mucho mayor o mucho menor tanto para la pena de muerte como para la función de penalización. Estos números indican que estas dos propuestas sólo encuentran la zona factible del espacio de búsqueda, pero son incapaces de recorrerla de manera suficiente y, en otros casos, no la pueden alcanzar con facilidad.

Variables de diseño	**Mejor solución encontrada**		
	Esta propuesta	**Pena de muerte**	**Penalización dinámica**
x_1	3.088605	3.058503	3.011848
x_2	3.034861	3.231301	3.127577
x_3	3.112081	3.111636	3.155891
x_4	3.079128	3.265051	0.696045
x_5	3.090021	3.034504	0.953469
x_6	2.949984	3.166241	2.275257
x_7	2.929546	2.937620	0.394465
x_8	2.905113	0.634435	3.043771
x_9	0.485143	3.446524	3.015541
x_{10}	1.372498	0.236528	0.413372
x_{11}	0.421162	1.321508	0.584528
x_{12}	0.479062	0.683721	2.957208
x_{13}	0.406244	0.143664	0.398566
x_{14}	0.502223	0.581542	0.460141
x_{15}	0.489684	0.710560	0.462809
x_{16}	0.378411	0.731854	0.411567
x_{17}	0.325555	0.656109	0.782174
x_{18}	0.345997	0.215301	0.731054
x_{19}	0.519377	0.894552	0.527680
x_{20}	0.364493	0.140838	0.613111
$g_1(\vec{x})$	-0.004704	-0.273670	-0.000492
$g_2(\vec{x})$	-119.720812	-117.798910	-121.983930
$f(\vec{x})$	**0.787933**	**0.653277**	**0.627836**

Tabla 6.5: Comparación de resultados para el problema 5.

Variables de diseño	**Mejor solución encontrada**		
	Esta propuesta	**Pena de muerte**	**Penalización dinámica**
x_1	78.003942	78.047573	78.071211
x_2	33.019521	33.006316	33.062146
x_3	30.020488	30.056935	30.182713
x_4	44.990450	44.996413	44.580623
x_5	36.732249	36.610017	36.586178
$g_1(\vec{x})$	-0.003615	-0.022641	-0.043730
$g_2(\vec{x})$	-91.996385	-91.977359	-91.956270
$g_3(\vec{x})$	-11.156384	-11.182611	-11.141734
$g_4(\vec{x})$	-8.843616	-8.817389	-8.858266
$g_5(\vec{x})$	-4.997549	-4.888446	-4.981513
$g_6(\vec{x})$	-0.002451	-0.000154	-0.018487
$f(\vec{x})$	$\mathbf{-30659.996878}$	$\mathbf{-30653.271076}$	$\mathbf{-30612.626000}$

Tabla 6.6: Comparación de resultados para el problema 6.

Variables de diseño	**Mejor solución encontrada**		
	Esta propuesta	**Pena de muerte**	**Penalización dinámica**
x_1	2.136325	-	2.260300
x_2	2.453076	-	2.186539
x_3	8.785347	-	8.316685
x_4	5.100696	-	5.124482
x_5	1.009871	-	0.748543
x_6	1.438481	-	1.216792
x_7	1.251897	-	1.511889
x_8	9.771741	-	9.943160
x_9	8.206078	-	7.947944
x_{10}	8.422531	-	7.071328
$g_1(\vec{x})$	0.000655	-	-0.073334
$g_2(\vec{x})$	-0.000127	-	-0.705110
$g_3(\vec{x})$	0.000874	-	-0.112262
$g_4(\vec{x})$	-0.087974	-	-14.686736
$g_5(\vec{x})$	0.000805	-	-1.844837
$g_6(\vec{x})$	0.000611	-	-1.527063
$g_7(\vec{x})$	-6.401676	-	-6.486482
$g_8(\vec{x})$	-50.13862	-	-43.128446
$f(\vec{x})$	**24.381843**	$-$	**28.936289**

Tabla 6.7: Comparación de resultados para el problema 7. El símbolo "-" significa que la técnica se estancó en la generación 0 por no encontrar ninguna solución factible.

Variables de diseño	**Mejor solución encontrada**		
	Esta propuesta	**Pena de muerte**	**Penalización dinámica**
x_1	703.958047	-	399.656632
x_2	1130.767301	-	2646.328920
x_3	5222.652058	-	4679.965400
x_4	204.860812	-	102.158812
x_5	306.637296	-	313.585028
x_6	227.775050	-	125.864665
x_7	328.142785	-	219.523076
x_8	415.516117	-	421.451646
$g_1(\vec{x})$	0.081590	-	-0.429941
$g_2(\vec{x})$	0.074798	-	0.077373
$g_3(\vec{x})$	0.088788	-	0.078666
$g_4(\vec{x})$	-2564.431170	-	-8538.057400
$g_5(\vec{x})$	-12182.618004	-	-46301.676000
$g_6(\vec{x})$	-85229.439285	-	-38774.609000
$f(\vec{x})$	**7057.377407**	$-$	**7725.950950**

Tabla 6.8: Comparación de resultados para el problema 8.El símbolo "-" significa que la técnica se estancó en la generación 0 por no encontrar ninguna solución factible.

Variables de diseño	**Mejor solución encontrada**		
	Esta propuesta	**Pena de muerte**	**Penalización dinámica**
x_1	5.000000	5.000000	5.000004
x_2	5.000000	5.000004	5.000003
x_3	5.000000	5.000004	5.000001
$g(\vec{x})$	0.000000	0.000000	0.000000
$f(\vec{x})$	**1.000000**	**1.000000**	**1.000000**

Tabla 6.9: Comparación de resultados para el problema 9.

Conclusiones

La propuesta presentada en este trabajo refrenda la idea de que el uso de técnicas diseñadas para optimización multiobjetivo, pueden aplicarse con éxito a problemas con un solo objetivo y restricciones.

La técnica presenta un bajo costo, tanto computacional como de implementación, generando buenos resultados en el conjunto de funciones de prueba que incluyen problemas con una gama diversa de situaciones, demostrando robustez.

En lo que respecta al tipo de representación, tanto la binaria como la real, junto con sus correspondientes operadores genéticos lograron muestrear adecuadamente el espacio de búsqueda y coadyuvar a localizar la zona factible. Pudo observarse que para problemas con dimensionalidad grande, la representación real pudo encontrar mejores resultados y para problemas con pocas variables, la representación binaria fue mejor.

Con respecto a los parámetros del algoritmo, de acuerdo con el análisis empírico de sensibilidad, no se observó una dependencia marcada hacia alguno de los parámetros. Por otro lado, se observó que el nuevo parámetro propuesto (S_r) debe mantenerse en un rango mayor a 0.8 para que la técnica funcione de manera adecuada.

En la comparación con otras técnicas de penalización, NPGA para manejo de restricciones mostró un mejor desempeño, tanto en los mejores resultados encontrados como en la consistencia de sus soluciones ya que logra muestrear con mayor precisión la zona factible del espacio de búsqueda de problemas de distintas características.

Como conclusión general podemos decir que esta técnica presenta una alternativa para manejo de restricciones donde no se deben definir factores de penalización por parte del usuario y que, a pesar de ser una técnica multiobjetivo donde su principal problema es mantener diversidad durante el proceso evolutivo, evita la convergencia prematura y logra encontrar de manera consistente la zona factible y la muestrea lo suficiente para obtener resultados competitivos a un costo computacional y de implementación bajos.

Trabajo futuro

Algunas de las extensiones posibles a este trabajo son las siguientes:

- Refinar la técnica resolviendo los desempates haciendo diferencia entre dos candidatos dominados y dos candidatos no dominados.
- Buscar nuevas alternativas para mantener diversidad sin la necesidad de agregar más parámetros al algoritmo.
- Extender el dominio de esta propuesta para resolver problemas con múltiples objetivos y restricciones.
- Resolver funciones con restricciones de igualdad.
- Analizar la factibilidad de acoplar esta técnica con algún algoritmo de optimización de otro tipo (tal vez otra heurística).

Bibliografía

[1] David H. Ackley, editor. *A Connectionist Machine for Genetic Hillclimbing.* Kluwer Academic Publishers, Boston, Massachusetts, 1995.

[2] Hojjat Adeli and Nai-Tsang Cheng. Augmented Lagrangian Genetic Algorithm for Structural Optimization. *Journal of Aerospace Engineering*, 7(1):104–118, January 1994.

[3] Robin Allenson. Genetic algorithms with gender for multi-function optimisation. Technical Report EPCC-SS92-01, Edinburgh Parallel Computing Centre, Edinburgh, Scotland, 1992.

[4] Thomas Bäck, Frank Hoffmeister, and Hans-Paul Schwefel. A Survey of Evolution Strategies. In R. K. Belew and L. B. Booker, editors, *Proceedings of the Fourth International Conference on Genetic Algori thms*, pages 2–9, San Mateo, California, 1991. Morgan Kaufmann Publishers.

[5] E. Baldwin. *Genética elemental.* Limusa, 1983.

[6] James C. Bean. Genetics and random keys for sequencing and optimization. Technical Report TR 92-43, Department of Industrial and Operations Engineering, The University of Michigan, 1992.

[7] James C. Bean. Genetics and random keys for sequencing and optimization. *ORSA Journal on Computing*, 6(2):154–160, 1994.

[8] James C. Bean and Atidel Ben Hadj-Alouane. A Dual Genetic Algorithm for Bounded Integer Programs. Technical Report TR 92-53, Department of Industrial and Operations Engineering, The University of Michigan, 1992. To appear in R.A.I.R.O.-R.O. (invited submission to special issue on GAs and OR).

[9] A. D. Belegundu, D. V. Murthy, R. R. Salagame, and E. W. Constant s. Multiobjective Optimization of Laminated Ceramic Composites U sing Genetic

Algorithms. In *Fifth AIAA/USAF/NASA Symposium on Multidisciplinary Analysis and Op timization*, volume Paper 84-4363-CP, pages 1015–1022, Panama City, Florida, 1994. AIAA.

[10] Sheela V. Belur. CORE: Constrained Optimization by Random Evolution. In John R. Koza, editor, *Late Breaking Papers at the Genetic Programming 1997 Conference*, pages 280–286, Stanford University, California, July 1997. Stanford Bookstore.

[11] George Bilchev and Ian C. Parmee. The Ant Colony Metaphor for Searching Continuous Design Spaces. In Terence C. Fogarty, editor, *Evolutionary Computing*, pages 25–39. Springer Verlag, Sheffield, UK, April 1995.

[12] George Bilchev and Ian C. Parmee. Constrained and Multi-Modal Optimisation with an Ant Colony Search Model. In Ian C. Parmee and M. J. Denham, editors, *Proceedings of 2nd International Conference on Adaptive Computing in Engineering Design and Control.* University of Plymouth, Plymouth, UK, March 1996.

[13] Hans J. Bremermann. Optimization through evolution and recombination. In M. C. Yovits, G. T. Jacobi, and G. D. Goldstein, editors, *Self-Organizing Systems—1962*, pages 93–106. Spartan Books, Washington, D.C., 1962.

[14] Bill P. Buckles and Fred E. Petry, editors. *Genetic Algorithms.* IEEE Computer Society Press, 1992.

[15] Eduardo Camponogara and SaroshÑ. Talukdar. A Genetic Algorithm for Constrained and Multiobjective Optimization. In Jarmo T. Alander, editor, *3rd Nordic Workshop on Genetic Algorithms and Their Applications (3NWGA)*, pages 49–62, Vaasa, Finland, August 1997. University of Vaasa.

[16] A. Charnes and W. W. Cooper. *Management Models and Industrial Applications of Linear Programming*, volume 1. John Wiley, New York, 1961.

[17] A. Chipperfield and P. Fleming. Gas Turbine Engine Controller Design using Multiobjective G enetic Algorithms. In A. M. S. Zalzala, editor, *Proceedings of the First IEE/IEEE International Conference on Genet ic Algorithms in Engineering Systems : Innovations and Applications, GALESIA'95*, pages 214–219, Halifax Hall, University of Sheffield, UK, September 1995. IEEE.

[18] Chan-Jin Chung and Robert G. Reynolds. A Testbed for Solving Optimization Problems using Cultural Algorithms. In Lawrence J. Fogel, Peter J. Angeline, and Thomas Bäck, editors, *Evolutionary Programming V:*

Proceedings of the Fifth Annual Conference on Evolutionary Programming, Cambridge, Massachusetts, 1996. MIT Press.

[19] Carlos A. Coello Coello. *An empirical study of evolutionary techniques for multiobjetive optimization in engineering design.* PhD thesis, Department of Computer Science, Tulane University, New Orleans, Louisiana, April 1996.

[20] Carlos A. Coello Coello. A Comprehensive Survey of Evolutionary-Based Multiobjective Optimization Techniques. *Knowledge and Information Systems. An International Journal*, 1(3):269–308, August 1999.

[21] Carlos A. Coello Coello. Constraint-handling through a Multiobjective Optimization Technique. In Annie S. Wu, editor, *Proceedings of the 1999 Genetic and Evolutionary Computation Conference (GECCO'99) Workshop Program*, pages 117–118, Orlando, Florida, USA, 1999.

[22] Carlos A. Coello Coello. A Survey of Constraint Handling Techniques used with Evolutionary Algorithms. Technical Report Lania-RI-99-05, Laboratorio Nacional de Informática avanzada, Xalapa, Veracruz, México, 1999.

[23] Carlos A. Coello Coello. An Updated Survey of Evolutionary Multiobjective Optimization Techniques : State of the Art and Future Trends. In *1999 Congress on Evolutionary Computation*, pages 3–13, Washington, D.C., July 1999. IEEE Service Center.

[24] Carlos A. Coello Coello. Constraint-Handling using an Evolutionary Multiobjective Optimization Technique. Technical Report Lania-RI-2000-03, Laboratorio Nacional de Informática Avanzada, Xalapa, Veracruz, México, 2000.

[25] Carlos A. Coello Coello. Functions used to test constraint-handling techniques. Technical Report Lania-RI-2000-04, Laboratorio Nacional de Informática avanzada, Xalapa, Veracruz, México, 2000.

[26] Carlos A. Coello Coello. Treating Constrains as Objetives for Single-Objetive Evolutionary Optimization. *Engineering Optimization*, 32(2):275–308, 2000.

[27] Carlos A. Coello Coello. Use of a self-adaptative penalty approach for engineering optimization problems. *Computers in Industry*, 41(2):113–127, 2000.

[28] Carlos A. Coello Coello, Filiberto Santos Hernández, and Francis co Alonso Farrera. Optimal design of reinforced concrete beams using genetic algorithms. *Expert Systems with Applications : An International Journal*, 12(1):101–108, January 1997.

[29] Nareli Cruz Cortés. Uso de emulaciones del sistema inmune para manejo de restricciones en algoritmos evolutivos. Master's thesis, Universidad Veracruzana, Xalapa, México, 2000.

[30] Dragan Cvetković, Ian Parmee, and Eric Webb. Multi-objective Optimisation and Preliminary Airframe Design, April 1998.

[31] Charles Darwin. *The Origin of Species by Means of Natural Selection or the preservation of Favored Races in the Struggle for life*. Random House, New York, 1993. (Publicado originalmente en 1929).

[32] Lawrence Davis, editor. *Handbook of Genetic Algorithms*. Van Nostrand Reinhold, New York, 1991.

[33] K. Deb and S. Agrawal. A Niched-Penalty Approach for Constrtaint Handling in Genetic Algorithms. In *Proceedings of the ICANNGA*, Portoroz, Slovenia, 1999.

[34] Kalyanmoy Deb. Genetic algorithms in multimodal function optimization. Master's thesis, University of Alabama, Alabama, USA, 1989.

[35] Kalyanmoy Deb and Mayank Goyal. A Combined Genetic Adaptive Search GeneAS for Engineering Design. *Computer Science and Informatics*, 26(4):30–45, 1996.

[36] David B. Fogel. An introduction to simulated evolutionary optimization. *IEEE Transactions on neural networks*, 5(1):3–14, January 1994.

[37] David B. Fogel. *Evolutionary Computation. Toward a New Philosophy of Machine Intelligence*. The Institute of Electrical and Electronic Engineers, New York, 1995.

[38] Lawrence J. Fogel. *Intelligence Through Simulated Evolution. Forty years of Evolutionary Programming*. John Wiley & Sons, New York, 1999.

[39] Carlos M. Fonseca and Peter J. Fleming. Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generali zation. In Stephanie Forrest, editor, *Proceedings of the Fifth International Conference on Genetic Algorithms*, pages 416–423, San Mateo, California, 1993. University of Illinois at Urbana-Champaign, Morgan Kauffman Publishers.

[40] M. P. Fourman. Compaction of symbolic layout using genetic algorithms. In *Genetic Algorithms and their Applications: Proceedings of the First International Conference on Genetic Algorithms*, pages 141–153. Lawrence Erlbaum, 1985.

[41] A. S. Fraser. Simulation of Genetic Systems by Automatic Digital Computers I. Introduction. *Australian Journal of Biological Sciences*, 10:484–491, 1957.

[42] G. J. Friedman. Digital simulation of an evolutionary process. *General Systems: Yearbook of the Society for General Systems Research*, 4:171–184, 1959.

[43] C. Fujiki. An evaluation of Holland's genetic operators applied to a program generator. Master's thesis, University of Idaho, Moscow, Idaho, 1986.

[44] Fred Glover and G. Kochenberger. Critical event tabu search for multidimensional knapsack problems. In *Proceedings of the International Conference on Metaheuristics for Optimization*, pages 113–133, Dordrecht, The Netherlands, 1995. Kluwer Publishing.

[45] David E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison Wesley Longman Inc., Alabama, 1989.

[46] David E. Goldberg and Kalyanmoy Deb. A comparison of selection schemes used in genetic algorithms. In G.J. E. Rawlins, editor, *Foundations of Genetic Algorithms*, pages 69–93. Morgan Kaufmann, San Mateo, California, 1991.

[47] David E. Goldberg and J. Richardson. Genetic algorithm with sharing for multimodal function optimization. In J. J. Grefenstette, editor, *Genetic Algorithms and Their Applications: Proceedings of the Secon d International Conference on Genetic Algorithms*, pages 41–49. Lawrence Erlbaum, 1987.

[48] David E. Goldberg and Robert E. Smith. Nonstationary Function Optimization Using Genetic Algorithms with Dominance and Diploidy. In John J. Grefenstette, editor, *Genetic Algorithms and Their Applications: Proceedings of the Second International Conference on Genetic Algorithms*, pages 59–68. Lawrence Erlbaum Associates, Hillsdale, New Jersey, July 1987.

[49] Atidel Ben Hadj-Alouane and James C. Bean. A Genetic Algorithm for the Multiple-Choice Integer Program. *Operations Research*, 45:92–101, 1997.

[50] P. Hajela and J. Lee. Constrained Genetic Search via Schema Adaptation. An Immune Network Solution. In Niels Olhoff and George I.Ñ. Rozvany, editors, *Proceedings of the First World Congress of Stuctural and Multidisciplinary Optimization*, pages 915–920, Goslar, Germany, 1995. Pergamon.

[51] P. Hajela and J. Lee. Constrained Genetic Search via Schema Adaptation. An Immune Network Solution. *Structural Optimization*, 12:11–15, 1996.

[52] P. Hajela and C. Y. Lin. Genetic search strategies in multicriterion optimal design. *Structural Optimization*, 4:99–107, 1992.

[53] J. F. Hicklin. Application of the genetic algorithm to automatic program generation. Master's thesis, University of Idaho, Moscow, Idaho, 1986.

[54] John H. Holland. *Adaptation in Natural and Artificial Systems. An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence.* Ann Harbor : University of Michigan Press, 1975.

[55] John H. Holland. *Adaptation in Natural an Artificial Systems.* MIT Press, Cambridge, Massachusetts, second edition, 1992.

[56] A. Homaifar, S. H. Y. Lai, and X. Qi. Constrained Optimization via Genetic Algorithms. *Simulation*, 62(4):242–254, 1994.

[57] J. Horn and N.Ñafpliotis. Multiobjective Optimization using the Niched Pareto Genetic Algorithm. Technical Report IlliGAl Report 93005, University of Illinois at Urbana-Champaign, Urbana, Illinois, USA, 1993.

[58] J. P. Ignizio. *Goal Programming and Extensions.* Heath, Lexington, Massachusetts, 1976.

[59] J. P. Ignizio. The determination of a subset of efficient solutions via goal programming. *Computing and Operations Research*, 3:9–16, 1981.

[60] Y. Ijiri. *Management Goals and Accounting for Control.* North-Holland, Amsterdan, 1965.

[61] Hisao Ishibuchi and Tadahiko Murata. Multi-Objective Genetic Local Search Algorithm. In Toshio Fukuda and Takeshi Furuhashi, editors, *Proceedings of the 1996 International Conference on Evolutionary Co mputation*, pages 119–124, Nagoya, Japan, 1996. IEEE.

[62] W. Jakob, M. Gorges-Schleuter, and C. Blume. Application of genetic algorithms to task planning and learning. In R. Männer and B. Manderick, editors, *Parallel Problem Solving from Nature, 2nd Workshop*, Lecture Notes in Computer Science, pages 291–300, Amsterdam, 1992. North-Holland Publishing Company.

[63] John Jenkins, editor. *Genetics.* Houghton Mifflin Company, Boston, Massachusetts, 1984.

[64] Fernando Jiménez and José L. Verdegay. Evolutionary techniques for constrained optimization problems. In *7th European Congress on Intelligent Techniques and Soft Computing (EUFIT'99)*, Aachen, Germany, 1999. Springer-Verlag.

[65] Xidong Jin and Robert G. Reynolds. Using Knowledge-Based Evolutionary Computation to Solve Nonlinear Constraint Optimization Problems: a Cultural Algorithm Approach. In *1999 Congress on Evolutionary Computation*, pages 1672–1678, Washington, D.C., July 1999. IEEE Service Center.

[66] J. Joines and C. Houck. On the use of non-stationary penalty functions to solve nonlinear constr ained optimization problems with GAs. In David Fogel, editor, *Proceedings of the first IEEE Conference on Evolutionary Computation*, pages 579–584, Orlando, Florida, 1994. IEEE Press.

[67] Gareth Jones, Robert D. Brown, David E. Clark, Peter Willett, and Robert C. Glen. Searching Databases of Two-Dimensional and Three-Dimensional Chemical Structures using Genetic Algorithms. In Stephanie Forrest, editor, *Proceedings of the Fifth International Conference on Genetic Algorithms*, pages 597–602, San Mateo, California, 1993. Morgan Kaufmann.

[68] A. K. De Jong. *An Analysis of the Behavior of a Class of Genetic Adaptative Systems*. PhD thesis, University of Michigan, 1975.

[69] H. Jutler. Liniejnaja modiel z nieskolkimi celevymi funkcjami (linear model with s everal objective functions). *Ekonomika i matematiceckije Metody*, 3:397–406, 1967. (In Polish).

[70] S. Kazarlis and V. Petridis. Varying Fitness Functions in Genetic Algorithms: Studying the Rate of Increase of the Dynamic Penalty Terms. In A. E. Eiben, T. Bäck, M. Schoenauer, and H.-P. Schwefel, editors, *Parallel Problem Solving from Nature V—PPSN V*, Amsterdam, The Netherlands, 1998. Springer-Verlag.

[71] J.-H. Kim and H. Myung. Evolutionary programming techniques for constrained optimization problems. *IEEE Transactions on Evolutionary Computation*, 1:129–140, July 1997.

[72] S. Kirkpatrick, Jr. C. D. Gelatt, and M. P. Vecchi. Optimization by Simulated Annealing. *Science*, 220:671–680, 1983.

[73] Ryszard Kowalczyk. Constraint Consistent Genetic Algorithms. In *Proceedings of the 1997 IEEE Conference on Evolutionary Computation*, pages 343–348, Indianapolis, USA, April 1997. IEEE.

[74] John R. Koza. *Genetic Programming. On the Programming of Computers by Means of Natural Selection.* MIT Press, Cambridge, Massachusetts, 1992.

[75] Slawomir Koziel and Zbigniew Michalewicz. A Decoder-based Evolutionary Algorithm for Constrained Parameter Optimization Problems. In T. Bäck, A. E. Eiben, M. Schoenauer, and H.-P. Schwefel, editors, *Proceedings of the 5th Parallel Problem Solving from Nature (PPSN V)*, pages 231–240, Amsterdam, September 1998. Springer-Verlag.

[76] Slawomir Koziel and Zbigniew Michalewicz. Evolutionary Algorithms, Homomorphous Mappings, and Constrained Parameter Optimization. *Evolutionary Computation*, 7(1):19–44, 1999.

[77] Frank Kursawe. A variant of evolution strategies for vector optimization. In H. P. Schwefel and R. Männer, editors, *Parallel Problem Solving from Nature. 1st Workshop, PPSN I*, volume 496 of *Lecture Notes in Computer Science*, pages 193–197, Berlin, Germany, oct 1991. Springer-Verlag.

[78] T. Van Le. A Fuzzy Evolutionary Approach to Constrained Optimization Problems. In *Proceedings of the Second IEEE Conference on Evolutionary Computation*, pages 274–278, Perth, November 1995. IEEE.

[79] J. G. Lin. Maximal vectors and multi-objective optimization. *Journal of Optimization Theory and Applications*, 18(1):41–64, jan 1976.

[80] Joanna Lis and A. E. Eiben. A Multi-Sexual Genetic Algorithm for Multi-objective Optimiz ation. In Toshio Fukuda and Takeshi Furuhashi, editors, *Proceedings of the 1996 International Conference on Evolutionary Co mputation*, pages 59–64, Nagoya, Japan, 1996. IEEE.

[81] Xiaojian Liu, D. W. Begg, and R. J. Fishwick. Genetic approach to optimal topology/controller design of adaptive structures. *International Journal for Numerical Methods in Engineering*, 41:815–830, 1998.

[82] Daniel H. Loughlin and S. Ranjithan. The Neighborhood constraint method: A Genetic Algorithm-Based Multiobjective Optimization Technique. In Thomas Bäck, editor, *Proceedings of the Seventh International Conference on Genetic Algorithms*, pages 666–673, San Mateo, California, July 1997. Michigan State University, Morgan Kaufmann Publishers.

[83] S. M. Mahfoud. Crowding and preselection revisited. In R. Männer and B. Manderick, editors, *Parallel problem Solving from Nature, 2nd Workshop*, Amsterdam, 1992. North-Holland Publishing Company.

[84] Zbigniew Michalewicz. Genetic Algorithms, Numerical Optimization, and Constraints. In Larry J. Eshelman, editor, *Proceedings of the Sixth International Conference on Genetic Algorit hms*, pages 151–158, San Mateo, California, July 1995. University of Pittsburgh, Morgan Kaufmann Publishers.

[85] Zbigniew Michalewicz. *Genetic Algorithms + Data Structures = Evolution Programs*. Springer-Verlag, third edition, 1996.

[86] Zbigniew Michalewicz and Naguib F. Attia. Evolutionary Optimization of Constrained Problems. In *Proceedings of the 3rd Annual Conference on Evolutionary Programming*, pages 98–108. World Scientific, 1994.

[87] Zbigniew Michalewicz and G.Ñazhiyath. Genocop III: A co-evolutionary algorithm for numerical optimization with nonlinear constraints. In David B. Fogel, editor, *Proceedings of the Second IEEE International Conference on Evolutionary Computation*, pages 647–651, Piscataway, New Jersey, 1995. IEEE Press.

[88] Zbigniew Michalewicz and Marc Schoenauer. Evolutionary Algorithms for Constrained Parameter Optimization Problems. *Evolutionary Computation*, 4(1):1–32, 1996.

[89] Zbigniew Michalewicz and Marc Schoenauer. Evolutionary algorithms for constrained parameter optimization problems. *Evolutionary Computation*, 4(1):1–36, 1996.

[90] Zbigniew Michalewicz and Jing Xiao. Evaluation of Paths in Evolutionary Planner/Navigator. In *Proceedings of the 1995 International Workshop on Biologically Inspired Evolutionary Systems*, pages 45–52, Tokyo, Japan, May 1995.

[91] E. Michielssen and D. S. Weile. Electromagnetic System Design using Genetic Algorithms. In *Genetic Algorithms and Evolution Strategies in Engineering and Comp uter Science*, pages 267–288. John Wiley and Sons, England, 1995.

[92] Angel Kuri Morales and Carlos Villegas Quezada. A Universal Eclectic Genetic Algorithm for Constrained Optimization. In *Proceedings 6th European Congress on Intelligent Techniques & Soft Computing, EUFIT'98*, pages 518–522, Aachen, Germany, September 1998. Verlag Mainz.

[93] H. Myung and J.-H. Kim. Hybrid Interior-Lagrangian Penalty Based Evolutionary Optimization. In V. W. Porto, N. Saravanan, D. Waagen, and A.E.

Eiben, editors, *Proceedings of the Seventh Anual Conference on Evolutionary Programming*, pages 85–94. Springer-Verlag, 1998.

[94] J.Ñash. The bargaining problem. *Econometrica*, 18:155–162, 1950.

[95] Shigeru Obayashi. Pareto Genetic Algorithm for Aerodynamic Design using the Nav ier-Stokes Equations. In D. Quagliarella, J. Périaux, C. Poloni, and G. Winter, editors, *Genetic Algorithms and Evolution Strategies in Engineering and Comp uter Science. Recent Advances and Industrial Applications*, chapter 12, pages 245–266. John Wiley and Sons, West Sussex, England, 1997.

[96] David Orvosh and Lawrence Davis. Shall We Repair? Genetic Algorithms, Combinatorial Optimization and Feasibility Constraints. In Stephanie Forrest, editor, *Proceedings of the Fifth International Conference on Genetic Algorit hms*, page 650. Morgan Kauffman Publishers, San Mateo, California, July 1993.

[97] David Orvosh and Lawrence Davis. Using a Genetic Algorithm to Optimize Problems with Feasibility Contraints. In *Proceedings of the First IEEE Conference on Evolutionary Computation*, pages 548–553. IEEE Press, 1994.

[98] A. Osyczka. An approach to multicriterion optimization problems for engineering des ign. *Computer Methods in Applied Mechanics and Engineering*, 15:309–333, 1978.

[99] A. Osyczka. An approach to multicriterion optimization for structural design. In *Proceedings of International Symposium on Optimal Structural Design*. University of Arizona, 1981.

[100] A. Osyczka and S. Kundu. A new method to solve generalized multicriteria optimization problems u sing the simple genetic algorithm. *Structural Optimization*, 10:94–99, 1995.

[101] Andrzej Osyczka. *Multicriterion Optimization in Engineering with FORTRAN programs*. Ellis Horwood Limited, 1984.

[102] Andrzej Osyczka. Multicriteria optimization for engineering design. In John S. Gero, editor, *Design Optimization*, pages 193–227. Academic Press, 1985.

[103] Jan Paredis. Co-evolutionary constraint satisfaction. In Julian Miller, Adrian Thompson, Peter Thomson, and Terence C. Fogarty, editors, *Proceedings of the 3rd Conference on Parallel Problem Solving from Nature*, pages 21–30, Edinburgh, Scotland, April 2000. Springer-Verlag.

[104] Vilfredo Pareto. *Cours D'Economie Politique*, volume I and II. F. Rouge, Lausanne, 1896.

[105] I.C. Parmee and G. Purchase. The Development of a Directed Genetic Search Technique for Heavily Constrained Design Spaces. In I. C. Parmee, editor, *Adaptive Computing in Engineering Design and Control PEDC-94*, pages 97–102, University of Plymouth, Uk, 1994. University of Plymouth.

[106] Jacques Périaux, Mourad Sefrioui, and Bertrand Mantel. Ga Multiple Objective Optimization Strategies for Electromagn etic Backscattering. In D. Quagliarella, J. Périaux, C. Poloni, and G. Winter, editors, *Genetic Algorithms and Evolution Strategies in Engineering and Comp uter Science. Recent Advances and Industrial Applications*, chapter 11, pages 225–243. John Wiley and Sons, West Sussex, England, 1997.

[107] Carlo Poloni and Valentino Pediroda. GA coupled with computationally expensive simulations: tools to imp rove efficiency. In D. Quagliarella, J. Périaux, C. Poloni, and G. Winter, editors, *Genetic Algorithms and Evolution Strategies in Engineering and Comp uter Science. Recent Advances and Industrial Applications*, chapter 13, pages 267–288. John Wiley and Sons, West Sussex, England, 1997.

[108] David Powell and Michael M. Skolnick. Using genetic algorithms in engineering design optimization with non-linear constraints. In Stephanie Forrest, editor, *Proceedings of the Fifth International Conference on Genetic Algorithms*, pages 424–431, San Mateo, California, July 1993. University of Illinois at Urbana-Champaign, Morgan Kaufmann Publishers.

[109] Domenico Quagliarella and Alessandro Vicini. Coupling Genetic Algorithms and Gradient Based Optimization Techniques. In D. Quagliarella, J. Périaux, C. Poloni, and G. Winter, editors, *Genetic Algorithms and Evolution Strategies in Engineering and Computer Science. Recent Advances and Industrial App lications*, chapter 14, pages 289–309. John Wiley and Sons, West Sussex, England, 1997.

[110] S. Ranjithan, J. W. Eheart, and J. C. Liebman. Incorporating fixed-cost component of pumping into stochastic groundwater management: A genetic algorithm-based optimiz ation approach. *Eos Transactions AGU*, 73(14):125, 1992. Spring meeting supplement.

[111] S. Rao. Game theory approach for multiobjective structural optimization. *Computers and Structures*, 25(1):119–127, 1986.

[112] S. S. Rao. Multiobjective optimization in structural design with uncertain paramet ers and stochastic processes. *AIAA Journal*, 22(11):1670–1678, nov 1984.

[113] S. S. Rao. Game theory approach for multiobjective structural optimization. *Computers and Structures*, 25(1):119–127, 1987.

[114] Thomas S. Ray. An approach to the synthesis of life. In C. G. Langton, C. Taylor, J. D. Farmer, and S. Rasmussen, editors, *Artificial Life II*, pages 371–408. Addison-Wesley, Reading, Massachusetts, 1992.

[115] Jon T. Richardson, Mark R. Palmer, Gunar Liepins, and Mike Hilliard. Some Guidelines for Genetic Algorithms with Penalty Functions. In J. David Schaffer, editor, *Proceedings of the Third International Conference on Genetic Algorithms*, pages 191–197, George Mason University, 1989. Morgan Kaufmann Publishers.

[116] Rodolphe G. Le Riche, Catherine Knopf-Lenoir, and Raphael T. Haftka. A Segregated Genetic Algorithm for Constrained Structural Op timization. In Larry J. Eshelman, editor, *Proceedings of the Sixth International Conference on Genetic Algorithms*, pages 558–565, San Mateo, California, July 1995. University of Pittsburgh, Morgan Kaufmann Publishers.

[117] Brian J. Ritzel, J. Wayland Eheart, and S. Ranjithan. Using genetic algorithms to solve a multiple objective groundwater p ollution containment problem. *Water Resources Research*, 30(5):1589–1603, may 1994.

[118] Katya Rodríguez-Vázquez, Carlos M. Fonseca, and Peter J. Fl eming. Multiobjective Genetic Programming : A Nonlinear System Ide ntification Application. In John R. Koza, editor, *Late Breaking Papers at the Genetic Programming 1997 Conference*, pages 207–212, Stanford University, California, July 1997. Stanford Bookstore.

[119] Thomas P. Runarsson and Xin Yao. Stochastic Ranking for Constrained Evolutionary Optimization. *IEEE Transactions on Evolutionary Computation*, 4(3):284–294, September 2000.

[120] Eric Sandgren. Multicriteria design optimization by goal programming. In Hojjat Adeli, editor, *Advances in Design Optimization*, pages 225–265. Chapman & Hall, London, 1994.

[121] J. David Schaffer. Some experiments in machine learning using vector evaluated genetic algortihms. No Publicado, 1984.

[122] J. David Schaffer. Multiple Objective Optimization with Vector Evaluated Genetic Algorithms. In *Genetic Algorithms and their Applications: Proceedings of the First International Conference on Genetic Algorithms*, pages 93–100. Lawrence Erlbaum, 1985.

[123] Marc Schoenauer and Zbigniew Michalewicz. Evolutionary Computation at the Edge of Feasibility. In H.-M. Voigt, W. Ebeling, I. Rechenberg, and H.-P. Schwefel, editors, *Proceedings of the Fourth Conference on Parallel Problem Solving from Nature*, pages 245–254. Springer-Verlag, Berlin, September 1996.

[124] Marc Schoenauer and Zbigniew Michalewicz. Sphere Operators and Their Applicability for Constrained Parameter Optimization Problems. In V. W. Porto, N. Saravanan, D. Waagen, and A. E. Eiben, editors, *Evolutionary Programming VII: Proceedings of the Seventh Annual Conference on Evolutionary Programming*, volume 1447 of *Lecture Notes in Computer Science*, pages 241–250. Springer-Verlag, San Diego, California, March 1998.

[125] Marc Schoenauer and Spyros Xanthakis. Constrained GA Optimization. In Stephanie Forrest, editor, *Proceedings of the Fifth International Conference on Genetic Algorithms*, pages 573–580, San Mateo, California, 1993. Morgan Kaufmann Publishers.

[126] Hans-Paul Schwefel. *Numerical Optimization of Computer Models*. John Wiley & Sons, Great Britain, 1981.

[127] Hans-Paul Schwefel, editor. *Evolution and Optimization Seeking*. John Wiley & Sons, New York, 1995.

[128] Robert E. Smith, Sthephanie Forrest, and Alan S. Perelson. Population diversity in an immune system model: Implications for gene tic search. In L. Darrell Whitley, editor, *Foundations of Genetic Algorithms 2*, pages 153–165. Morgan Kaufmann Publishers, San Mateo, California, 1993.

[129] R. Solich. Zadanie programowania liniowego z wieloma funkcjami celu (linear progra mming problem with several objective functions). *Przeglad Statystyczny*, 16:24–30, 1969. (In Polish).

[130] N. Srinivas and K. Deb. Multiobjective optimization using nondominated sorting in genetic algor ithms. Technical report, Department of Mechanical Engineering, Indian Institute of Technol ogy, Kanput, India, 1993.

[131] N. Srinivas and Kalyanmoy Deb. Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms. *Evolutionary Computation*, 2(3):221–248, fall 1994.

[132] Patrick D. Surry and Nicholas J. Radcliffe. The COMOGA Method: Constrained Optimisation by Multiobjective Genetic Algorithms. *Control and Cybernetics*, 26(3), 1997.

[133] Patrick D. Surry, Nicholas J. Radcliffe, and Ian D. Boyd. A Multi-Objective Approach to Constrained Optimisation of Gas Supply Networks : The COMOGA Method. In Terence C. Fogarty, editor, *Evolutionary Computing. AISB Workshop. Selected Papers*, Lecture Notes in Computer Science, pages 166–180. Springer-Verlag, Sheffield, U.K., 1995.

[134] Gilbert Syswerda. Uniform Crossover in Genetic Algorithms. In J. David Schaffer, editor, *Proceedings of the Third International Conference on Genetic Alg orithms*, pages 2–9, San Mateo, California, jun 1989. George Mason University, Morgan Kaufmann Publishers.

[135] Gilbert Syswerda and Jeff Palmucci. The Application of Genetic Algorithms to Resource Scheduling. In Richard K. Belew and Lashon B. Booker, editors, *Proceedings of the Fourth International Conference on Genetic Algorithms*, pages 502–508, San Mateo, California, 1991. Morgan Kaufmann.

[136] T. Kang T. Ray and S.k. Chye. An Evolutionary Algorithm for Constrained Optimization. In D. Whitley, D. Goldberg, E. Cantú-Paz, L. Spector, I. Parmee, and H.-G. Beyer, editors, *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'2000)*, pages 771–777, San Francisco, California, 2000. Morgan Kaufmann.

[137] Hishashi Tamaki, Hajime Kita, and Shigenobu Kobayashi. Multi-Objective Optimization by Genetic Algorithms : A Revi ew. In Toshio Fukuda and Takeshi Furuhashi, editors, *Proceedings of the 1996 International Conference on Evolutionary Co mputation*, pages 517–522, Nagoya, Japan, 1996. IEEE.

[138] Hishashi Tamaki, M. Mori, M. Araki, and H. Ogai. Multicriteria optimization by genetic algorithms : a case of scheduling in hot rolling process. In *Proceedings of the 3rd APORS*, pages 374–381, 1995.

[139] Kay Chen Tan and Yun Li. Multi-Objective Genetic Algorithm Based Time and Frequency Domain Design Unification of Linear Control Systems. Technical Report CSC-97007, Department of Electronics and Electrical Engineering, University of Glasglow, Glasglow, Scotland, 1997.

[140] C. H. Tseng and T. W. Lu. Minimax multiobjective optimization in structural design. *International Journal for Numerical Methods in Engineering*, 30:1213–1228, 1990.

[141] Alan Mathison Turing. Computing Machinery and Intelligence. *Mind*, 59:94–101, 1950.

[142] Manuel Valenzuela-Rendón and Eduardo Uresti-Charre. A Non-Generational Genetic Algorithm for Multiobjective Opt imization. In Thomas Bäck, editor, *Proceedings of the Seventh International Conference on Genetic Algo rithms*, pages 658–665, San Mateo, California, July 1997. Michigan State University, Morgan Kaufmann Publishers.

[143] Ganesh Vedarajan, Louis Chi Chan, and David E. Goldberg. Investment Portfolio Optimization using Genetic Algorithms. In John R. Koza, editor, *Late Breaking Papers at the Genetic Programming 1997 Conference*, pages 255–263, Stanford University, California, July 1997. Stanford Bookstore.

[144] A Wetzel. Evaluation of Efectiveness of Genetic Algorithms in Combinatorial Optimization. No Publicado, 1983.

[145] D. Whitley. The genitor algorithm and selection pressure: Why rank-based allocat ion of reproductive trials is best. In J. David Schaffer, editor, *Proceedings of the Third Conference on Genetic Algorithms*, pages 116–121, San Mateo, California, jun 1989. George Mason University, Morgan Kaufmann Publishers.

[146] P. B. Wienke, C. Lucasius, and G. Kateman. Multicriteria target optimization of analytical procedures using a genetic algorithm. *Analytical Chimica Acta*, 265(2):211–225, 1992.

[147] P. B. Wilson and M. D. Macleod. Low implementation cost IIR digital filter design using genetic algorithms. In *IEE/IEEE Workshop on Natural Algorithms in Signal Processing*, pages 4/1–4/8, Chelmsford, U.K., 1993.

[148] Jing Xiao, Zbigniew Michalewicz, and Krzysztof Trojanowski. Adaptive Evolutionary Planner/Navigator for Mobile Robots. *IEEE Transactions on Evolutionary Computation*, 1(1):18–28, 1997.

[149] Jing Xiao, Zbigniew Michalewicz, and Lixin Zhang. Evolutionary Planner/Navigator: Operator Performance and Self-Tuning. In *Proceedings of the 3rd IEEE International Conference on Evolutionary Computation*, Nagoya, Japan, May 1996. IEEE Press.

[150] X. Yang and M. Gen. Evolution program for bicriteria transportation problem. In M. Gen and T. Kobayashi, editors, *Proceedings of the 16th International Conference on Computers and Industrial Engineering*, pages 451–454, Ashikaga, Japan, 1994.