Universidad del Bío-Bío

Facultad de Ciencias Empresariales

Departamento de Ciencias de la Computación y
Tecnologías de la Información

Estudio del Problema de Identificación de Parámetros en la Teoría de Sedimentación Aplicando Algoritmos Genéticos

Por

Natanael Humberto Guerrero Carrasco

Tesis para optar al grado de Magíster
en Ciencias de la Computación

Profesores guía : **Dr. Aníbal Coronel Pérez**
Dr. Gilberto A. Gutiérrez Retamal

Chillán - Chile

2013

.. a mi Reina, mi príncipe y princesa.

Agradecimientos

Hay muchas personas a las que me gustaría agradecer, pues de una u otra forma han contribuido y me han apoyado en este proceso de formación que significa mucho para mí.

Agradecer a mis directores de tesis, Dr. Gilberto Gutiérrez R., que con su docta visión crítica y exigente permitió mejorar cada vez más este trabajo. Al Dr. Aníbal Coronel P., quién me entregó todo su apoyo y conocimientos del tema. Su siempre buena disposición y entusiasmo para llegar a buen término con esta tarea fueron claves. Gracias Aníbal.

Agradecer a mis padres quienes siempre han estado muy presente en mi vida y en cada desafío que he adquirido. Sus sabios consejos y la formación valórica cristiana que me han entregado han sido indispensables para mi crecimiento.

Agradecer a mi familia, en especial a mi esposa Ana Carmen, mi Reina..., con quién tengo la dicha de compartir nuestras vidas y los hermosos hijos que me ha dado, Joaquín y Natalia..... gracias Dios.

Mis agradecimientos al proyecto interno 124109 3/R y al proyecto de grupo de investigación consolidado N° 121909 GI/C. También agradecer al profesor Richard Lagos Aguilera de la Universidad de Magallanes.

Por último y en forma especial, agradecer al programa de Magister en Ciencias de la Computación y equipo docente con quienes tuve la oportunidad de compartir y aprender. Agradecer a la comisión revisora y a mi Universidad.

Sinceramente gracias.

Resumen

En la presente tesis se investiga el diseño de una solución algorítmica utilizando estrategias evolutivas, en particular algoritmos genéticos, con el fin de resolver el problema de optimización correspondiente a la identificación de parámetros en un modelo para el proceso de sedimentación. La sedimentación, separación de mezcla sólido-fluido, corresponde a un proceso mecánico y es de gran utilidad en múltiples actividades industriales y de investigación. El fenómeno de sedimentación puede ser modelado por una ecuación diferencial parcial con condiciones iniciales y de borde, donde los coeficientes dependen de parámetros que provienen de igualdades de tipo algebraico o leyes constitutivas y que en aplicaciones experimentales deben ser calibrados. Actualmente la calibración de parámetros se realiza de forma experimental y numéricamente utilizando la metodología del gradiente. El problema de identificación se realiza considerando como observación un perfil de concentración de sólidos en tiempo fijo. Se estudió la calibración del modelo de sedimentación resolviendo un problema de optimización mediante algoritmos genéticos. Se presentan pruebas numéricas que validan el método propuesto y permiten concluir que esta estrategia supera las serias dificultades de la metodología del gradiente y de la identificación experimental aplicada y utilizada en trabajos previos.

Índice general

Dedicatoria I

Agradecimientos II

Resumen III

Lista de Figuras VII

Lista de Tablas IX

1. Introducción 1

1.1. Introducción 1

1.2. Objetivos de la Tesis 4

1.2.1. Objetivo General 4

1.2.2. Objetivos Específicos 4

1.3. Alcances y Contribución de la tesis 4

1.4. Descripción del documento 5

2. Teoría de Sedimentación: Problema Directo e Inverso 7

2.1. Teoría de Mezclas 7

2.2. Modelo de Sedimentación 8

2.3. Problema Directo. 11

2.4. Problema Inverso de Identificación de Parámetros 12

3. Computación Evolutiva, Optimización y Algoritmos Genéticos **16**

3.1. Aspectos Generales de Computación Evolutiva 16

3.2. Computación Evolutiva en Acción 17

3.3. Acerca de la Optimización 19

3.4. Algoritmos Genéticos 22

3.4.1. Características Básicas 25

3.4.2. Esquema o Secuencia Básica de un Algoritmo Genético 25

3.4.3. Un Pseudocódigo del Algoritmo Genético Básico 26

3.4.4. Componentes Básicos Necesarios Para el Algoritmo Genético 26

3.4.5. Método de Representación de un Algoritmo Genético 27

3.4.6. Método de Selección de un Algoritmo Genético 27

3.4.7. Métodos de Cambio - Cruza y Mutación en un Algoritmo Genético 28

3.4.8. Algunas Aplicaciones de Algoritmos Genéticos 28

3.5. Problemas Inversos Resueltos Mediante Algoritmos Genéticos 30

4. Investigación y Resultados **32**

4.1. Elementos a considerar 32

4.2. Considerando el algoritmo 34

4.3. Algoritmo de Identificación propuesto 35

4.3.1. Etapa1. Construcción de la función costo. 35

4.3.2. Etapa2. Construcción del Algoritmo Genético. 35

4.3.3. Esquema General del Algoritmo Implementado 40

4.4. Especificación del Algoritmo en Pseudocódigo 41

4.5. Resultados 44

4.5.1. Identificación flujo tipo Burger 44

4.5.2. Identificación flujo de sedimentación 44

4.5.3. Análisis general de la implementación y otros ejemplos 48

5. Conclusiones y Perspectivas **65**

A. Anexo **73**

Lista de Símbolos **73**

A.1. Lista de Símbolos 73

Índice de figuras

2.1. Columna y proceso de sedimentación 9
2.2. Columna de sedimentación con los sensores para obtener las observaciones 12
2.3. Representación esquemática del problema de identificación de parámetros 14
2.4. Ilustración de una función con varios mínimos 15

3.1. Clasificación de los métodos de optimización más relevantes presentados en [41]. . 21
3.2. Clasificación de las diversas Técnicas de Búsquedas. 24

4.1. Esquema del algoritmo creado. 40
4.2. Perfil para el mejor individuo y el perfil observado, después de cinco generaciones para los valores presentados en el Cuadro 4.2. 47
4.3. Valor de la función costo sobre el mejor individuo de cada generación. 48
4.4. Individuos del 1 al 4 de la primera generación 51
4.5. Individuos del 5 al 8 de la primera generación 52
4.6. Individuos del 9 al 12 de la primera generación 52
4.7. Individuos del 13 al 16 de la primera generación 53
4.8. Individuos del 17 al 20 de la primera generación 53
4.9. Individuos del 17 al 20 de la última generación 54
4.10. Individuos del 13 al 16 de la última generación 55
4.11. Individuos del 9 al 12 de la última generación 55
4.12. Individuos del 5 al 8 de la última generación 56
4.13. Individuos del 1 al 4 de la última generación, mejor perfil evaluado el individuo número 1 56
4.14. Valor de la función costo sobre el mejor individuo de cada generación 57

4.15. Perfil del mejor individuo después de cinco generaciones para los valores presentados en el Cuadro 4.5. 57
4.16. a) Valor de la función costo sobre el mejor individuo de cada generación; n=40; $MaxIter$=8; M=50. b) Perfil del mejor individuo según los siguientes valores: n=40; $MaxIter$=8; M=50. 58
4.17. a) Valor de la función costo sobre el mejor individuo de cada generación; n=100; $MaxIter$=8; M=50. b) Perfil del mejor individuo según los siguientes valores: n=100; $MaxIter$=8; M=50. 59
4.18. a) Valor de la función costo sobre el mejor individuo de cada generación; n=20; $MaxIter$=5; M=100. b) Perfil del mejor individuo según los siguientes valores: n=20; $MaxIter$=5; M=100. 60
4.19. a) Valor de la función costo sobre el mejor individuo de cada generación; n=100; $MaxIter$=5; M=100. b) Perfil del mejor individuo según los siguientes valores: n=100; $MaxIter$=5; M=100. 61
4.20. a) Valor de la función costo sobre el mejor individuo de cada generación; n=100; $MaxIter$=5; M=200. b) Perfil del mejor individuo según los siguientes valores: n=100; $MaxIter$=5; M=200. 62
4.21. Gráfico relación de tiempo requerido por el algoritmo genético para distintos tamaños de población, distintas generaciones y M=50. 63
4.22. Gráfico que muestra los costos asociados a los mejores individuos identificados por el algoritmo genético, para distintos tamaños de población y generaciones. . . . 64

Índice de cuadros

4.1. Ubicación de los sensores y valores de la concentración observada. 46
4.2. Población inicial aleatoria de 20 individuos para la identificación. 47
4.3. Población inicial aleatoria de 20 individuos. 50
4.4. Población final después de iterar 5 generaciones. 51
4.5. Individuos y costos óptimo por generación. 51

Capítulo 1

Introducción

En este capítulo se presenta los aspectos generales de la Tesis. Se inicia presentando una introducción global al trabajo, se especifican los objetivos, se señalan las contribuciones de la tesis y se describe la estructura del informe.

1.1. Introducción

El estudio de la identificación de parámetros es un área de investigación que se ha incrementando en las últimas décadas debido al creciente desarrollo de modelos matemáticos y su simulación mediate algoritmos computacionales [30, 36, 42]. A pesar que en este trabajo el interés es el estudio de un problema inverso de carácter matemático se debe señalar que este tipo de problemas no son exclusivos de las ciencias exactas. Es un tema transversal a las distintas ciencias. Por ejemplo, en la Semántica se presenta lo siguiente: si uno tiene una palabra y desconoce su significado se enfrenta a un problema directo, pero en cambio si uno tiene un concepto y se desea encontrar la palabra que define este concepto se está ante un problema inverso. Otros ejemplos de problemas inversos en Medicina e Historia son: el diagnóstico de una enfermedad a partir de sus síntomas y la reconstrucción de un suceso pasado a partir de datos actuales. Los problemas inversos que surgen en el seno de la Matemática implican la creación de métodos para resolverlos. Que cumplan con criterios de existencia, unicidad y estabilidad de las soluciones. En la mayoría de los casos, los problemas inversos son mal planteados en el sentido que, o bien tienen soluciones múltiples o bien son inconsistentes (sin solución). Por ejemplo, si se conoce la longitud de la

hipotenusa en un triángulo rectángulo entonces el problema inverso de determinar los catetos tiene infinitas soluciones.

En el ámbito de la Ingeniería, es frecuente utilizar modelos matemáticos. Estos modelos matemáticos utilizan teorías matemáticas de distinta naturaleza. Sin embargo, lo que predomina actualmente son los modelos basados en Ecuaciones Diferenciales (tanto Ordinarias como Parciales). En esta tesis el interés está centrado en un problema inverso del área de las ecuaciones diferenciales parciales. Una de las areas de problemas inversos en ecuaciones diferenciales es la calibración numérica de modelos matemáticos, entre los cuales se distinguen tres tipos: problema de asimilación de datos, problema de identificación de los coeficientes (o parámetros) y el problema de la identificación de las condiciones de frontera.

Para precisar, el problema inverso que se estudia en esta tesis corresponde a la identificación de los parámetros de un modelo matemático para el fenómeno de sedimentación de partículas sólidas inmersas en un fluido. Particularmente, se toma el concepto de sedimentación como un proceso mecánico utilizado para la separación de una mezcla de tipo sólido-fluido. Su gran presencia en procesos industriales lo convierten en un fenómeno relevante para la investigación científica [5, 13]. Los principales aspectos históricos de la evolución de esta teoría aparecen detallados en [8, 14]. En estos trabajos, se establece como modelo matemático básico para la descripción del fenómeno de separación sólido-fluido, una ecuación hiperbólica de primer orden o ley de conservación escalar. Se debe señalar que existen otros modelos para los procesos de separación, como por ejemplo aquel que lo aborda desde el punto de vista del análisis estocástico [48]. El modelo de Kynch [37] para la sedimentación de partículas tiene como coeficiente principal el flujo de sólidos. Las variables de este modelo, que son posibles de medir en el laboratorio, corresponden a la concentración de sólidos en la mezcla, la condición inicial y las tasas de alimentación y descarga de la columna (condiciones de frontera). Sin embargo, el flujo no es posible medirlo directamente [5]. Esto origina un problema inverso el cual se puede resolver, al menos de dos maneras distintas: empíricamente o numéricamente. En ambos casos la solución requiere de observaciones experimentales. En el caso de la solución empírica se necesita de un set de datos de concentración vs. velocidad y luego la función de flujo es ajustada vía técnicas de regresión. La velocidad se mide observando la evolución de la interfase entre la mezcla y el líquido claro. La desventaja de esta técnica radica en el hecho que se tiene que hacer múltiples

experimentos, en realidad uno por cada observación de velocidad que se quiera estimar. En relación a la solución numérica, la cantidad de experimentos necesarios es la óptima, en principio, teóricamente se necesita de un sólo perfil de concentración medido en un tiempo fijo anterior a la consolidación completa del proceso o un perfil de concentración en una altura fija muy cercana o en el borde de la descarga. Naturalmente la solución numérica resulta bastante más económica al sustituir los experimentos de laboratorio por simulaciones computacionales. Sin embargo, la solución numérica requiere de la elaboración de códigos computacionales con técnicas avanzadas.

En la práctica el problema inverso de identificación de parámetros se resuelve mediante la definición de un funcional que compara el experimento con la simulación. Por ejemplo, el funcional de mínimos cuadrados es: $J(f) = \int_a^b |f(x) - \bar{f}(x)|^2 dx$, donde f es la solución simulada y $\bar{f}$ es el experimento. La solución del problema inverso está dada por los mínimos del funcional. En consecuencia, las técnicas matemáticas naturales, para resolver el problema inverso, son todas las utilizadas para resolver problemas de optimización.

El método clásico para resolver el problema de optimización involucrado es el gradiente. Sin embargo, este método presenta serias dificultades que son detalladas en [2]. Entre estas se destaca el hecho de la convergencia local. En la práctica esta desventaja se traduce en que se debe poseer una estimación de la solución del problema inverso, desde la cual se hace evolucionar el gradiente. Esto último no necesariamente es cierto, al menos que a priori se conozca alguna solución experimental del problema inverso. Así, para evitar este tipo de inconveniente es necesario optar por la utilización de técnicas de optimización global.

En este trabajo se utilizará una técnica de computación evolutiva para resolver el problema de optimización. Esta es una de las áreas de las Ciencias de la Computación, la que tienen una fuerte base e influencia lógico-matemática. Por medio de sus diversos campos o disciplinas de acción, esta Ciencia, busca resolver, automatizar y optimizar procesos, algunos muy complejos como el descrito en [46]. Para precisar, la técnica evolutiva que se utilizará es la de Algoritmos Genéticos introducida en la década del setenta por Holland [28], y que actualmente tiene versiones perfeccionadas que conservan la filosofía original. Estos algoritmos simulan la evolución natural para buscar y encontrar soluciones a problemas complejos. Así mismo, los Algoritmos Genéticos, tienen una amplia gama de aplicaciones [25], de forma introductoria podemos señalar que estos algoritmos son un tipo de algoritmo matemático que transforma un conjunto o grupo de individuos

(que forman un población de posibles soluciones) donde cada uno de estos tienen asociado un valor de adaptación, en un nuevo grupo o población utilizando una serie de operadores basados en los principios de supervivencia del más apto. La representación de estos individuos o soluciones, desde el inicio de los algoritmos genéticos, se han realizado utilizando codificación binaria dada la perfecta adaptación al tipo de operaciones y el tipo de operadores que se utilizan en un algoritmo genético. Sin embargo, las representaciones binarias no son siempre efectivas por lo que se empezaron a utilizar otro tipo de representaciones. La representación seleccionada para abordar el presente trabajo es la representación real también denominada representación continua.

1.2. Objetivos de la Tesis

1.2.1. Objetivo General

Calibrar numéricamente las leyes constitutivas que definen los coeficientes de una ecuación diferencial parcial que modela la separación por sedimentación de una mezcla de tipo sólido-fluido, mediante la construcción y el análisis de un algoritmo genético.

1.2.2. Objetivos Específicos

(O_1) Investigar e implementar un algoritmo para simular la sedimentación de partículas mediante el modelo matemático de Kynch.

(O_2) Desarrollar un algoritmo genético que resuelva problemas de optimización para funciones de varias variables.

(O_3) Aplicar el algoritmo genético a la función que resuelve el problema de identificación de parámetros en el modelo de sedimentación.

1.3. Alcances y Contribución de la tesis

Los alcances de la investigación esán relacionados con: economizar recursos en procesos de calibración experimental (en laboratorio), reducir el tiempo de los procesos de calibración y mejorar las simulaciones numéricas. Además, la técnica generada se podrá aplicar a otros

problemas de identificación de parámetros mediante modelos de leyes de conservación.

En términos generales se puede señalar que se tienen las siguientes contribuciones:

1. Se implementó un algoritmo genético para funciones de varias variables y se validó este algoritmo con funciones clásicas, entre ellas las funciones de Griewank. de Rosenbrock y de Rastrigrin.

2. Se implementó un algoritmo genético continuo para la calibración del flujo en las leyes de conservación escalares de flujo convexo.

3. Se implementó un algoritmo genético continuo para la identificación de parámetros en la teoría de sedimentación, de flujo no convexo y un punto de inflección.

En base a lo investigado se generaron las siguientes actividades y publicaciones.

1. Natanael Guerrero. "Estudio del Problema de Identificación de Parámetros en la Teoría de Sedimentación Aplicando Algoritmos Genéticos". 1er Encuentro de Tesistas. Jornadas Chilenas de Computación 2010. Noviembre de 2010.

2. Natanael Guerrero, Aníbal Coronel. "A continuous genetic algorithm for flux alibration in scalar conservation laws". Enviado y publicado en Canadian Journal on Computing in Mathematics, Natural Sciences, Engineering and Medicine Vol. 2 No. 6, June 2011.

3. Natanael Guerrero, Aníbal Coronel, Gilberto Gutiérrez. "Identificación de parámetros en teoría de sedimentación vía algoritmos genéticos". Enviado.

1.4. Descripción del documento

En el Capítulo 1, se dió una extensa introducción al tema dando a conocer los conceptos básicos de un problema inverso sus alcances y la relación que tiene con el problema de identificación de parámetros en el proceso de sedimentación. También se presentan las ideas relacionadas a la sedimentación y la forma de abordar la identificación de parámetros por medio de una solución algorítmica, específicamente utilizando algoritmos genéticos, dado que este problema

pasa a ser un problema de optimización. Además, se presentan los objetivos generales y específicos del trabajo, continuado con los alcances y contribución de esta Tesis.

En el Capítulo 2, se muestra la teoría de mezclas y se exponen las hipótesis físicas que permiten encontrar el modelo de separación de tipo sólido-fluido. También se expone la forma en que se llega a establecer el problema directo y luego el problema inverso de identificación de parámetros. Además, se presenta, en mayor detalle, la descripción del problema.

En el Capítulo 3, se aborda y se investigan los aspectos relacionados a la computación evolutiva, su relación con los problemas de optimización, presentando trabajos en donde se han utilizado este tipo de estrategias algorítmicas. Se muestra en mayor detalle la relación que tienen los problemas de optimización y los problemas de búsquedas. Se presenta una clasificación de los métodos de optimización tradicionales y heurísticos. También se detalla la estrategia de Algoritmos Genéticos, dando a conocer sus orígenes, características principales y algunas aplicaciones en las cuales se utilizan.

En el capítulo 4, se muestra un estado del arte y trabajos relacionados o cercanos al problema planteado en la tesis. Además, se describen las propuestas del algoritmo genético creado para resolver el problema de identificación pasando por la etapa de construcción, ajustes, algunos pseudocódigos y resultados de pruebas estándares de validación.

Para finalizar, en el Capítulo 5 se sintetizan las conclusiones, se listan las perspectivas y se describen futuros trabajos.

Capítulo 2

Teoría de Sedimentación: Problema Directo e Inverso

En este capítulo se presenta una síntesis de la teoría de mezclas de la mecánica del medio continuo. Se exponen las hipótesis físicas que permiten establecer el modelo de separación del tipo sólido-fluido por sedimentación gravitatoria. El modelo o problema directo obtenido es una ley de conservación escalar con flujo no convexo y con un punto de inflexión. Así mismo se presenta la definición concisa del problema inverso que se estudia en la tesis.

2.1. Teoría de Mezclas

La base de la teoría de mezclas, de la mecánica del medio continuo, se encuentra en los trabajos de Fick, Stefan y Maxwell, pero la formalización, en el marco teórico, con el cual se conoce ahora fue establecida por Truesdell. Esta teoría asume que cada punto del espacio puede ser ocupado por un número finito de diferentes partículas, una de cada componente del sistema. En consecuencia, la mezcla puede ser representada como la superposición de n medios contínuos intersectados, cada una de las cuales preserva su propio movimiento con las restricciones impuestas por las interacciones.

Lo relevante para la presente tesis es la ecuación de conservación que debe satisfacer cada componente. Si se denota por G_i una propiedad extensiva por unidad de masa de la i-ésima

componente de la mezcla de densidad ρ_i, entonces

$$\frac{\partial}{\partial t}(\rho_i G_i) + \nabla \cdot J_i - r_i = 0, \tag{2.1}$$

donde J_i es el flujo de densidad y r_i es la razón de generación de la i-ésima componente por unidad de volumen. El flujo puede ser separado en su parte convectiva y difusiva y establecer la siguiente ecuación

$$\frac{\partial}{\partial t}(\rho_i G_i) + \nabla \cdot (\rho_i G_i \mathbf{v}) + \nabla \cdot J_D - r_i = 0, \tag{2.2}$$

donde $\mathbf{v}$ es la velocidad promedio de la mezcla y J_D el flujo difusivo.

Si se define la propiedad extensiva como $G_i = m_i$, siendo m_i la masa y la velocidad de difusión como $\mathbf{u}_i = \mathbf{v}_i - \mathbf{v}$, entonces de la ecuación (2.2) se obtiene la siguiente ecuación para la conservación de masa:

$$\frac{\partial}{\partial t}(\rho_i) + \nabla \cdot (\rho_i \mathbf{v}_i) - r_i = 0. \tag{2.3}$$

Similarmente, definiendo $G_i = m_i \mathbf{v}_i$, $J_D = -T_i + \rho_i \mathbf{v}_i \mathbf{u}_i$ y $r_i = \rho_i F_i + f_i$ se deduce la ecuación de conservación del momentum:

$$\frac{\partial}{\partial t}(\rho_i \mathbf{v}_i) + \nabla \cdot (\rho_i \mathbf{v}_i \mathbf{v}) + \nabla \cdot (\rho_i \mathbf{v}_i \mathbf{u}_i - T_i) - \rho_i F_i - f_i = 0. \tag{2.4}$$

En la ecuación (2.4), T_i es el tensor de esfuerzo, F_i la fuerza de cuerpo y f_i la fuerza externa.

2.2. Modelo de Sedimentación

Si se considera una mezcla de partículas de sólido inmersa en un fluido, según se muestra en la Figura 2.1 y satisfaciendo las siguientes propiedades:

(A1) Todas las partículas del sólido son pequeñas con del mismo tamaño, forma y densidad.

(A2) El sólido y el fluído de la mezcla son incompresibles.

(A3) No hay transferencia de masa entre sus componentes.

(A4) La velocidad relativa (sólido - fluido),es $\mathbf{v}_r = \mathbf{v}_s - \mathbf{v}_f$, donde $\mathbf{v}_s$ es la velocidad del sólido y $\mathbf{v}_f$ es la velocidad del fluido, la cual depende solamente de la concentración local de sólidos u.

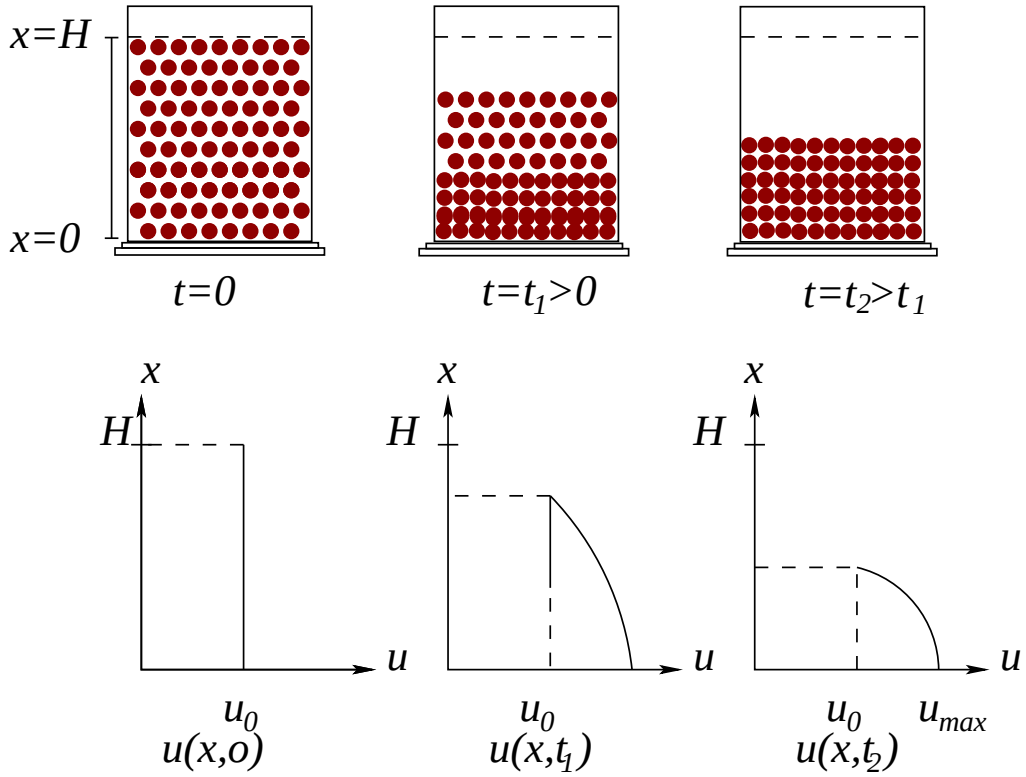


Figura 2.1: Columna y proceso de sedimentación

(A5) Los efectos de pared son despreciables.

En estas condiciones, un balance de masa para el sólido y fluído implica

$$u_t + \nabla \cdot (u\mathbf{v}_s) = 0, \tag{2.5}$$

$$(1-u)_t + \nabla \cdot ((1-u)\mathbf{v}_f) = 0, \tag{2.6}$$

respectivamente. Ahora, definiendo la velocidad volumétrica promedio $\mathbf{q}(r,t) = u\mathbf{v}_s + (1-u)\mathbf{v}_f$ y sumando ambas ecuaciones se tiene que

$$\nabla \cdot (\mathbf{q}(r,t)) = 0. \tag{2.7}$$

Así, se define el flujo total de sólidos $\mathbf{f}$ y el flujo relativo de densidad de sólidos $\mathbf{f}_b$ como

$$\mathbf{f}(u,t) = u\mathbf{v}_s, \quad \mathbf{f}_b(u) = u(1-u)\mathbf{v}_r,$$

respectivamente. Luego se tiene que

$$\mathbf{f}(u,t) = u\mathbf{q}(t) + \mathbf{f}_b(u). \tag{2.8}$$

El primer término de **f** modela el movimiento global de la mezcla y el segundo término modela la interacción sólido - fluído.

Remplazando (2.8) en (2.5) se tiene que la ecuación modelo es de la siguiente forma

$$u_t + \nabla \cdot (u\mathbf{q}(t) + \mathbf{f}_b(u)) = 0. \tag{2.9}$$

Se observa que la ecuación (2.9) es una ecuación escalar multidimensional. En 1952, Kynch introdujo condiciones que permiten simplificar (2.9) y reducirlo a un modelo unidimensional conservando propiedades relevantes del proceso físico. Esta teoría es conocida como teoría de sedimentación de Kynch y establece básicamente que:

(K1) La suspención es ideal.

(K2) La concentración de las partículas es constante a lo largo de cualquier corte transversal en la columna.

(K3) La suspención tiene una concentración inicial homogénea u_0.

(K4) En la parte inferior del recipiente existe un incremento contínuo pero rápidamente creciente de la concentración, partiendo desde u_0 hasta la concentración máxima u_{max}.

En virtud de (K2) se deduce que la concentración u depende únicamente de la altura y del tiempo, es decir $u = u(x,t)$. Esto implica que (2.9) se pueda escribir como

$$u_t + (uq(t) + f_b(u))_x = 0.$$

En el caso de sedimentación bath, es decir, cuando la columna es cerrada en la parte inferior, se tiene que $q = 0$ en $x = 0$, lo cual, en virtud de (2.7) implica que la velocidad promedio sea nula: $q = 0$. En consecuencia, para la sedimentación bath la ecuación diferencial que modela el problema es de la siguiente forma

$$u_t + (f_b(u))_x = 0, \quad x \in [0,H], \quad t > 0, \tag{2.10}$$

donde H denota la altura de la mezcla contenida en la columna. De acuerdo a las evidencias empíricas, las propiedades que debe cumplir la función de flujo son

$$f_b(u) < 0,\ u \in [0,u_{\max}],\ f_b(0) = f_b(u_{\max}) = 0,\ f_b'(0) < 0 \text{ y } f_b'(u_{\max}) > 0. \tag{2.11}$$

La ecuación (2.10) es de tipo hiperbólica no lineal de primer orden conocida también como ley de conservación y debe ser completada con condiciones iniciales y de frontera adecuados. Las condiciones (K3) y (K4) implican que

$$u(x,0) = u_0, \quad x \in [0,H], \tag{2.12}$$

$$u(0,t) = u_{\max}, \quad t > 0, \tag{2.13}$$

$$u(H,t) = 0, \quad t > 0, \tag{2.14}$$

donde u_0 y u_{max} son constantes tales que $0 < u_0 < u_{max} < 1$.

En consecuencia el modelo de sedimentación en la columna es dado por la ecuación (2.10) con condiciones iniciales y de frontera dadas por (2.12), (2.13) y (2.14) con la función de flujo satisfaciendo (2.11).

2.3. Problema Directo.

En esta sección se resume lo planteado en la sección anterior y así mismo se presenta el problema en la notación y terminología que se utilizará en las secciones posteriores.

El proceso de sedimentación es un mecanismo utilizado para separar mezclas de tipo sólido-fluido. Este tipo de separación aparece en variados procesos industriales que lo convierten en un fenómeno muy relevante para la investigación científica [5, 13]. Así, los modelos matemáticos de estos procesos son de evidente importancia teórica y práctica. Asumiendo como válidas las hipótesis consideradas por Kynch [37], en Concha y Bürger [14], también en Bürger y Wendland [8], se presentan los principales aspectos históricos de la evolución de esta teoría y se establece como modelo matemático básico, para el fenómeno de separación por sedimentación, un problema de valores iniciales y en la frontera para una ecuación hiperbólica de primer orden o ley de conservación. Más precisamente el modelo matemático toma la siguiente forma general

$$\frac{\partial \phi}{\partial t} + \frac{\partial}{\partial x}(q(t)\phi + f_b(\phi)) = 0, \quad (x,t) \in]0,L[\times]0,T[\tag{2.15}$$

$$\phi(x,0) = \phi_0, \quad x \in [0,L], \tag{2.16}$$

$$\phi(0,t) = D(t), \quad t > 0, \tag{2.17}$$

$$\phi(L,t) = I(t), \quad t > 0, \tag{2.18}$$

donde $\phi(x,t)$ es la fracción volumétrica local de sólidos en el punto x y tiempo t, $q(t)$ es la velocidad volumétrica promedio, f es la función de densidad de flujo de sólidos de Kynch, ϕ_0 es la concentración inicial de sólidos, la cual usualmente es constante, L es la altura de la mezcla, $I(t)$ y $D(t)$ son las concentraciones de sólidos que entra y sale de la columna de sedimentación en tiempo t, respectivamente. De acuerdo a evidencias empíricas, las propiedades que debe satisfacer el flujo son las dadas en (2.11). Se debe notar que el modelo matemático (2.15)-(2.14), denominado en lo que sigue como problema directo, ha sido generalizado conservando lo central de las ideas de Kynch [37], para mayores detalles consultar a Bustos [5], y la revisiones realizadas por Bürger, Wendland y Concha [8, 14]

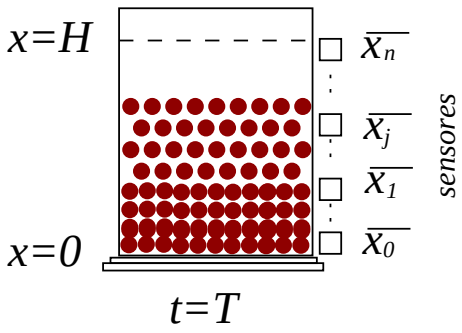


Figura 2.2: Columna de sedimentación con los sensores para obtener las observaciones

2.4. Problema Inverso de Identificación de Parámetros

Para la calibración del modelo (2.15)-(2.14) se dispone de un conjunto de mediciones tomadas a lo alto de la columna en un tiempo fijo $t = T$, ver Figura 2.2. Los n lugares donde se toman las muestras están ubicados en los puntos denotados por $\bar{x}_i$ y los valores de la concentración de sólidos que se miden en estos puntos son $u_i^{\exp}$, es decir el conjunto de datos que se disponen para la calibración del flujo es el siguiente

$$\begin{aligned} \text{Datos} \quad = \quad & \{(\bar{x}_i, \bar{\phi}_i) \in [0,H] \times [0,u_{\max}] \ : \\ & \bar{x}_0 = 0 < \bar{x}_1 < \cdots < \bar{x}_{n-1} < \bar{x}_n = H\}. \end{aligned}$$

En la práctica los datos son ajustados por una curva, es decir se define $u^{exp} : [0,H] \to [0,u_{\text{máx}}]$ tal que $u^{exp}(\bar{x}_i) = \overline{\phi}_i$, la cual se denominará el perfil experimental. En términos de esta información experimental, el problema inverso de determinación del flujo es formulado como el siguiente problema de optimización

$$\begin{cases} \text{minimizar} & J(u,f) := \dfrac{1}{2}\displaystyle\int_I |(u-u^{exp})(x)|^2 dx, \\ \text{sujeto a} & E(u,p;f) = 0,\ f \in U_{ad}, \end{cases} \tag{2.19}$$

donde $E(u,p;f)$ es la formulación integral de (2.15)-(2.14), es decir

$$\begin{aligned} E(u,p;f) \quad = \quad & -\int\int_{Q_T} \Big\{ up_t + f(u)p_x \Big\} dxdt \\ & + \int_{\mathcal{T}} \Big\{ f(g_1(t))p(1,t) - f(g_0(t))p(0,t) \Big\} dt \\ & - \int_I u_0(x)p(x,0)dx, \quad \forall p \in C_0^1(Q_T), \end{aligned}$$

y U_{ad} es el conjunto admisible definido como

$$\mathcal{U}_{ad} = \{ f : \mathbb{R} \to \mathbb{R} \mid f \in C^2(\mathbb{R}),\ f \text{ satisface (2.11)} \}.$$

En la práctica el problema (2.19) es reducido a un problema de identificación de parámetros, en el cual se asume que la función de flujo es parametrizada por $\mathbf{e} = (e_1, \ldots, e_d) \in \mathbb{R}^d$, es decir $f(\cdot) = f(\cdot;\mathbf{e})$.

En consecuencia, para la identificación de parámetros se resuelve el siguiente problema de optimización de varias variables:

$$\begin{cases} \text{minimizar} & \mathcal{J}(\mathbf{e}), \quad \mathcal{J}(\mathbf{e}) := J\Big(u(\mathbf{e}), f(u(\mathbf{e});\mathbf{e})\Big), \\ \text{sujeto a} & E\Big(u(\mathbf{e}), p; f(u(\mathbf{e});\mathbf{e}))\Big) = 0, \\ & \mathbf{e} \in D := \{ \mathbf{x} \in \mathbb{R}^d : f(\cdot,\mathbf{x}) \in U_{ad} \}. \end{cases} \tag{2.20}$$

El problema (2.20) es útil para aproximar las leyes constitutivas que intervienen en el modelo del proceso de sedimentación de tipo sólido-fluido, el tratamiento matemático se traduce en un problema de identificación de parámetros que es una situación particular de un problema inverso que puede ser interpretado como un problema de optimización con restricciones. Para la identificación, se supone que los parámetros de las leyes constitutivas, son agrupados en el denominado vector de parámetros denotado por $\mathbf{e} \in \mathbb{R}^d$, el cual depende de las propiedades del material considerado.

De manera genérica el modelo de la sedimentación de partículas sólidas inmersas en un fluido está constituido por dos grandes componentes: la ecuación diferencial parcial y las leyes constitutivas, mostrado en la Figura 2.3. El modelo puede ser resuelto numéricamente. Esta solución numérica requiere del conocimiento del vector de parámetros $\mathbf{e} \in \mathbb{R}^d$, de la concentración inicial $\phi(x,0)$ y de las condiciones de borde $\phi(0,t)$ y $\phi(L,t)$ [5, 13] . Al ingresar estos datos se obtiene un perfil de simulación de la concentración en un tiempo dado, $T > 0$, este perfil es denotado por $\phi^{\exp(x)}$. En este contexto el problema de calibración de parámetros consiste en minimizar la función objetivo que compara el perfil de simulación $\phi(x,T;e)$ con un perfil experimental $\phi^{\exp(x)}$.

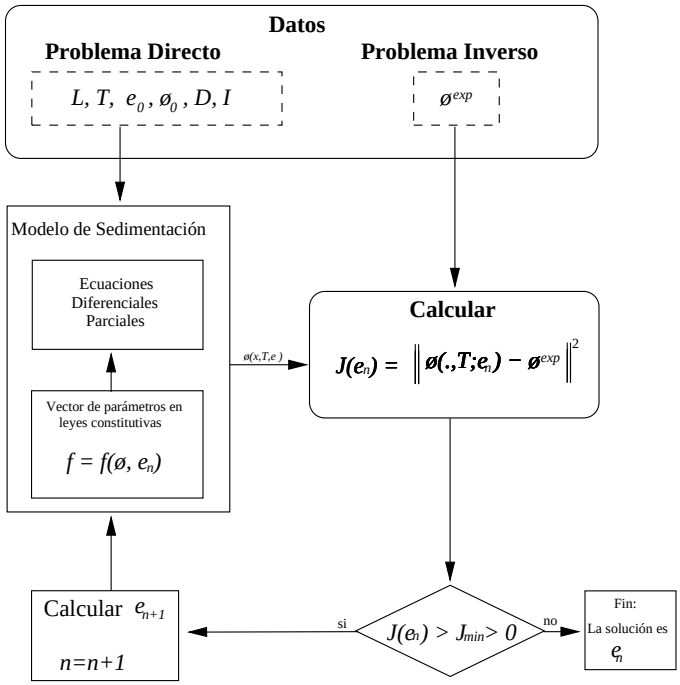


Figura 2.3: Representación esquemática del problema de identificación de parámetros

Por medio del cálculo formal del gradiente, explicado en detalle en [32] se puede resolver el problema de optimización, asociado al problema inverso. Sin embargo, este procedimiento presenta algunos inconvenientes, entre ellos la no obtención del mínimo global de la función objetivo cuando las restricciones representan una región suficientemente amplia tal que existen

varios mínimos locales. Esto es, dado que el método del gradiente consiste en un proceso iterativo que realiza el cálculo de las derivadas de la función objetivo con respecto a los parámetros y se detiene cuando la norma del gradiente es nula (o de una tolerancia fijada). El criterio de detención puede suceder en cualquier mínimo local, tal como se muestra en al Figura 2.4.

La intervención de la propuesta en la presente tesis, supera las desventajas del método del gradiente utilizando las estrategias de búsqueda global de los algoritmos genéticos.

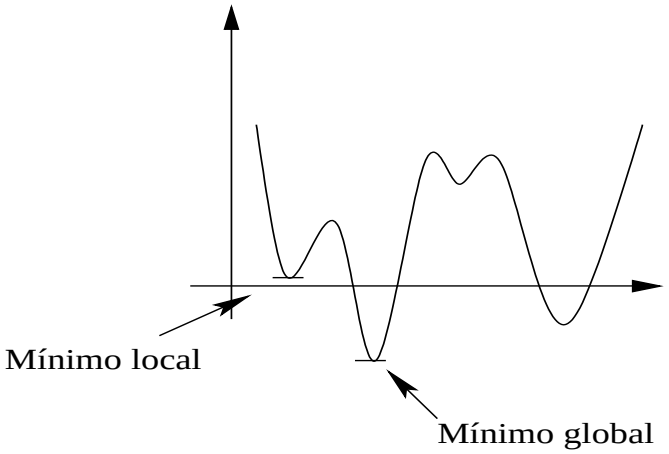


Figura 2.4: Ilustración de una función con varios mínimos

Capítulo 3

Computación Evolutiva, Optimización y Algoritmos Genéticos

En este capítulo se aborda los aspectos generales relacionados al desarrollo de la Computación Evolutiva, sus alcances y algunos, de los muchos trabajos, en los cuales ha estado presente. También se presentan los aspectos del problema de optimización y los alcances de los algoritmos genéticos.

3.1. Aspectos Generales de Computación Evolutiva

La capacidad del ser humano en buscar soluciones a los problemas que se ve enfrentado, ha despertado el interés de investigar e indagar en los procesos naturales del medio ambiente y de la evolución. Es así, como la teoría Neo-Darwinismo influyó en la motivación de muchos científicos e investigadores para crear estrategias de programación y métodos de algoritmos que le permitieran dar soluciones a problemas que de otra forma resultarían imposibles. De esta manera nace la computación evolutiva en sus diferentes variantes, la que ha resultado ser muy exitosa en dar solución a una amplia gama de problemas del mundo real.

Para hablar de computación evolutiva se debe retroceder a la década de los treinta, cuando Walter D. Cannon plantea en su libro The Winsdom of de Body (la sabiduría del cuerpo) que la evolución natural planteada por Darwin fue vista como un proceso de aprendizaje, algo similar al aprendizaje por ensayo y error que suele estar presente en los humanos. Alan Turing también

señala una conexión entre la evolución y el aprendizaje de máquina en su artículo Computing Machinery and Intelligense (maquinaria inteligencia) [49].

El término “computación evolutiva” también llamado “algoritmos evolutivos” contiene una serie de técnicas inspiradas en los principios de la teoría Neo-Darwineana de la evolución natural. Esta teoría establece que la vida en nuestro planeta puede ser explicada a través de varios procesos estadísticos que actúan sobre y dentro de las poblaciones y especies [27], la reproducción, la mutación, la competencia y la selección.

En términos generales, para simular procesos evolutivos en un computador se requiere:

- Codificar las estructuras que se replicarán, o sea, la estructura de datos utilizada para almacenar a un “individuo”.
- Operaciones que afecten al “individuo”. Típicamente se usa cruce y mutación.
- Una función de aptitud que indique que tan buena es una solución con respecto a las otras.
- Un mecanismo de selección que implemente el principio de “supervivencia del más apto” (teoría de Darwin).

El tema aquí abordado ha proliferado considerablemente y juntamente con eso la gran cantidad de información, casos de uso, aplicaciones y variaciones en sus técnicas de implementación. Por tal motivo, este trabajo pretende ser un aporte más en este sentido.

3.2. Computación Evolutiva en Acción

Continuando con las bases históricas del comienzo de la computación y algoritmos evolutivos, se sabe que entre 1950 y principio de 1960 varios científicos de manera independiente comenzaron a estudiar los sistemas evolutivos teniendo la intuición que la idea de la evolución se podría utilizar para crear herramientas que ayudara a resolver problemas de optimización.

Alexander Fraser publicó una serie de trabajos acerca de la evolución de sistemas biológicos utilizando un computador digital [38], estos estudios dieron origen a inspiraciones que más tarde se materializarían en algoritmos genéticos [28].

En general, muchos científicos interesados en el tema eran biólogos evolutivos que buscaban explícitamente realizar modelos de aspectos de la evolución natural, pero ninguno de ellos aún se les ocurría pensar que ésta estrategia podría aplicarse en forma más general a otros problemas.

Otro desarrollo importante, surgió en 1965 cuando Ingo Rechemberg de la Universidad Técnica de Berlin introdujo una técnica que se llamó estrategia evolutiva. L. J. Fogel et al. [20], introdujeron en América una técnica llamada programación evolutiva. En esta técnica las soluciones candidatas para los problemas se representaba como máquinas de estado finito y al igual que la estrategia evolutiva de Rechemberg su algoritmo funcionaba mutando aleatoriamente una de estás máquinas simuladas y conservando la mejor de las dos [24, 38].

En 1967 Beagley diseñó un algoritmo genético para buscar parámetros en funciones de evaluación de juegos. Pero otro científico, es considerado el creador de los algoritmos genéticos: John Holland, este ya en 1962 realizaría trabajos sobre sistemas adaptativos y estableció las bases para desarrollos posteriores. Fue el primero en explicitar propuestas de cruzamiento y otros operadores de combinación. Sin embargo, el trabajo fundamental en el campo de los algoritmos genéticos apareció en el año 1965 con la publicación del libro "Adaptation In Natural and Artificial System" (adaptación en sistemas naturales y artificiales) [28].

Basado en artículos anteriores del propio Holland y otros colegas de la Universidad de Michigan este libro fue el primero en presentar sistemática y rigurosamente el concepto de sistemas digitales adaptativos utilizando la mutación, la selección y el cruzamiento, simulando el proceso de la evolución biológica como estrategias para resolver problemas. Al poco tiempo de la publicación del libro, la tesis de Kenneth D. Jhong mostró el potencial de los algoritmos genéticos, demostrando que se podían desenvolver bien en una gran variedad de funciones de prueba, incluyendo paisajes de búsqueda ruidosos o mejor dicho funciones con multiples máximos y mínimos, discontinuos y multimodales [24].

Los trabajos mencionados anteriormente, sumados a otros, establecieron un gran interés en el desarrollo de la computación evolutiva. En particular los algoritmos genéticos, ya a mediados de los ochentas, se aplicaban a muchas áreas, desde problemas matemáticos abstractos (ej problema de la mochila, coloración de grafos, etc.) hasta asuntos tangibles en ingeniería, como el control de flujo en líneas de ensamblaje, reconocimiento y clasificación de patrones, y problemas de optimización estructural [24].

Hoy en día, los algoritmos genéticos han migrado desde el sector de la investigación hacia el sector comercial, dada la importancia en el crecimiento exponencial en la capacidad de cómputos y por otro lado el desarrollo de Internet. La computación evolutiva es aun un campo floreciente, pero también una realidad como método que permite resolver cientos de problemas.

Algunos ejemplos de las áreas de aplicaciones son:

- Predicción en la bolsa y planificación en la cartera de valores.
- Ingeniería aeroespacial.
- Diseño de microchips.
- Bioquímica y biología molecular.
- Diseño de horarios en aeropuertos.
- Líneas de montaje.

En particular, la predicción en la bolsa y planificación en la cartera de valores así como temas relacionados a inversiones bursátiles, podemos encontrar varios trabajos que desarrollan estas líneas de investigación. Uno de ellos se encuentra en [4].

En términos generales se conoce de la existencia de distintos tipos de algoritmos evolutivos, por razones sobre todo históricas, suelen hablarse de tres tipos o paradigmas principales: Programación evolutiva, Estrategia evolutiva y Algoritmo genético.

3.3. Acerca de la Optimización

En este punto se presentan algunas ideas básicas sobre búsqueda y optimización, así como problemas sobre los cuales se aplican diversos algoritmos evolutivos, en especial algoritmos genéticos. Una definición sucinta de optimización es aquella que la muestra como la “ciencia” encargada de determinar las mejores soluciones a problemas matemáticos que a menudo modelan una realidad física. Problemas complejos de optimización multidimensionales, no lineales y altamente multimodales pueden encontrarse en campos como ingeniería, economía, geofísica, etc.

La optimización persigue resolver un problema encontrando una solución óptima con un costo computacional razonable. El objetivo de esta, en términos amplios, es buscar respuestas a un tipo de problema obteniendo y seleccionando al "mejor" entre un conjunto de elementos o candidatos. En Flecher [21] y Nocedal [39] se detallan los aspectos básicos de la teoría de optimización no lineal.

La optimización, según lo ya indicado, está fuertemente ligada a problemas de búsqueda. En informática se habla de búsqueda cuando hay que encontrar información en un conjunto de datos almacenados, siguiendo un determinado criterio. Sin embargo, existe otro tipo de algoritmo de búsqueda aquellos que, dando el espacio posible a todos esos problemas, son capaces de encontrar la mejor solución o la única. Un ejemplo de este tipo de problema se encuentra en los rompecabezas y juegos, que se suelen abordar en inteligencia artificial; otro ejemplo es el problema de las ocho reinas, las torres de Hanoi, etc. Este tipo de problemas se suelen abordar con algoritmos clásicos, algoritmos recursivos o de tipo voraz.

Existe otro tipo de problema, mucho más complicado, los llamados NP-completos (cuya complejidad crece con el tamaño del problema en forma exponencial), estos no pueden ser abordados con los algoritmos de búsqueda clásicos. Algunos de estos problemas son por ejemplo: 8-puzzle y problema del viajante, entre otros. En muchos casos la búsqueda se encuentra guiada por una función que señala lo buena que es esta solución, o el costo de esta o lo cerca que esta de la solución final, si es que se conoce; este problema se convierte entonces en un problema de optimización.

Otra clasificación que aparece en la literatura en relación a algoritmo que satisfagan los problemas señalados anteriormente, son aquellos denominados metaheurísticos. Término introducido por Glover [23]. Esta clase de algoritmo incluye: colonias de hormigas, búsqueda local iterada, búsqueda tabú, enfriamiento simulado y algoritmos evolutivos.

Se destacan de la metaheurística las siguientes propiedades que caracterizan este tipo de método:

- Estrategias o plantillas generales que guían el proceso de búsqueda.
- El objetivo es una exploración eficiente del espacio de búsqueda, para encontrar soluciones óptimas.

- Pueden incorporar mecanismos para evitar regiones no comprometedoras de espacios de búsqueda.
- Hacen uso de conocimiento del problema que se trata de resolver en forma de heurísticos específicos, que son controlados a más alto nivel.

Resumiendo, una metaheurística es una estrategia de alto nivel que usa diferentes métodos para explorar el espacio de búsqueda de gran tamaño. El problema de la optimización es tratado en un área de la matemática llamado investigación operacional. Aunque prácticamente, todas las áreas de las matemáticas y áreas de las ciencias necesitan tratar problemas de optimización.

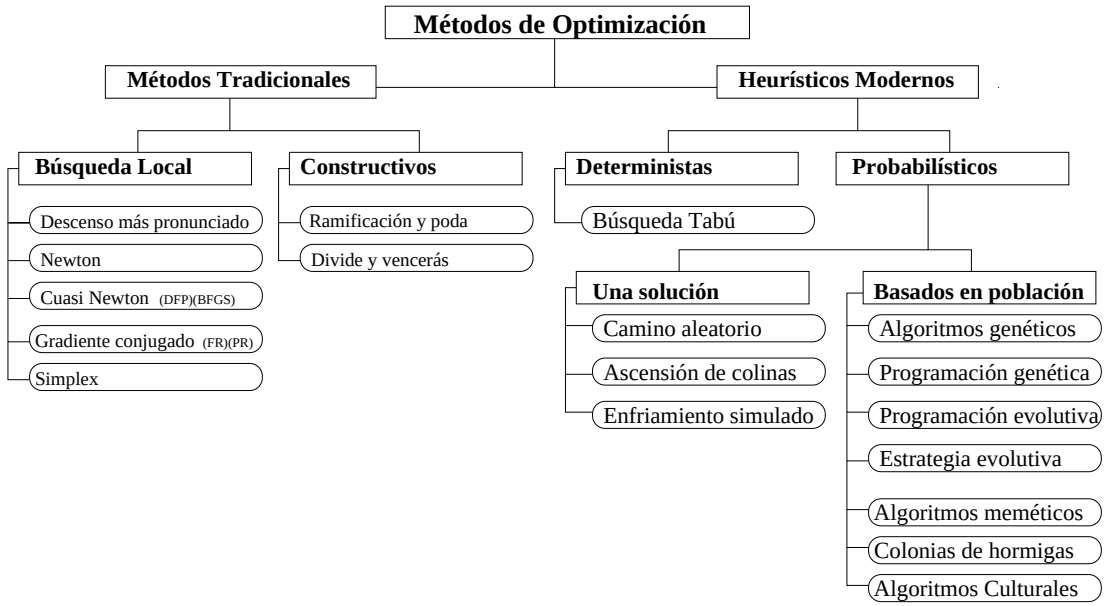


Figura 3.1: Clasificación de los métodos de optimización más relevantes presentados en [41].

Un problema de optimización generalmente se compone de tres elementos claves, los cuales son: la función costo $J : \mathbb{R}^n \rightarrow \mathbb{R}$, las variables de decisión $x \in \mathbb{R}^n$ y las restricciones $\Omega \subset \mathbb{R}^n$ y matemáticamente, de forma general el problema de minimización se formula de la siguiente manera:

$$\text{Encontrar } x_1 \in \Omega : J(x_1) \leq J(x),\ \forall x \in \Omega,\ j : \mathbb{R}^n \rightarrow \mathbb{R} \tag{3.1}$$

El problema (3.1) para el caso de un problema de maximización se remplaza la desigualdad por $J(x_1) \geq J(x)$. Siguiendo las ideas de Pérez [41], los métodos para resolver, analítica y computacionalmente, un problema del tipo (3.1), se pueden clasificar en métodos de búsqueda local y global. Esta clasificación, particular e ilustrativa detallada en la Figura 3.1 muestran, por un lado, los llamados métodos tradicionales en donde encontramos las técnicas de búsqueda local basados en derivadas, incluyendo el método símplex y los métodos constructivos que básicamente trabajan sobre soluciones parciales del problema aislando el espacio de búsqueda [21]. Alternativamente están los denominados métodos Heurísticos modernos, los cuales en la década pasada y en los últimos años han sido el foco de atención en diversos tipos de investigaciones, adquiriendo una notable aceptación en diferentes campos de la ingeniería y disciplinas afines [24].

Los métodos locales obtienen la mejor solución posible en las inmediaciones del punto de partida inicial. Se les atribuye una fuerte dependencia del punto de arranque del algoritmo. En el extremo opuesto las técnicas de optimización global muestran una gran independencia de la naturaleza del espacio de soluciones, a diferencia de las técnicas de búsqueda local son capaces de realizar una búsqueda con múltiples mínimos o máximos locales y alcanzar una solución global de los problemas [23].

Los métodos Heurísticos modernos tienen en común el hecho de utilizar una población o conjunto de soluciones potenciales y someterlos a un proceso iterativo con diversas estrategias en función del tipo de algoritmo. En la literatura existe una clasificación de las estrategias evolutivas las cuales están definidas en cuatro grandes áreas, a éstas se les conoce como cómputo evolutivo [34]. Una de estas áreas es la de los algoritmos genéticos y corresponde al método que ha recibido mayor atención en los últimos años.

3.4. Algoritmos Genéticos

En esta sección se darán a conocer las propiedades generales de un algoritmo genético. Esta metodología, según lo ya señalado, fue introducida por Holland y colaboradores [24, 28] y es de tipo evolutivo, por tanto es inspirada en la teoría propuesta por Darwin de la evolución de las especies, según la cual se establece como ley fundamental el hecho que los individuos más aptos de una población son los que sobreviven al adaptarse mejor a los cambios de su entorno.

La razón del creciente interés por los algoritmos genéticos es que estos son un método global y robusto de búsqueda [24]. Utilizan operadores probabilísticos, en vez de los típicos operadores determinísticos de las técnicas tradicionales. Otra de las ventajas, es que los algoritmos genéticos destacan por su habilidad para manipular muchos parámetros simultáneamente e inclusive en la presencia de problemas con múltiples objetivos. Holland los definió como un mecanismo efectivo para resolver problemas de búsqueda y optimización a nivel computacional.

Dicho lo anterior podemos especificar que, un algoritmo genético genera una población inicial de individuos y luego aplica operadores genéticos para cada nueva reproducción. Cada individuo es codificado mediante una cadena de valores *"cromosoma"* y este representa una posible solución. Esta solución que representa el individuo es evaluada con respecto a una función objetivo dada para el problema. Mediante alguna técnica o método de selección permanecen aquellos individuos o soluciones que tienen un mayor valor de adaptación, estos tendrán mayores probabilidades de reproducirse intercambiando, con otros, parte de su información *"genética"*, mediante un proceso denominado *cruce*. También, se emplea el proceso de *"mutación"* mediante el cual se alteran algunos *genes* en el individuo. Producto de estos procesos se generan nuevos individuos que reemplazarán a algunos miembros de la actual población. Este ciclo constante de evaluación, selección, reproducción (cruce y mutación) y remplazo individuos a la población se hacer repetir hasta que se encuentra una solución satisfactoria o hasta que se alcance algún criterio de término del algoritmo.

Se debe enfatizar que los algoritmos genéticos son muy efectivos sobre todo para aquellos problemas con un gran espacio de búsqueda y para los que no existe una técnica específica para resolverlo. Junto con ello han demostrado su efectividad al "escapar" de los óptimos locales y descubrir el óptimo global, incluso en "paisajes adaptativos muy escabrosos" y complejos, esto es, funciones con muchos máximos y mínimos. Para una detallada información, en este sentido, se puede consultar [24] y Sivanamdam [45].

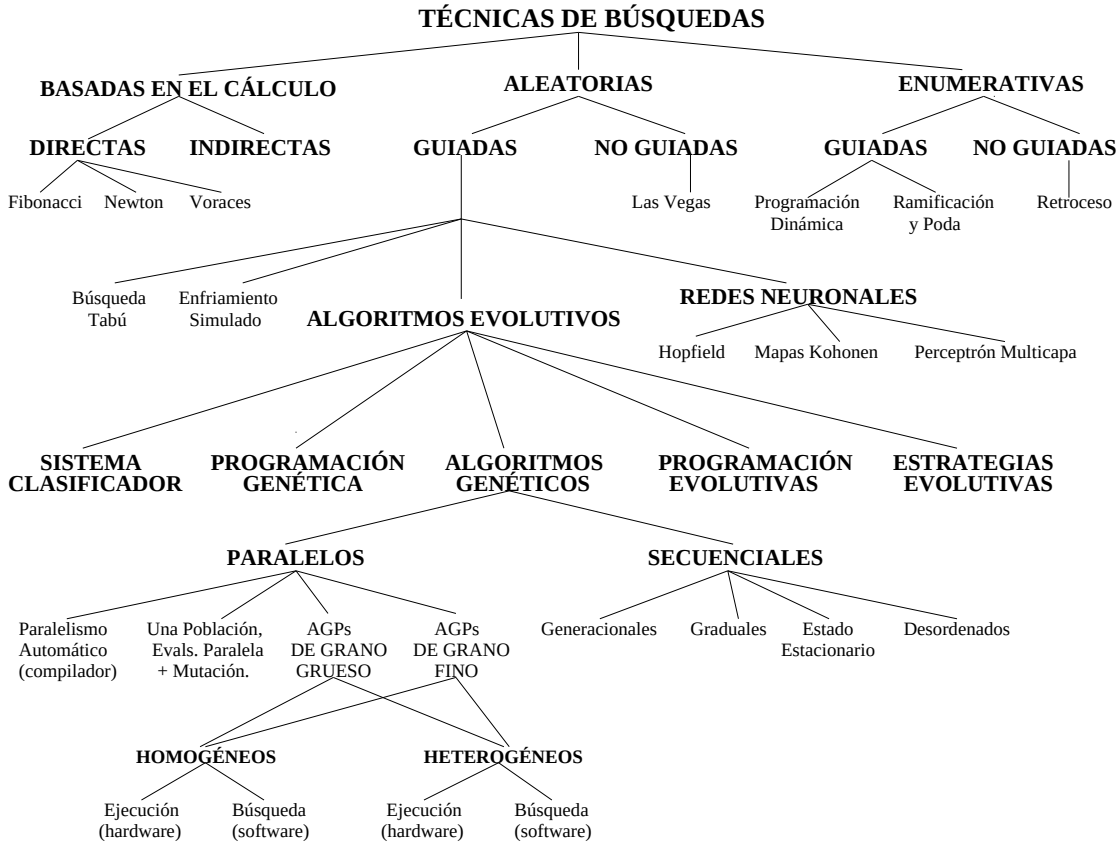


Figura 3.2: Clasificación de las diversas Técnicas de Búsquedas.

En la subsección anterior, específicamente en la Figura 3.1 se muestra una clasificación de los métodos de optimización más relevantes y conocidos, ahora, en la Figura 3.2 se muestra otra clasificación orientada a reconocer las diversas técnicas de búsquedas y de forma más detallada se muestran algunas de las diversas variantes de los algoritmos genéticos [45]. En este trabajo de tesis se construyó un algoritmo genético secuencial, generacional. Si bien hay otras variantes de algoritmos genéticos como: los algoritmos genéticos celulares y los algoritmos genéticos distribuidos.

3.4.1. Características Básicas

Dentro de las características relevantes de de un algoritmo genético están:

- No son deterministas.
- Optan más por reglas de probabilidad.
- Trabajan sobre una población de individuos representados por uno o más cromosomas (conjunto de genes) formados con material genético (atributos), esto permite diferenciar a cada individuo del resto.
- A partir de un conjunto inicial y aleatorio de posibilidades y soluciones (individuos) para un problema ya acotado, se realizan operaciones de cruce y mutación sobre el contenido de algunas soluciones (individuos seleccionados previamente según sus atributos), con el objetivo de incorporar posibles mejoras en cada generación al reemplazar proporcionalmente los individuos de la población.

3.4.2. Esquema o Secuencia Básica de un Algoritmo Genético

Los pasos de un algoritmo genético básico son:

1. **Inicio:** Generar (aleatoriamente) una población inicial de individuos.
2. **Evaluar:** Calcular la aptitud *fitness* de cada individuo de la actual población P_i.
3. **Condición de término:** Verificar si se cumplen las condiciones para detener la ejecución del algoritmo. De no cumplir las condiciones ir al siguiente punto, de lo contrario se termina la ejecución y se muestran los resultados.
4. **Generar siguiente población:** Generar la siguiente población P_{i+1} de acuerdo a los siguientes paso:

   a) (Selección) Realizar un proceso de selección de individuos de la actual población P_i, con algún método especifico de selección, de acuerdo y con directa relación a la aptitud y de una función de probabilidad.

b) (Cruce) Aplicar la operación de cruce entre individuos, según alguna regla especifica, para generar nuevos individuos.

c) (Mutación) Realizar el proceso de mutación a los individuos según la tasa y probabilidad definida.

d) (Añadir) Agregar los nuevos individuos a población P_{i+1}.

e) (Iterar) Volver al paso *a*) hasta completar el tamaño de la población P_{i+1}.

5. **Remplazar:** Remplazar la población P_i por P_{i+1} y volver al paso 2.

3.4.3. Un Pseudocódigo del Algoritmo Genético Básico

Un pseudocódigo básico se presenta en la tabla de algoritmo 3.1

```
1: P = GeneraPoblaciónInicial();
2: evaluar(P);
3: while condicion ≠ {⟨condicionParada()⟩} do
4:    P'=selecciondePadres(P);
5:    P'=OperadoresDeVariación(P');
6:    evaluar(P');
7:    P = GeneraNuevaPoblación(P,P');
8: end while
9: return La mejor solución encontrada
```

Alg. 3.1: Pseudocódigo de un algoritmo genético básico

3.4.4. Componentes Básicos Necesarios Para el Algoritmo Genético

Para hacer uso de un algoritmo genético se deben tener claro, al menos, los siguientes componentes básicos.

1. Una representación de las soluciones potenciales del problema.

2. Una forma de crear una población inicial de posibles soluciones.

3. Una función de evaluación que clasifique las soluciones (en términos de su aptitud).

4. Operadores genéticos que alteren la composición de los individuos que se producirán para la siguiente generación.

5. Valores para los diferentes parámetros que utiliza los algoritmos genéticos.

    - Tamaño de la Población.
    - Probabilidad de cruza.
    - Probabilidad de mutación.
    - Número máximo de generaciones

3.4.5. Método de Representación de un Algoritmo Genético

Un primer enfoque, muy común, es codificar las soluciones como cadenas binarias (0 y 1), donde el dígito de cada posición representa cada aspecto de la posición. Otro enfoque consiste en codificar las soluciones como cadenas de enteros o números reales. Un tercer enfoque consiste en representar a los individuos como cadenas de letras, donde cada letra representa un aspecto específico de las soluciones. Hiroaki Kitano, empleó esta técnica basado en "codificación gramatical". A un algoritmo genético se le encargó la tarea de evolucionar un conjunto de reglas llamadas gramática libres de contexto, que a su vez se utilizaba para generar redes neuronales para una una variedad de problemas [38].

3.4.6. Método de Selección de un Algoritmo Genético

Un algoritmo genético puede utilizar diversas técnicas para seleccionar los individuos que deben copiarse para la siguiente generación, algunos pueden ser utilizados en combinación con otro, otros son exclusivos, algunos de estos métodos son:

- Selección escalada: Método muy útil cuando todos los individuos tengan una aptitud relativamente alta. Al incrementarse la aptitud media de la población, la fuerza de la presión selectiva también aumenta y la función de aptitud se hace más discriminadora.
- Selección torneo: Se eligen subgrupos de individuos de la población y los miembros de cada subgrupo compiten entre ellos.

- Selección generacional: La descendencia de los individuos seleccionados en cada generación se convierte en toda la siguiente generación. No se conservan individuos entre las generaciones

- Selección elitista: La idea principal es que se debe garantizar la selección de los miembros más aptos de cada generación.

- Selección proporcionar a la aptitud: Aquí los individuos más aptos tienen más probabilidad de ser seleccionados, pero no la certeza.

- Selección por rueda de ruleta: La probabilidad de que un individuo sea seleccionado es proporcional a la diferencia entre su aptitud y la de sus competidores. Esto puede representarse como un juego de ruleta - cada individuo obtiene una sección de la ruleta, pero los más aptos obtienen secciones mayores que las de los menos aptos.

Existen otros métodos no mencionados aquí y muchas variaciones de estos. Según el problema dado o la convergencia obtenida se puede optar por uno y otro método.

3.4.7. Métodos de Cambio - Cruza y Mutación en un Algoritmo Genético

Una vez seleccionados los individuos aptos, estos deben ser "alterados" aleatoriamente con la finalidad de mejorar su aptitud, comúnmente se utilizan dos prácticas.

La primera es el cruzamiento y corresponde al proceso que se realiza entre dos individuos, el cual permite intercambien segmentos de sus códigos, produciendo una "descendencia artificial". Se pretende simular el proceso análogo a la recombinación que se da en los cromosomas durante la reproducción sexual. Se puede dar un solo punto de cruce o más de un punto.

La segunda práctica corresponde al proceso llamado mutación, éste genera pequeñas alteraciones concretas del código de un individuo. Intercambia un gen por algún valor distinto al guardado.

3.4.8. Algunas Aplicaciones de Algoritmos Genéticos

Se ha señalado que los algoritmos genéticos se han aplicado a diversos problemas, modelos de ingeniería y de ciencia en general, cabe destacar principalmente los problemas de

optimizaciones, no en vano fueron la fuente de inspiración para los creadores de estos algoritmos. Otras aplicaciones de estos algoritmos son en la "programación automática", para el diseño de estructuras computacionales, tales como el autómata celular y las redes de clasificación. El aprendizaje de máquina también se ha visto beneficiado en temas como predicción del tiempo, estructura de una proteína, sistemas particulares de aprendizaje, etc. Otras áreas de aplicación son: ecología, genética de poblaciones, economía, sistema inmune, sistemas sociales.

Notables aplicaciones donde los algoritmos genéticos han obtenido resultados que ninguna otra solución hubiera dado tan asertivamente son:

- Acústica [43]: se utilizaron algoritmos genéticos para diseñar una sala de concierto maximizando la calidad del sonido para la audiencia, para el director y para los músicos en el escenario.

- Fogel y Fogel [57] describe la utilización de estos algoritmos para adiestrar redes neuronales para distinguir reflexiones sonoras desde distintos tipos de objetos: montañas submarinas, peces y plantas, ruidos aleatorios de fondo.

- Ingeniería Aeroespacial: utilización de un algoritmo genético de múltiple objetivo para diseñar la forma del ala de un avión supersónico [40].

- Astronomía y astrofísica [35] - Ingeniería eléctrica. - Juegos. - Geofísica. - Ingeniería de materiales.

- Ejército y cumplimiento de la ley: utilizaron algoritmos genéticos para evolucionar planes tácticos para batallas militares.

- Robótica. - Reconocimiento de patrones. - Ingeniería de sistemas - Etc.

Desde la introducción de los algoritmos genéticos, por Holland en 1975, se han publicado varias reinterpretaciones y adaptaciones, una de las referencias más completas en este sentido son los trabajos desarrollados por Z. Michalewicz [51]. También se pueden ver ejemplos y consultar [33]. Además, sobre los aspectos más detallados de implementación se pueden ver en Haupt y Haupt [25] y Sivandam [45].

3.5. Problemas Inversos Resueltos Mediante Algoritmos Genéticos

En este punto se presenta, de forma más explícita, el concepto de problema inverso como así también algunos trabajos directamente relacionados a la resolución de problemas inversos mediante algoritmos genéticos. Esto es, dado que un principal objetivo de esta tesis es aplicar un algoritmo genético a la función que resuelve el problema de identificación de parámetros en el modelo de sedimentación. El cual, como ya hemos visto, corresponde a la resolución de un problema inverso.

Según lo ya tratado, un problema inverso puede definirse como aquel en donde los valores o parámetros de algún modelo, deben ser obtenidos por medio de datos observados [10], esto es en contraste al problema directo en donde se predicen los datos a partir de parámetros y un modelo. Según [19] el problema inverso es el conjunto de métodos usados para extraer información útil de nuestro entorno a partir de medidas físicas o datos.

También hemos dicho que mediante la técnica del gradiente se han realizado trabajos que abordan el problema inverso, algunos de ellos están relacionados al modelo de la cromatografía y otros similares.

Ahora, en la literatura se encuentran múltiples trabajos que hacen uso de estrategias evolutivas y algoritmos genéticos para realizar optimización a ajustes de parámetros, de modelos específicos, producto de un problema inverso. Este punto es de especial interés pues es justamente un problema de identificación de parámetros el que se aborda en la presente tesis. Se mostrarán algunos casos.

Un interesante trabajo, en la línea de geometría computacional y bases de datos, está presentado en [47] donde se estudia el problema inverso el cual se refiere a la descomposición de polígonos en sumas de Minkowski. Dado un polígono S de n lados, se desea encontrar dos polígonos P y Q de k y k' lados respectivamente, tales que S sea la suma de Minkowski de P y Q. Dada la complejidad del algoritmo, de orden exponencial, los autores proponen una solución algorítmica utilizando algoritmos genéticos. En otro ámbito de aplicación, concretamente en los sondeos de eléctricos verticales, presentado en [19] se persigue obtener, a partir de observaciones en superficie de resistividades aparentes, el vector de parámetros m que describe el modelo estratificado de terreno. En su formulación tradicional es, pues, un problema de identificación de parámetros, un problema inverso. La propuesta presentada es realizar la optimización utilizando

algoritmos genéticos. También, en la construcción de chimeneas industriales se ha investigado acerca de la optimización en su diseño y construcción teniendo en cuenta varios factores de calidad en cuanto a su nivel de tiro, niveles de emisión de contaminantes y otros. Este tema ha sido abordado en [31] donde se plantea el problema como un problema inverso; se tiene los resultados deseables de un proceso y se deben encontrar las condiciones de operación y construcción que lleven a dichos resultados. Para resolver el problema inverso los autores proponen una metodología, para el diseño de chimeneas, que consigue minimizar el impacto ambiental, debido a la emisión de gases contaminantes, a la vez que se minimizan los costos de construcción y explotación. Esta metodología utiliza los algoritmos genéticos para resolver el problema de optimización.

Una optimización a los ajustes de parámetros fue realizada en [9] donde se presenta el modelo de vuelo de un helicóptero. Estos ajustes al modelo fueron realizados por algoritmos genéticos.

En las fases de planificación de un proyecto, en Ingeniería de Software, los algoritmos genéticos ttambién se han implementado con éxito [11]. La idea central es encontrar automáticamente ”buenas”soluciones a la gestión de un proyecto: asignación de tareas, tiempos, recursos, etc. Problema inverso.

La literatura sobre optimización utilizando algoritmos genéticos aborda una serie de áreas, técnicas y aplicaciones. Cada día aparecen nuevos problemas que se abordan bajo estas estrategias evolutivas dado el ”éxito”que estas ofrecen y como alternativa a soluciones complejas y de mucho costo computacional. De la literatura vista, no se encuentran trabajos que hayan abordado el objetivo de este proyecto. Si bien hay trabajos que utilizan la estrategia del gradiente, no hay trabajos que estudien el problema por medio de estrategias evolutivas.

Capítulo 4

Investigación y Resultados

En esta sección se analizan y presentan los elementos necesarios para abordar la solución al problema ya planteado. Recordando parte de las variables, etapas y reglas a considerar para la construcción de la solución algorítmica. También se presenta el algoritmo de identificación propuesto, los diagramas y pseudocódigo involucrado. Luego se describen un par de experimentos en donde fue probado el algoritmo.

4.1. Elementos a considerar

Recordando lo ya investigado y deducido en el Capítulo 2 tenemos que, de manera genérica el modelo de la sedimentación de partículas sólidas inmersas en un fluido está constituido por dos componentes: por un lado está la ecuación diferencial parcial y por otro las leyes constitutivas, mostrado en la Figura 2.3. El modelo puede ser resuelto numéricamente. Esta solución numérica requiere del conocimiento del vector de parámetros $\mathbf{e} \in \mathbb{R}^d$, de la concentración inicial $\phi(x,0)$ y de las condiciones de borde $\phi(0,t)$ y $\phi(L,t)$ [5, 13] . Al ingresar estos datos se obtiene un perfil de simulación de la concentración en un tiempo dado, $T > 0$, este perfil es denotado por $\phi^{\exp(x)}$. En este contexto el problema de calibración de parámetros consiste en minimizar la función objetivo que compara el perfil de simulación $\phi(x,T;e)$ con un perfil experimental $\phi^{\exp(x)}$.

Sin embargo; el vector de parámetros $\mathbf{e} \in \mathbb{R}^d$ es desconocido a priori, no puede ser obtenido por mediciones experimentales directas y tampoco puede ser deducido directamente por el modelamiento matemático. Luego, para su determinación se necesita formular y resolver

un problema de calibración del modelo. Existen al menos dos maneras de estimar el vector de parámetros $\mathbf{e} \in \mathbb{R}^d$. Esta, por un lado, la calibración empírica introducida por Becker [1] y recientemente retomada en [12], en la cual se realizan varios experimentos de separación para un mismo material, se mide la velocidad de sedimentación y mediante un ajuste de los datos concentración-velocidad se determina el vector $\mathbf{e}$. En segundo lugar, se encuentra la identificación numérica vía la solución de un problema inverso y que fue introducida en [32] en el contexto de la separación de compuestos por cromatografía y aplicada exitosamente al caso de la identificación en teoría de sedimentación en [15] y también en [2, 3, 6, 7]. En este trabajo de tesis se aplicará el segundo enfoque.

Es de saber también que los métodos numéricos utilizados para resolver numéricamente los problemas de optimización son de tipo iterativo. Esquemáticamente y de manera genérica, contextualizando estos métodos para obtener la solución numérica del problema de identificación de parámetros en el modelo (2.15)-(2.14) se obtiene la Figura 2.3. En este caso se distinguen tres etapas:

(a) Se considera como variables de entrada la altura de la mezcla L, el tiempo en el cual se obtiene el perfil experimental T, un vector inicial de parámetros $\mathbf{e}_0 \in \mathbb{R}^d$, las condiciones de frontera I y D, la condición inicial ϕ_0 y el perfil experimental ϕ^{exp};

(b) Se resuelve numéricamente el problema directo (2.15)-(2.14), para obtener el perfil $\phi(\cdot, T; \mathbf{e}_0)$ que permite evaluar la función costo J en $\mathbf{e}_0$, la cual naturalmente se define como comparación de $\phi(\cdot, T; \mathbf{e}_0)$ con el perfil experimental en el sentido de la norma (o sentido integral);

(c) Si los perfiles no coinciden, es decir el valor de la función costo es superior a una tolerancia previamente fijada, se reinicia en el punto (b) con un nuevo vector de parámetros, el cual es actualizado en función de la estrategia utilizada para la optimización numérica. Los pasos (b)-(c) se repiten iterativamente hasta que la tolerancia fijada para la función costo es alcanzada.

La metodología de optimización numérica utilizada para actualizar el vector de parámetros ha sido el gradiente. Es necesario notar que la no linealidad de la función de flujo implica que el gradiente de la función de costo con respecto a los parámetros se debe calcular aplicando

estrategias desarrolladas por la teoría de control óptimo [22]. Las estrategias más comunes son: la ecuación adjunta y la ecuación de sensitividad. El cálculo formal del gradiente de la función costo vía la definición de un estado adjunto fue introducido en [15] y también [2, 3] siguiendo las ideas desarrolladas en [32]. El cálculo del gradiente vía la definición de la ecuación de sensitividad fue desarrollado en [2, 6, 7]. En ambos casos (adjunto y sensitividad) se presentan algunos inconvenientes, entre ellos la no obtención del mínimo global de la función objetivo cuando las restricciones representan una región suficientemente amplia tal que existen varios mínimos locales. Esta desventaja es ampliamente conocida para los métodos de optimización numérica basados en el gradiente y en el caso de los trabajos [2, 3, 6, 7, 15] se evita esta dificultad asumiendo que el vector inicial $\mathbf{e}_0$ está en una vecindad del vector de parámetros que se desea identificar. Esta hipótesis es solamente cierta en situaciones donde se conocen propiedades físicas del material utilizado en la mezcla. Desafortunadamente, en el mayor de los casos estas propiedades no son conocidas y se tiene que resolver el problema de optimización aplicando estrategias de convergencia global.

4.2. Considerando el algoritmo

En lo concerniente a los algoritmos genéticos se tomaron en cuenta las ideas presentadas por Haupt y Haupt [25] y aplicadas recientemente para la identificación del flujo en leyes de conservación escalares en [16]. Ahora bien, la aplicación de algoritmos genéticos en la solución de problemas inversos es una teoría emergente y han sido utilizados exitosamente en una variedad de situaciones. Sin embargo, la aplicación al caso de identificación de flujo en leyes de conservación, de acuerdo al conocimiento e investigación del estado del arte, ha sido desarrollado independientemente [29, 16]. El trabajo presentado en [29] se centra en la identificación de parámetros en la ecuación de Saint-Venan. En tanto que el trabajo de [16] se centra en validar numéricamente un algoritmo genético propuesto para identificar un flujo fuertemente convexo. Los algoritmos propuestos en [29, 16] se basan en la definición de operadores continuos para realizar el proceso de selección natural, es decir sin recurrir a la codificación binaria de los genes, y su diferencia radica en que los operadores utilizados son algebraicamente diferentes, lo cual naturalmente los convierte en algoritmos matemáticamente distintos. Dicho lo anterior resulta de interés el proponer un algoritmo genético que aborde lo propuesto en esta tesis.

4.3. Algoritmo de Identificación propuesto

La metodología de identificación de parámetros propuesta y utilizada en el presente trabajo consta de dos etapas: construcción de la función costo discreta y construcción y aplicación del algoritmo genético.

4.3.1. Etapa1. Construcción de la función costo.

La aproximación numérica natural de (2.20) es

$$\min_{\mathbf{e}\in D} \mathcal{J}_\Delta(\mathbf{e}), \quad \mathcal{J}_\Delta(\mathbf{e}) := \frac{1}{2}\sum_{j=0}^{M} |\phi_j^N(\mathbf{e}) - \phi_j^{exp}|^2 \Delta x, \tag{4.1}$$

donde

$$\phi_j^{exp} = \frac{1}{\Delta x}\int_{K_j} \phi^{exp}(x)dx, \quad j = 0,\ldots,M, \tag{4.2}$$

y $\phi_j^N(\mathbf{e})$ para $j \in \{0,\ldots,M\}$ es la solución numérica del problema directo (2.15)-(2.14) en $t = T$, obtenida aplicando un Método de Volúmenes Finitos para leyes de conservación que de manera general se puede escribir como el siguiente esquema

$$\phi_0^{n+1} = \phi_0^n - \lambda\Big\{g(\phi_0^n,\phi_1^n) - f_b(\phi_0^n)\Big\}, \tag{4.3a}$$

$$\phi_j^{n+1} = \phi_j^n - \lambda\Big\{g(\phi_j^n,\phi_{j+1}^n) - g(\phi_{j-1}^n,\phi_j^n)\Big\}, \tag{4.3b}$$

$$\phi_M^{n+1} = \phi_M^n - \lambda\Big\{f_b(\phi_M^n) - g(\phi_{M-1}^n,\phi_M^n)\Big\}, \tag{4.3c}$$

definido para $n = 0,\ldots,N$ y $j = 0,\ldots,M$ y donde

$$\lambda = \Delta t/\Delta x, \quad \Delta t = T/N, \quad \Delta x = H/M,$$
$$\phi_j^0 = \frac{1}{\Delta x}\int_{K_j} \phi_0(x)dx, \quad K_j =]x_{j-1/2}, x_{j+1/2}[,$$
$$x_{j+1/2} = (x_j + x_{j+1})/2, \quad x_j = j\Delta x, \quad x_{-1/2} = x_0$$
$$x_{M+1/2} = x_M, \quad t_n = n\Delta t$$

y g es un flujo numérico monótono.

4.3.2. Etapa2. Construcción del Algoritmo Genético.

Para resolver el problema inverso (2.20) vía la aproximación de la función costo dada por (4.1), se propone un algoritmo genético continuo, consistente de dos grandes pasos: la selección de la población inicial y la selección natural. Para mayores detalles consultar [16, 29].

(a) **Población Inicial**. Se define la matriz que almacena la población inicial aleatoria:

$$\mathbb{E} = \left[\mathbf{e}_1|\mathbf{e}_2|\mathbf{e}_3\cdots|\mathbf{e}_n\right]^t,$$

donde cada fila $\mathbf{e}_j \in \Omega \subset \mathbb{R}^d$, $j = 1,\ldots,n$, con Ω el hipercubo definido como $\Omega := \prod_{i=1}^{d}[l_i,u_i] \subset D$, donde $l_i < u_i$ para $i = 1,\ldots,d$.

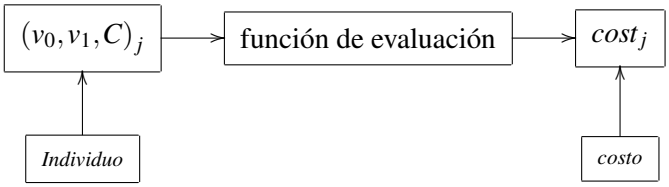
Cada elemento $\mathbf{e}_j$, de la matriz $\mathbb{E}$, representa a un individuo de la población generada y es un vector que contiene los parámetros (v_0, v_1, C) generados aleatoriamente, también llamados genes y que serán ajustados mediante el algoritmo genético hasta identificar aquellos que satisfagan la condición de admisibilidad.

La matriz $\mathbb{E}$ expandida es:

$$\mathbb{E} = \begin{pmatrix} (v_0)_{1x1} & (v_1)_{1x2} & (C)_{1x3} \\ (v_0)_{2x1} & (v_1)_{2x2} & (C)_{2x3} \\ (v_0)_{3x1} & (v_1)_{3x2} & (C)_{3x3} \\ \vdots & \vdots & \vdots \\ (v_0)_{nx1} & (v_1)_{nx2} & (C)_{nx3} \end{pmatrix}$$

(b) **Selección natural**.

(b1) **Costo de la población.** Se define el vector **cost** $\in \mathbb{R}^n$, mediante evaluación de la función costo en cada uno de individuos de la población $\mathbb{E}$, es decir **cost** $:= (\mathcal{J}_\Delta(\mathbf{e}_1),\ldots,\mathcal{J}_\Delta(\mathbf{e}_n)) \in \mathbb{R}^n$, y se define la matriz $\hat{\mathbb{E}} = [\mathbb{E}|\mathbf{cost}]$.



El vector **cost** queda así:

$$\mathbf{cost} = \begin{pmatrix} cost_1 \\ cost_2 \\ cost_3 \\ \vdots \\ cost_n \end{pmatrix}$$

Se conforma la matriz $\hat{\mathbb{E}}$:

$$\hat{\mathbb{E}} = \left(\begin{array}{ccc|c} (v_0)_{1x1} & (v_1)_{1x2} & (C)_{1x3} & cost_1 \\ (v_0)_{2x1} & (v_1)_{2x2} & (C)_{2x3} & cost_2 \\ (v_0)_{3x1} & (v_1)_{3x2} & (C)_{3x3} & cost_3 \\ \vdots & \vdots & \vdots & \vdots \\ (v_0)_{nx1} & (v_1)_{nx2} & (C)_{nx3} & cost_n \end{array}\right)$$

Si existe $\ell \in \{1,\ldots,n\}$ tal que $\mathcal{J}_\Delta(\mathbf{e}_\ell) \leq J_{\text{mín}}$, la solución de (4.1) es $\mathbf{e}_\ell$ y se terminan las iteraciones. La notación $J_{\text{mín}}$ es utilizada para la tolerancia de la evaluación de la función costo.

(b2) **Selección de los padres.** Se escogen los padres en tres pasos. Primeramente, se actualiza la matriz $\hat{\mathbb{E}}$ permutando las filas hasta que $\hat{\mathbb{E}}_{1,d+1} \leq \hat{\mathbb{E}}_{2,d+1} \leq \ldots \leq \hat{\mathbb{E}}_{n,d+1}$. Dicho de otra forma, se ordena la matriz en base al costo de menor a mayor, tal que; $cost_1 \leq cost_2 \leq cost_3 \leq \ldots cost_n$.

$$\hat{\mathbb{E}}Ordenada = \left(\begin{array}{ccc|c} (v_0)_{1x1} & (v_1)_{1x2} & (C)_{1x3} & cost_1 \\ (v_0)_{2x1} & (v_1)_{2x2} & (C)_{2x3} & cost_2 \\ (v_0)_{3x1} & (v_1)_{3x2} & (C)_{3x3} & cost_3 \\ \vdots & \vdots & \vdots & \vdots \\ (v_0)_{nx1} & (v_1)_{nx2} & (C)_{nx3} & cost_n \end{array}\right)$$

En segundo lugar, si $s \in]0,1]$ denota la tasa de selección, entonces se seleccionan las primeras $[ns]$ filas de $\hat{\mathbb{E}}$ y se almacena la sub-matriz en la denominada matriz $\mathbb{F}$. Si $h = [ns]$ entonces nuestra nueva matriz será:

$$\mathbb{F} = \left(\begin{array}{ccc|c} (v_0)_{1x1} & (v_1)_{1x2} & (C)_{1x3} & cost_1 \\ \vdots & \vdots & \vdots & \vdots \\ (v_0)_{hx1} & (v_1)_{hx2} & (C)_{hx3} & cost_h \\ \hline \vdots & \vdots & \vdots & \vdots \\ (v_0)_{nx1} & (v_1)_{nx2} & (C)_{nx3} & cost_n \end{array}\right)$$

$$\mathbb{F} = \left(\begin{array}{ccc|c} (v_0)_{1x1} & (v_1)_{1x2} & (C)_{1x3} & cost_1 \\ \vdots & \vdots & \vdots & \vdots \\ (v_0)_{hx1} & (v_1)_{hx2} & (C)_{hx3} & cost_h \end{array}\right)$$

En tercer lugar, aplicando un proceso estocástico más precisamente la regla de la ruleta se seleccionan los padres desde los individuos de $\mathbb{F}$. Este proceso consiste en simular una ruleta en donde cada individuo estará representado en un segmento de ésta, el tamaño de este segmento estará estrechamente relacionado a la aptitud de cada uno de los individuos. Para nuestro algoritmo aquellos individuos que tenga un menor costo serán los que tendrán un mayor segmento representado en la ruleta.

- Calculamos el número de apareamiento o entrecruzamiento: $M = (n-h)/2$.
- Calculamos la probabilidad de selección en relación a los individuos seleccionados.
- Calculamos la probabilidad acumulativa en relación a los individuos seleccionados.
- Generamos M números al azar entre $[0, 1]$.
- Hacemos girar la ruleta tantas veces como apareamientos se requieran para comparar nuestro número al azar y ver en que segmento de la ruleta cae. Aquellos individuos que tengan un menor costo, por ende un mayor segmento representado en la ruleta, tendrán una mayor probabilidad de salir seleccionados.
- Guardamos los índices de los individuos que fueron aleatoriamente seleccionados como "padres".

(b3) **Apareamiento.** Se define la regla algebraica para el entrecruzamiento de los padres. En este trabajo de tesis se obtienen los nuevos individuos aplicando una combinación

convexa de los genes de los padres seleccionados vía un punto de entrecruzamiento. El proceso de apareamiento termina cuando $n - [ns]$ individuos son generados.

En este punto la matriz $\mathbb{E}$ es actualizada teniendo en cuenta que los padres se almacenan en las primeras $[ns]$ filas y los nuevos en las siguientes filas.

(b4) **Mutación.** Si $\mu \in [0,1]$ denota la tasa de mutación, se define el número total de mutaciones: $m := [\mu(n-1)d]$. En este punto se repite m veces el siguiente proceso de mutación: el gen aleatorio $\mathbb{E}_{ij}$ es reemplazado por un número aleatorio elegido en el intervalo $[l_j, u_j]$. Se observa que después que la mutación termina la matriz $\mathbb{E}$ es naturalmente actualizada.

4.3.3. Esquema General del Algoritmo Implementado

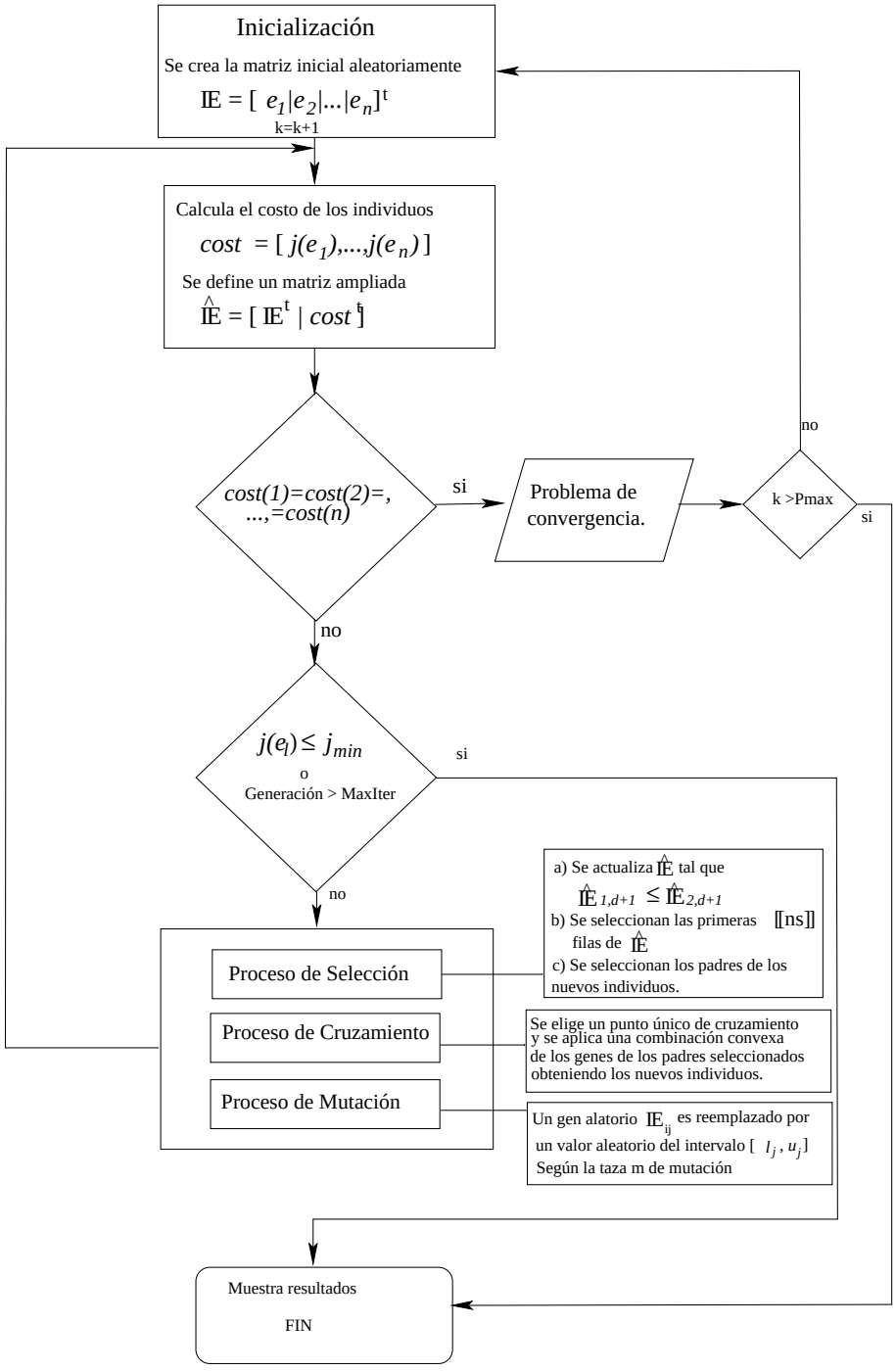


Figura 4.1: Esquema del algoritmo creado.

4.4. Especificación del Algoritmo en Pseudocódigo

Aquí se presentan los principales algoritmos propuestos mediante pseudocódigos. Los algoritmos 1 y 2 corresponden a la primera etapa presentada en la Sección 4.3.1 que tiene relación con la construcción de la función costo. Específicamente el algoritmo presentado en el cuadro Alg. 4.1 corresponde a la expresión matemática presentada en (4.3c). El algoritmo que se muestra en 4.2 satisface la expresión dada en (4.1). El algoritmo 3 presentado en 4.3 corresponde a la etapa 2 "Construcción del Algoritmo Genético" que se ilustra en la Sección 4.3.2 y que se construyo de forma tal que permita responder a la calibración del flujo en las leyes de conservación escalares con flujo convexo y también a la identificación de parámetros en la teoría de sedimentación, con flujo no convexo y punto de inflección.

1: INPUT $a, b, M, T, e, u_0, f, f'g$

2: OUTPUT $x, u0, u_N$

3: Calcule el espacio de la longitud del paso Δx,

4: Elija el número de pasos temporales N de tal manera que la CFL (6) se cumple

5: Calcular la longitud del paso temporal Δt,

6: $\lambda \leftarrow \Delta t / \Delta x$

7: Discretizar el dominio especial [a,b] en M subintervalos K_j de largo Δx y centro en $x_j \leftarrow a + j/\Delta x$ with $j = 0, \ldots, M$

8: Discretizar la condición inicial u_0 : Definida $u0 = [u_0 \ldots u_M]$, donde u_j es el promedio de u_0 más K_j

9: $u \leftarrow u_0$

10: **for** $n = 0$ to N **do**

11: $gold \leftarrow g(u_0, u_1)$

12: $uN_0 \leftarrow u_0 - \lambda(gold - f(u_0))$

13: **for** $j = 1$ to $M - 1$ **do**

14: $gnew \leftarrow g(u_j, u_j + 1)$

15: $uN_j \leftarrow u_j - \lambda(gnew - gold)$

16: $gold \leftarrow gnew$

17: **end for**

18: **end for**

Alg. 4.1: Algoritmo 1: Algoritmo para encontrar la solución numérica al perfil $u(., T)$ del problema directo

1: INPUT u^{exp}

2: OUTPUT J

3: Calcular $x, u0, u_N$ con Algoritmo 1

4: Discretizar u^exp mas I utilizando x

5: Calcular J vía valores experimentales dados en 4.8

Alg. 4.2: Algoritmo 2: Algoritmo para la función costo

1: INPUT J,d,n, MaxIter, MinCost, $s \in]0,1]$ (factor de selección), $m \in [0,1]$ (factor de mutación),

2: OUTPUT c^*

3: Generar aleatoriamente la población inicial $e_i \in \mathbb{R}^d, i = 1,\ldots,n$.

4: $\mathbb{E} \leftarrow \left[\mathbf{e}_1|\mathbf{e}_2|\cdots|\mathbf{e}_n\right]$

5: Calcular **cost** $\in \mathbb{R}^n$ tal que $cost_i = J(e_i)$, mediante el Algoritmo 2.

6: generación $\leftarrow 0$

7: **while** (generación $<$ MaxIter) and (MinCost $< cost_1$) **do**

8: generación $\leftarrow$ generación $+1$

9: *// Proceso de selección y actualiza población*

10: se actualiza el **cost** $\in \mathbb{E}$ y ordena atendiendo a $cost_1 \leq cost_2 \leq \ldots \leq cost_n$

11: se eligen los individuos que permanecerán según $n * s$

12: estocásticasmente se define los padres

13: *// Proceso de cruzamiento de los padres.*

14: se escoge un único punto de cruzamiento

15: se aplica una combinación convexa de los genes de los padres

16: se obtienen nuevos individuos y se agregan la población hasta completar $n - n * s$

17: *// Proceso de mutación*

18: se define el número de individuos a mutar según $m * n$ y aleatoriamente se seleccionan.

19: se selecciona aleatoriamente el gen del individuo a mutar

20: el gen seleccionado es reemplazado por un valor aleatorio del intervalo.

21: $e^* \leftarrow e_1$

22: Calcular **cost** $\in \mathbb{R}^n$ tal que $cost_i = J(e_i)$, mediante el Algoritmo 2.

23: **end while**

24: se muestran los valores obtenidos

25: fin

Alg. 4.3: Algoritmo 3: Algoritmo genético propuesto para encontrar los óptimos $e^* \in \mathbb{R}^d$ en J definido como $\Omega := \prod_{i=1}^{d} [min(e_i), max(e_i)]$

4.5. Resultados

En esta sección se presentan los principales resultados de aplicación de los algoritmos propuestos en la sección anterior.

4.5.1. Identificación flujo tipo Burger

El algoritmo genético presentado en este trabajo fue validado en [16]. El algoritmo se aplicó a la identificación de un flujo convexo. Más precisamente se consideró los siguientes datos en el problema directo: en la ecuación (2.15) $q(t) = 0$, $\phi_0(x) = 0.001$, $D(t) = 0.5$, $I(t) = 1$ y la condición de admisibilidad (2.11) es reemplazada por f tal que: $f'' > 0$ y $f(0) = f'(0) = 0$. En tanto que para la identificación de parámetros se consideró el flujo de tipo Burger dependiendo del parámetro c y definido como: $f(\phi) = c^{-1}\phi^c$, el cual satisface la condición de admisibilidad modificada para $c > 1$. Es claro que si $c = T = 2$, la solución del problema directo puede ser obtenida analíticamente vía el método de las características [53] y es dada por

$$\phi(x,T) = \begin{cases} 0.001, & x \in [0, 0.002], \\ x/2, & x \in]0.002, 1]. \end{cases} \tag{4.4}$$

Para el problema inverso, considerando que el perfil de observación es igual al perfil analítico (4.4), es decir $\phi^{exp} = \phi(\cdot, T)$ y el flujo numérico de Godunov [53], al aplicar el algoritmo propuesto con una población de 200 cromosomas se estimó el parámetro como $c = 2.0178$. Para mayores detalles consultar [16].

4.5.2. Identificación flujo de sedimentación

En el caso específico de la función de flujo del problema de sedimentación que se identificó fue de la siguiente forma

$$f_b(u) = v_0 u \left(1 - \frac{u}{u_{\text{máx}}}\right)^C + v_1 u^2 (u_{\text{máx}} - u),$$

donde $(v_0, v_1, C) \in \mathbb{R}_- \times \mathbb{R}_+ \times]1, \infty[$. El parámetro $u_{\text{máx}} = 0.64156$ es considerado fijo y se identifican los otros tres, es decir (v_0, v_1, C). En este ejemplo el valor de p_c en el flujo numérico

$$g(u,v) = \begin{cases} f_b(v) & u \leq p_c, \quad v \leq p_c, \\ f_b(p_c) & u \leq p_c, \quad v > p_c, \\ f_b(u) + f_b(v) - f_b(p_c) & u > p_c, \quad v \leq p_c, \\ f_b(u) & u > p_c, \quad v > p_c. \end{cases} \tag{4.5}$$

es aproximado mediante la solución de la ecuación nolineal $f_b'(\phi) = 0$ para $\phi \in [0, \phi_{\text{máx}}]$

Para obtener el perfil observado se consideró una simulación del problema directo con $T = 235$, $M = 200$, el flujo [18] y la condición inicial definidas por

$$f_b(u) = \begin{cases} 3.38433u - 13.7672u^2 \\ \quad +1.62275u^3 + 1.1264u^4 \\ \quad -9.02235u^5, & u \in (0, u_{\text{máx}}), \\ 0 & \text{en otro caso}, \end{cases} \tag{4.6}$$

$$u_0(x) = \begin{cases} 0.34, & x \in]0.08, 1[, \\ 0, & \text{en otro caso.} \end{cases} \tag{4.7}$$

Se asume que los sensores están ubicados en $\bar{x}_i = \ell/20, i = 0, \ldots, 20$. El conjunto de valores para la concentración que se obtiene vía la simulación del problema directo se muestran en la Tabla 4.1. Así, utilizando un ajuste de los datos se obtiene como perfil de observación en $T = 235$ la siguiente función:

$$u^{exp}(x) = \begin{cases} ax^2 + bx + c, & x \in [0, 0.85[\\ 0 & \text{en otro caso}, \end{cases} \tag{4.8}$$

donde $a = 0.1796, b = 0.08979$ y $c = -0.002797$.

Este perfil observado experimental se muestra en el Figura 4.2.

Para el algoritmo genético se consideró:

$$\Omega = [-0.005, -0{,}001] \times [0.0001, 0{,}0005] \times [1.1, 4].$$

La población aleatoria elegida por el algoritmo se muestran en la Tabla 4.2. Los resultados de la identificación con $M = 100$ se muestran en la Figura 4.2 y el valor de los parámetros identificados

son $v_0 = -2.5726e-3$, $v_1 = 4.2527e-4$ y $C = 1.8929$ con el valor de la función costo $1.8238e-4$. En la Figura 4.3 se muestra el valor de la función costo en el mejor individuo de la respectiva generación.

i	$\bar{x}_i$	$\bar{\phi}_i$	i	$\bar{x}_i$	$\bar{\phi}_i$
0	0.00	4.1484531e-7	11	0.50	8.7744698e-2
1	0.05	3.7612864e-4	12	0.55	1.0071052e-1
2	0.10	4.8569663e-3	13	0.60	1.1449055e-1
3	0.15	1.2928261e-2	14	0.65	1.2924273e-1
4	0.20	2.2111102e-2	15	0.70	1.4518754e-1
5	0.25	3.1848510e-2	16	0.75	1.6266122e-1
6	0.30	4.2039813e-2	17	0.80	1.8421528e-1
7	0.35	5.2683287e-2	18	0.85	2.0767693e-1
8	0.40	6.3811086e-2	19	0.90	0
9	0.45	7.5475023e-2	20	0.95	0
10	0.50	8.7744698e-2	21	1.00	0

Cuadro 4.1: Ubicación de los sensores y valores de la concentración observada.

	v_0	v_1	C	J
1	-1.1995e-3	1.2316e-4	3.5305	7.9575e-3
2	-4.0754e-3	2.4115e-4	1.1570	5.4107e-3
3	-2.5726e-3	4.2527e-4	3.0757	2.4235e-3
4	-3.0561e-3	1.0394e-4	2.2005	4.8278e-4
5	-1.4348e-3	1.5556e-4	3.5122	7.0359e-3
6	-1.9516e-3	1.8111e-4	2.5582	3.2236e-3
7	-3.1741e-3	1.7949e-4	3.1575	1.1953e-3
8	-4.9260e-3	3.4152e-4	2.3438	3.4752e-3
9	-1.7144e-3	2.0888e-4	1.9834	2.8673e-3
10	-3.2212e-3	1.7953e-4	1.6500	2.2667e-3

	v_0	v_1	C	J
11	-2.5383e-3	1.0611e-4	1.6610	8.5471e-4
12	-1.8323e-3	3.9871e-4	3.0784	4.8033e-3
13	-1.3127e-3	2.7804e-4	1.9780	4.8276e-3
14	-2.0472e-3	4.7273e-4	2.6709	3.2572e-3
15	-4.2949e-3	2.8640e-4	1.5375	4.5461e-3
16	-3.3772e-3	2.6746e-4	3.1239	8.3245e-4
17	-1.2581e-3	4.3849e-4	2.1973	5.7111e-3
18	-1.3324e-3	3.1006e-4	3.5940	7.5848e-3
19	-3.3589e-3	1.8106e-4	3.5756	1.5528e-3
20	-1.4254e-3	3.6885e-4	2.8213	6.0268e-3

Cuadro 4.2: Población inicial aleatoria de 20 individuos para la identificación.

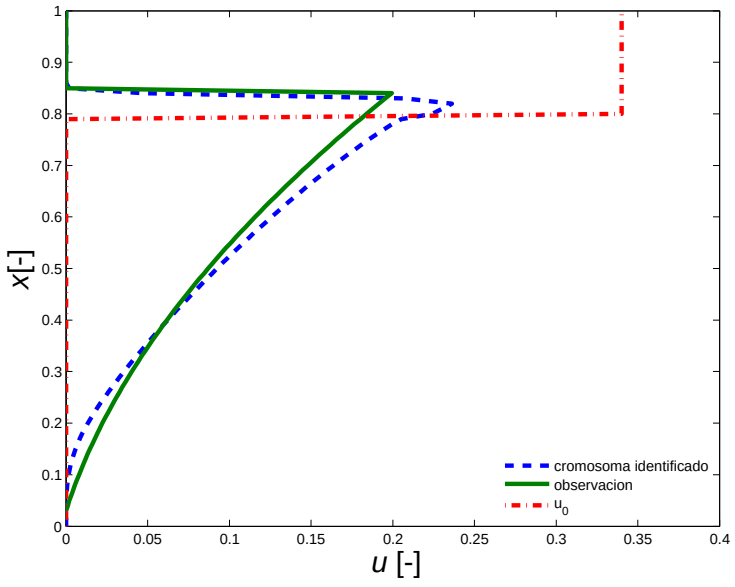


Figura 4.2: Perfil para el mejor individuo y el perfil observado, después de cinco generaciones para los valores presentados en el Cuadro 4.2.

Analizando los resultados obtenidos tenemos que, en la Figura 4.2 se muestra el perfil observado experimental, línea continua, que da cuenta del conjunto de valores para la

concentración observada, estos valores se detallan en el Cuadro 4.1. También se muestra el mejor perfil obtenido por medio de la simulación con el algoritmo genético, línea punteada, con una población de veinte individuos, haciendo iterar cinco generaciones y con un M=100.

Se puede inferir que la simulación por medio del algoritmo se ajusta muy bien, o mejor, dicho alcanza valores muy cercanos al perfil inicial observado, validando el accionar del algoritmo genético al encontrar muy buenos parámetros.

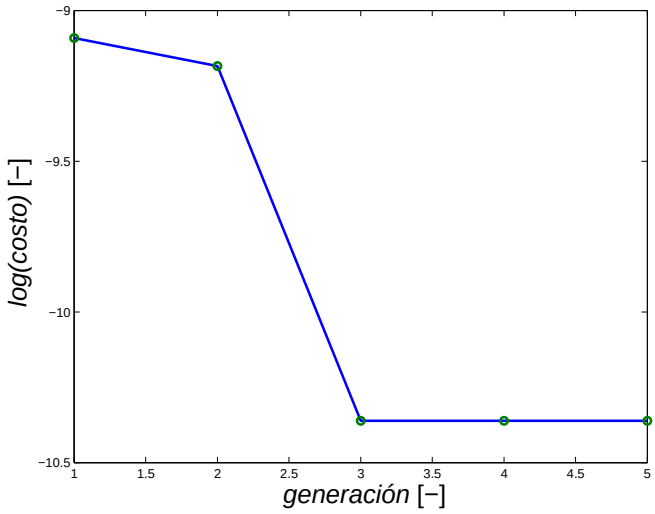


Figura 4.3: Valor de la función costo sobre el mejor individuo de cada generación.

La Figura 4.3 muestra los valores de la función costo sobre el mejor individuo encontrado en la respectiva generación y la convergencia que obtiene el algoritmo genético. Se puede ver que ya en la tercera generación se converge a valores mínimos de comparación.

4.5.3. Análisis general de la implementación y otros ejemplos

Los primeros algoritmos genéticos creados, para probar funciones simples, fueron desarrollados en el lenguaje C. Luego el desarrollo de los algoritmos aquí presentados se realizó en Matlab R(2010a). La arquitectura del computador utilizado para hacer los experimentos con los algoritmos fue un equipo marca DELL modelo vostro 3500 con la siguiente configuración:

- Procesador Intel Core I5 M520 de 2.40Ghz
- 4 Gb en memoria RAM
- 500 Gb disco duro de 720 RPM
- Sistema operativo Windows 7 (por la versión de Matlab)

En relación al mayor almacenamiento temporal requerido para el algoritmo genético propuesto, se puede indicar que está dado por el vector que contiene los genes de cada individuo, más un vector en donde se almacena la evaluación del costo de dichos genes. Estos dos vectores, multiplicados por la cantidad de individuos asignados para la experimentación nos entregan una matriz de datos con la cual se trabaja. El detalle se desglosa así:

- 1 individuo tiene 3 genes = 3 reales tipo double (64 bits c/u)
- Costo del individuo = 1 real tipo double (64 bits)

Estos datos se almacenan en un vector:

- $Vect_1 = [gen_1\ gen_2\ gen_3\ costo_1]$ = 4 reales tipo double (64 bits c/u); 1 individuo = 256 bits
- $Matriz$ = número de individuos $n * Vetc_1$
- Si n=20, entonces;
- $Matriz$ = (20 * 256) = 5.120 bits; 640 Bytes;
- Si n=100, entonces;
- $Matriz$ = (100 * 256) = 25.600 bits; 3.200 Bytes;

Adicional a la matriz señalada se tiene una segunda matriz temporal para el momento de la selección de los individuos que pasan a la siguiente generación. Como el algoritmo genético propuesto desecha los individuos “menos aptos” y los reemplaza por nuevos individuos (hijos) no se produce un crecimiento en la matriz original, se mantiene su cantidad de datos según los parámetros entregados inicialmente.

A continuación se presenta un experimento adicional con las siguientes condiciones: número de individuos de la población 20 (n=20), máxima iteración de generaciones 5 ($MaxIter$=5), M=50.

En el Cuadro 4.3 se muestra la población aleatoria generada de acuerdo a los parámetros definidos anteriormente. Junto a los valores generados se encuentra la columna J que muestra el costo, según la función de evaluación, asociado a cada individuo. Por otro lado, en el Cuadro 4.4 se muestra la última generación obtenida luego de que el algoritmo genético realizara las operaciones de selección, cruce y mutación definidas. Se puede observar que la columna J ya está ordenada dejando en primer lugar el individuo que tiene mejor costo de representación, en este caso, el menor valor.

Se ha incluido un tercer cuadro de resultados, el Cuadro 4.5 muestra los mejores individuos y sus costos asociados de cada generación iterada, se aprecia que para esta experimentación en particular, el algoritmo obtuvo su mayor convergencia en la cuarta generación.

También se han incluido las gráficas que muestran, por un lado el perfil observado experimental y el perfil simulado con el algoritmo genético. Estas gráficas corresponden a la primera generación, figuras de la 4.4. a 4.8. y luego a la última generación, figuras de la 4.9. a 4.13.

	V0	*V1*	*C*	*J*
1	-3,071E-03	3,848E-04	3,767E+00	4,949E-03
2	-2,840E-03	2,952E-04	2,216E+00	1,018E-03
3	-4,116E-03	3,470E-04	1,572E+00	3,907E-03
4	-4,616E-03	1,855E-04	3,411E+00	3,649E-03
5	-4,759E-03	3,583E-04	1,430E+00	5,177E-03
6	-1,722E-03	2,523E-04	1,561E+00	1,229E-03
7	-1,914E-03	1,415E-04	2,132E+00	2,870E-03
8	-4,217E-03	2,510E-04	3,558E+00	3,737E-03
9	-1,420E-03	2,051E-04	2,790E+00	8,471E-03
10	-2,263E-03	1,965E-04	2,800E+00	6,709E-03

	V0	*V1*	*C*	*J*
11	-2,373E-03	3,492E-04	3,785E+00	6,453E-03
12	-1,038E-03	3,092E-04	2,768E+00	9,276E-03
13	-4,865E-03	2,653E-04	1,129E+00	6,037E-03
14	-3,303E-03	1,871E-04	3,447E+00	4,753E-03
15	-3,040E-03	4,434E-04	2,866E+00	5,182E-03
16	-2,666E-03	4,444E-04	2,492E+00	3,641E-03
17	-4,667E-03	2,136E-04	1,878E+00	3,586E-03
18	-2,359E-03	3,462E-04	1,848E+00	3,548E-04
19	-4,791E-03	4,118E-04	2,495E+00	6,515E-04
20	-2,773E-03	4,819E-04	1,759E+00	5,991E-04

Cuadro 4.3: Población inicial aleatoria de 20 individuos.

	V0	*V1*	*C*	*J*
1	-2,557E-03	3,462E-04	1,848E+00	1,322E-04
2	-2,425E-03	3,462E-04	1,848E+00	2,167E-04
3	-2,367E-03	3,462E-04	1,848E+00	3,353E-04
4	-2,359E-03	3,462E-04	1,848E+00	3,548E-04
5	-1,930E-03	2,523E-04	1,561E+00	4,259E-04
6	-2,313E-03	3,462E-04	1,848E+00	4,910E-04
7	-2,565E-03	-5,501E-04	2,216E+00	5,502E-04
8	-2,773E-03	4,819E-04	1,759E+00	5,991E-04
9	-2,536E-03	-5,501E-04	2,216E+00	6,207E-04
10	-4,791E-03	4,118E-04	2,495E+00	6,515E-04

	V0	*V1*	*C*	*J*
11	-2,565E-03	4,185E-04	1,561E+00	9,091E-04
12	-2,359E-03	3,462E-04	1,450E+00	9,403E-04
13	-2,367E-03	-5,199E-04	2,216E+00	1,039E-03
14	-2,061E-03	3,462E-04	1,848E+00	1,309E-03
15	-2,773E-03	2,599E-03	1,759E+00	1,333E-03
16	-1,977E-03	3,462E-04	1,848E+00	1,630E-03
17	-2,565E-03	3,160E-04	2,216E+00	1,667E-03
18	-2,773E-03	-1,635E-03	1,759E+00	1,809E-03
19	-4,791E-03	-1,981E-03	1,848E+00	4,629E-03
20	-2,367E-03	2,739E-03	1,848E+00	5,847E-03

Cuadro 4.4: Población final después de iterar 5 generaciones.

Generación	*V0*	*V1*	*C*	*J*
1	-2,3594E-03	3,4616E-04	1,8485E+00	3,5482E-04
2	-2,3594E-03	3,4616E-04	1,8485E+00	3,5482E-04
3	-2,3594E-03	3,4616E-04	1,8485E+00	3,5482E-04
4	-2,5569E-03	3,4616E-04	1,8485E+00	1,3223E-04
5	-2,5569E-03	3,4616E-04	1,8485E+00	1,3223E-04

Cuadro 4.5: Individuos y costos óptimo por generación.

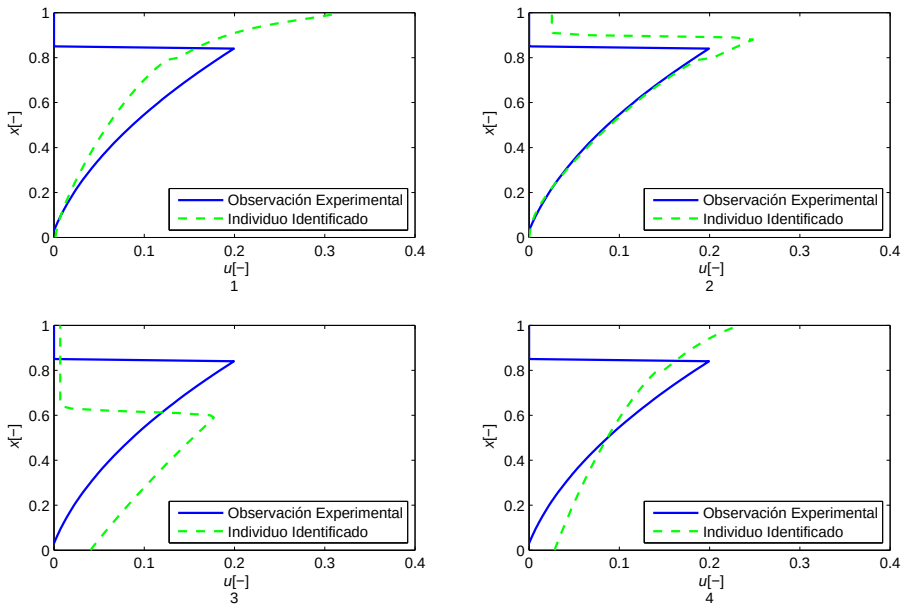


Figura 4.4: Individuos del 1 al 4 de la primera generación

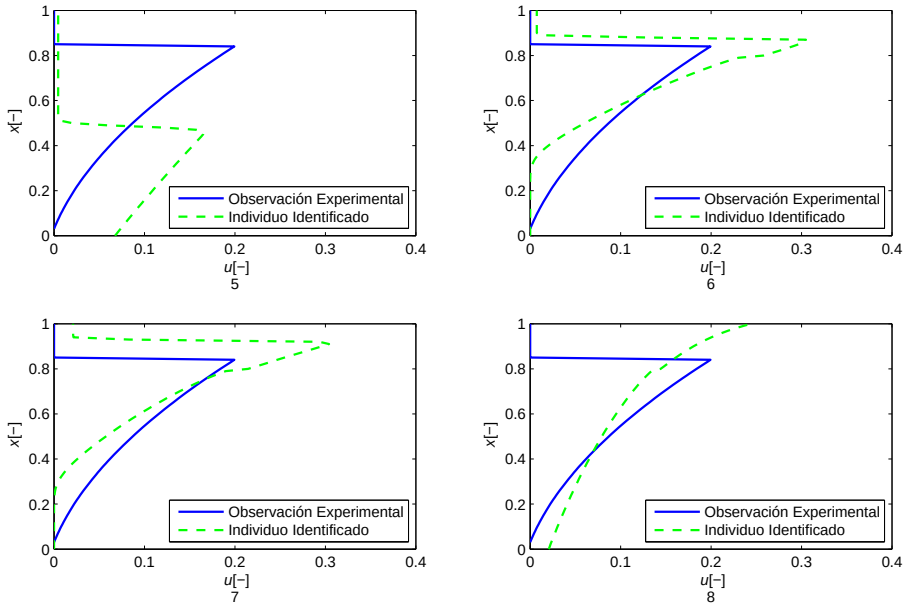


Figura 4.5: Individuos del 5 al 8 de la primera generación

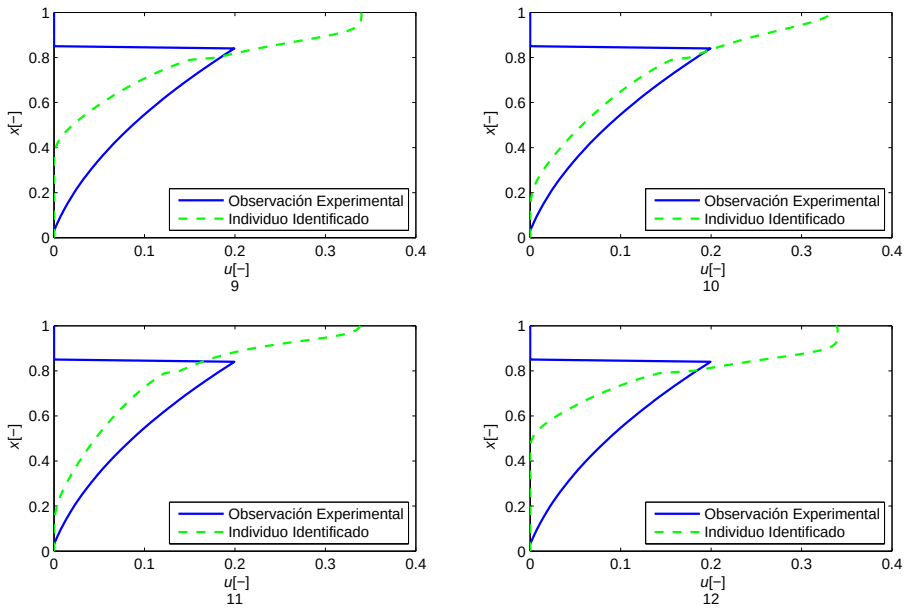


Figura 4.6: Individuos del 9 al 12 de la primera generación

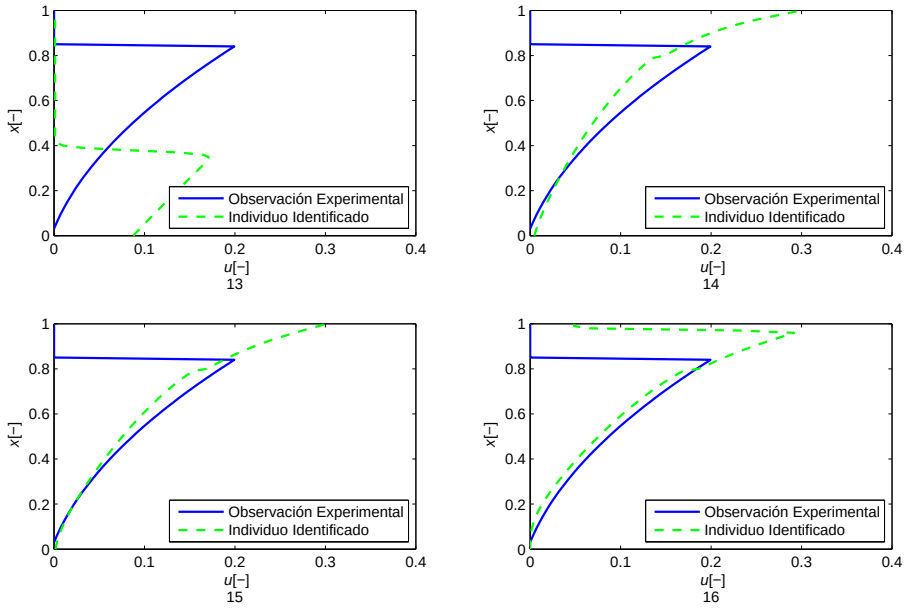


Figura 4.7: Individuos del 13 al 16 de la primera generación

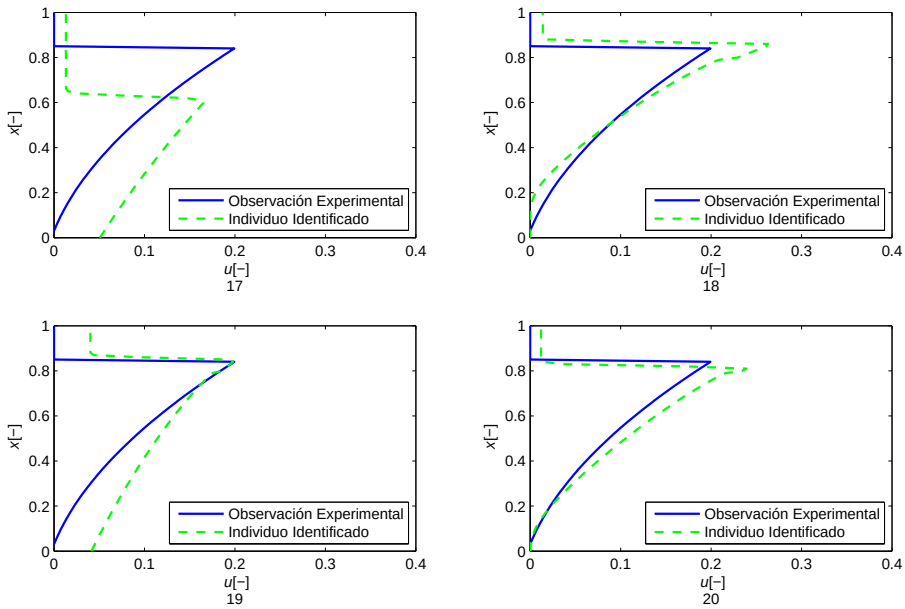


Figura 4.8: Individuos del 17 al 20 de la primera generación

A continuación se presentan las gráficas de la última generación iterada, se opta por mostrarlos desde el último individuo, gráfico etiquetado como 20 dentro de la figura 4.9., al primer individuo, gráfico etiquetado como 1 dentro de la figura 4.13. De esta forma se busca ilustrar, gráficamente, como el perfil simulado y obtenido por el algoritmo genético se va ajustando el perfil observado experimental a medida que el algoritmo itera.

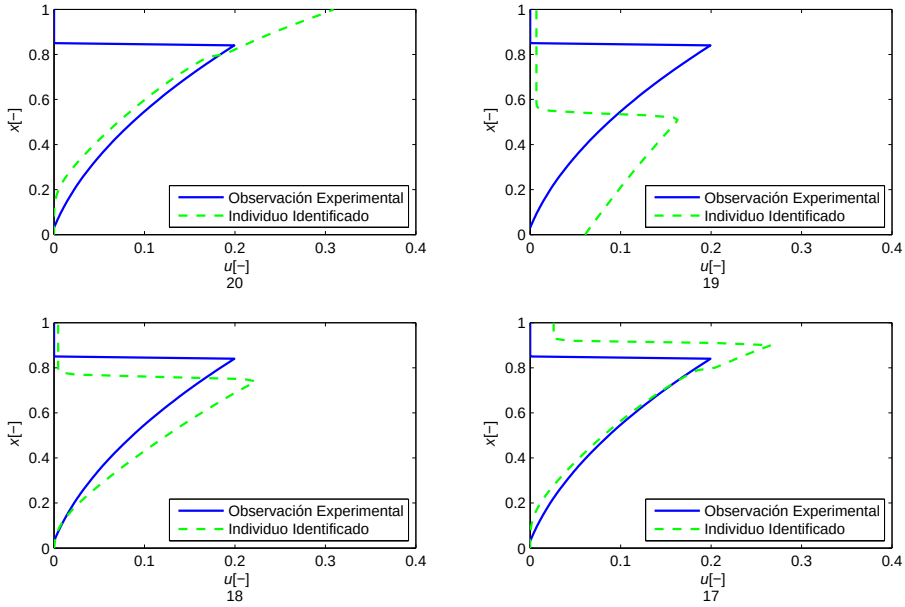


Figura 4.9: Individuos del 17 al 20 de la última generación

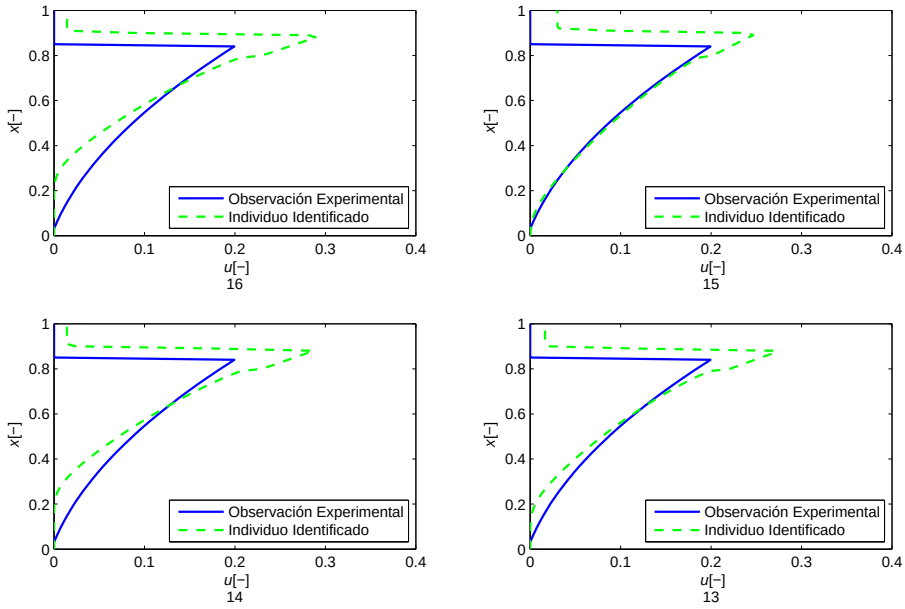


Figura 4.10: Individuos del 13 al 16 de la última generación

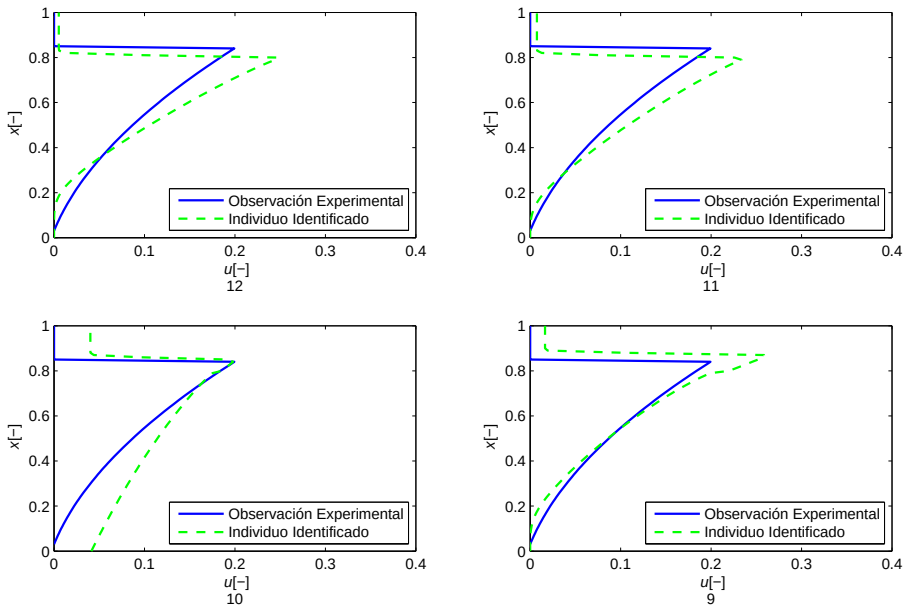


Figura 4.11: Individuos del 9 al 12 de la última generación

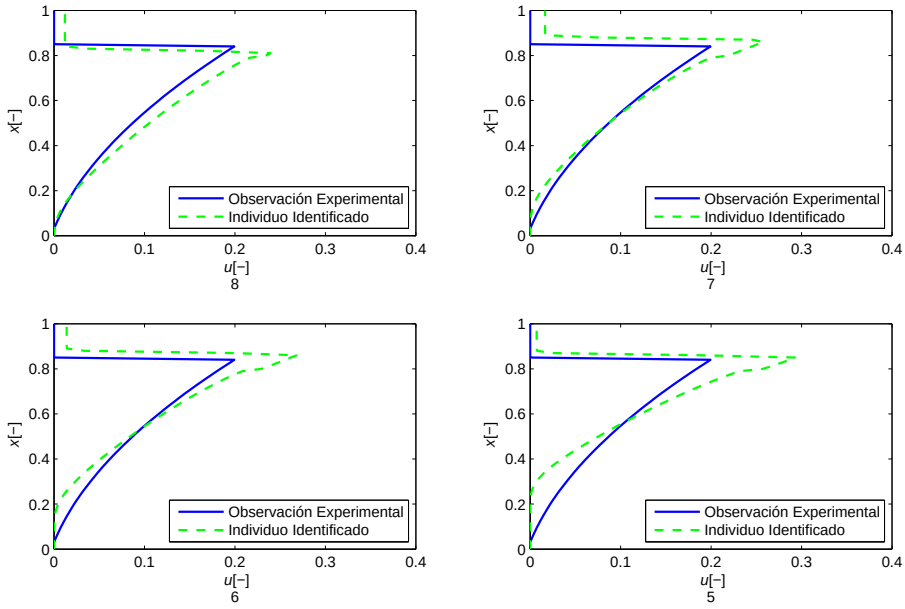


Figura 4.12: Individuos del 5 al 8 de la última generación

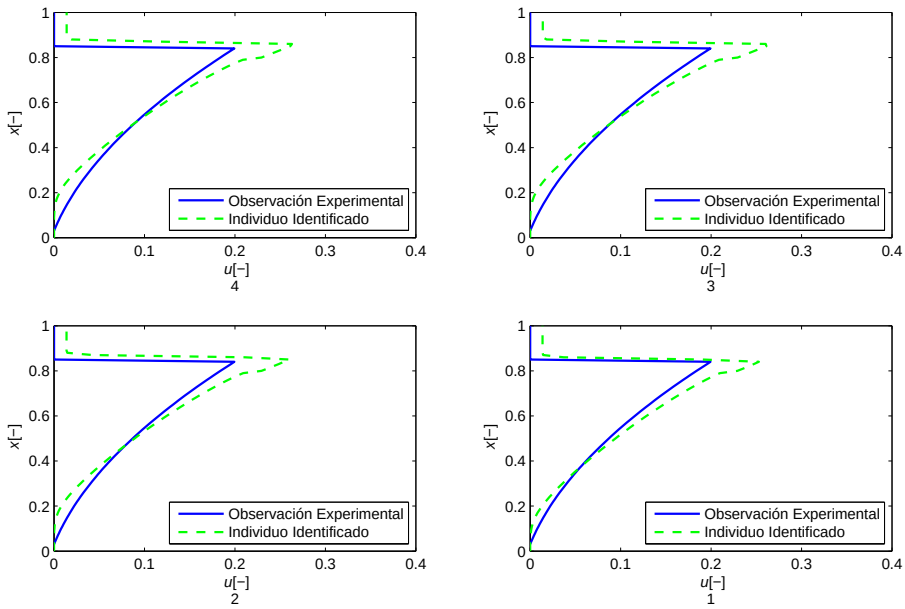


Figura 4.13: Individuos del 1 al 4 de la última generación, mejor perfil evaluado el individuo número 1

La Figura 4.14 muestra gráficamente el mejor individuo de cada generación y la Figura 4.15 muestra el perfil de este individuo, similar al perfil observado presentado en la Figura 4.2.

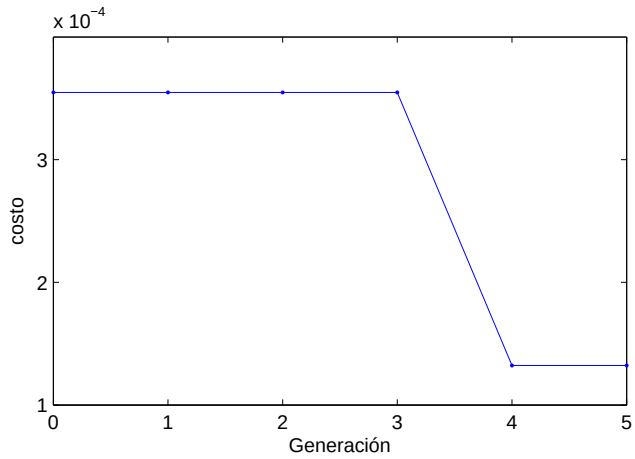


Figura 4.14: Valor de la función costo sobre el mejor individuo de cada generación

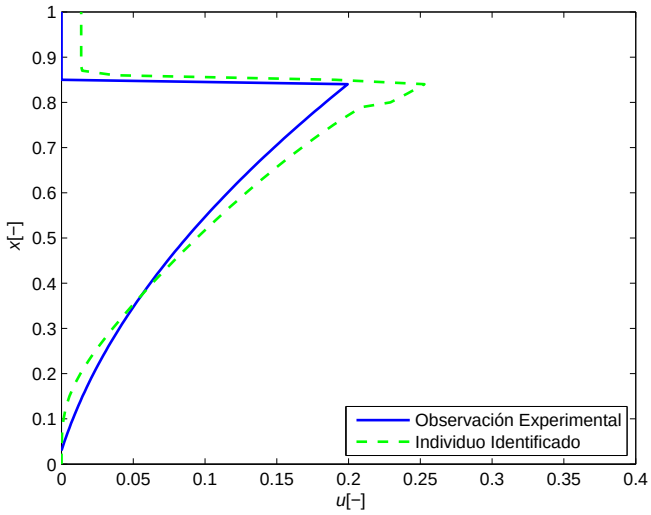
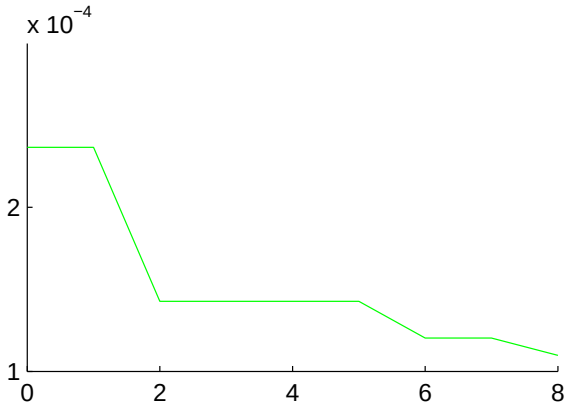
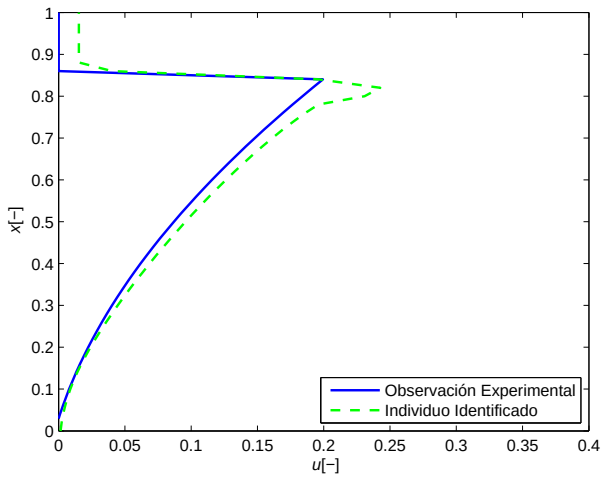


Figura 4.15: Perfil del mejor individuo después de cinco generaciones para los valores presentados en el Cuadro 4.5.

A continuación se muestran otras gráficas de convergencia y perfiles obtenidos mediante la simulación del algoritmo genético con distintos valores para el número de individuos, generaciones iteradas y el valor de M.

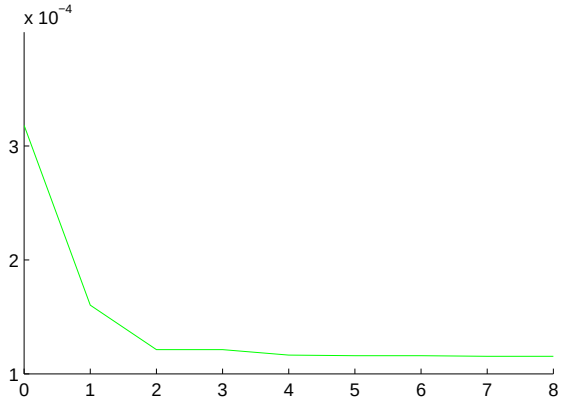


(a)

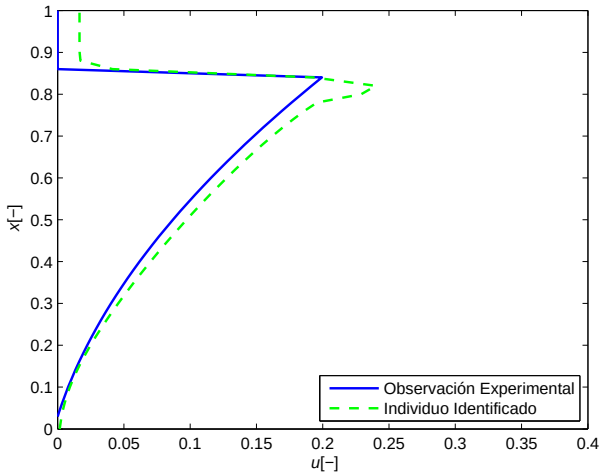


(b)

Figura 4.16: a) Valor de la función costo sobre el mejor individuo de cada generación; n=40; $MaxIter$=8; M=50. b) Perfil del mejor individuo según los siguientes valores: n=40; $MaxIter$=8; M=50.

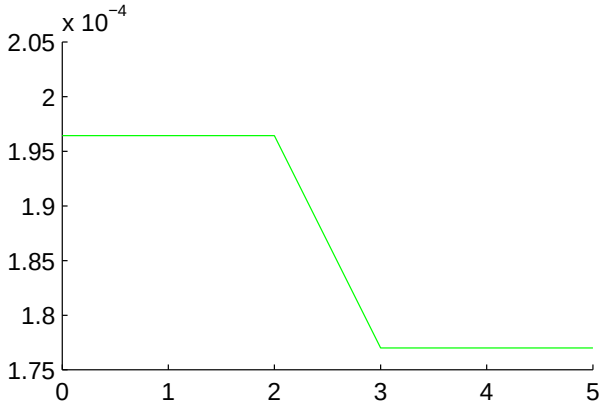


(a)

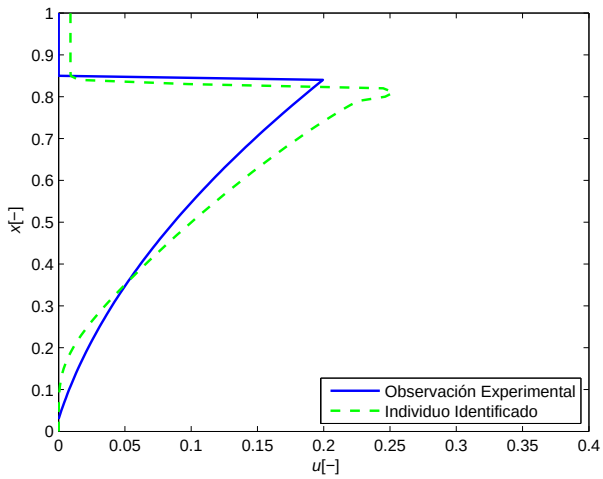


(b)

Figura 4.17: a) Valor de la función costo sobre el mejor individuo de cada generación; n=100; $MaxIter$=8; M=50. b) Perfil del mejor individuo según los siguientes valores: n=100; $MaxIter$=8; M=50.

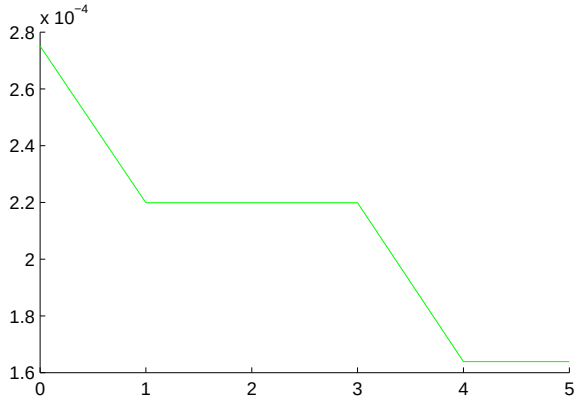


(a)

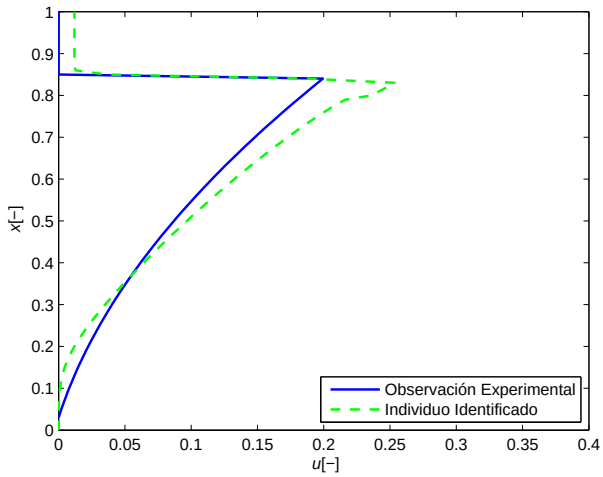


(b)

Figura 4.18: a) Valor de la función costo sobre el mejor individuo de cada generación; n=20; $MaxIter$=5; M=100. b) Perfil del mejor individuo según los siguientes valores: n=20; $MaxIter$=5; M=100.

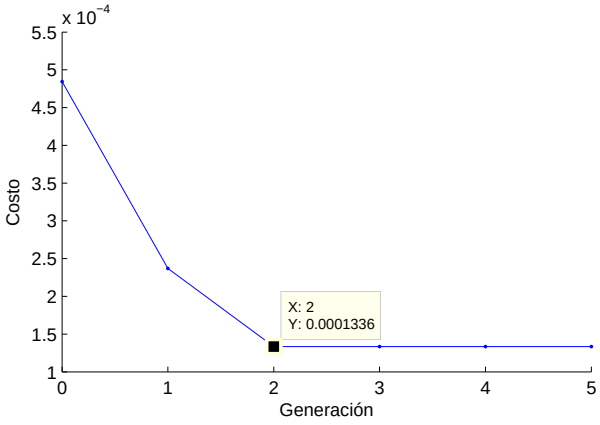


(a)

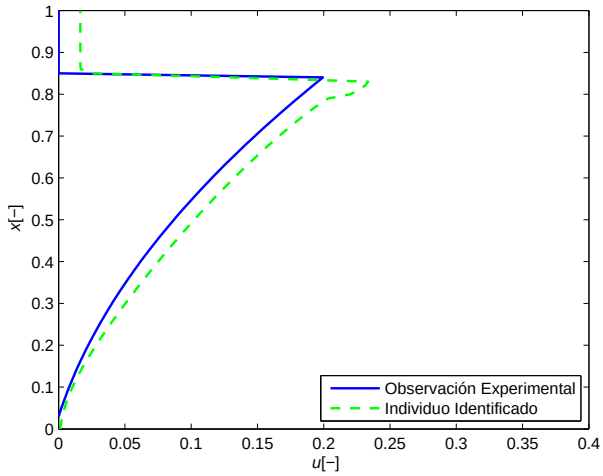


(b)

Figura 4.19: a) Valor de la función costo sobre el mejor individuo de cada generación; n=100; $MaxIter$=5; M=100. b) Perfil del mejor individuo según los siguientes valores: n=100; $MaxIter$=5; M=100.



(a)



(b)

Figura 4.20: a) Valor de la función costo sobre el mejor individuo de cada generación; n=100; $MaxIter$=5; M=200. b) Perfil del mejor individuo según los siguientes valores: n=100; $MaxIter$=5; M=200.

Un análisis importante de realizar es el de conocer los tiempos promedios que le toma, al algoritmo genético propuesto, realizar la simulación e identificar los parámetros adecuados para entregar una óptima solución. En este contexto se realizaron múltiples simulaciones para diferentes valores de n (número de individuos) y distintas generaciones.

El Gráfico 4.21 muestra en el eje X los distintos tamaños de población para el cual fue testeado, en el eje Y se muestra el tiempo en segundos. Se proyectan las líneas de tendencia

para cinco y ocho generaciones. De acuerdo a los valores entregados por la muestra se obtiene que; a mayor cantidad de población mayor es el tiempo que le toma al algoritmo en entregar el mejor individuo. Así también a mayor número de generaciones es mayor el tiempo que le toma al algoritmo. Esta última observación no es lineal producto que, en cierta combinación de generaciones y tamaño de población, los tiempos de ejecución son muy cercanos.

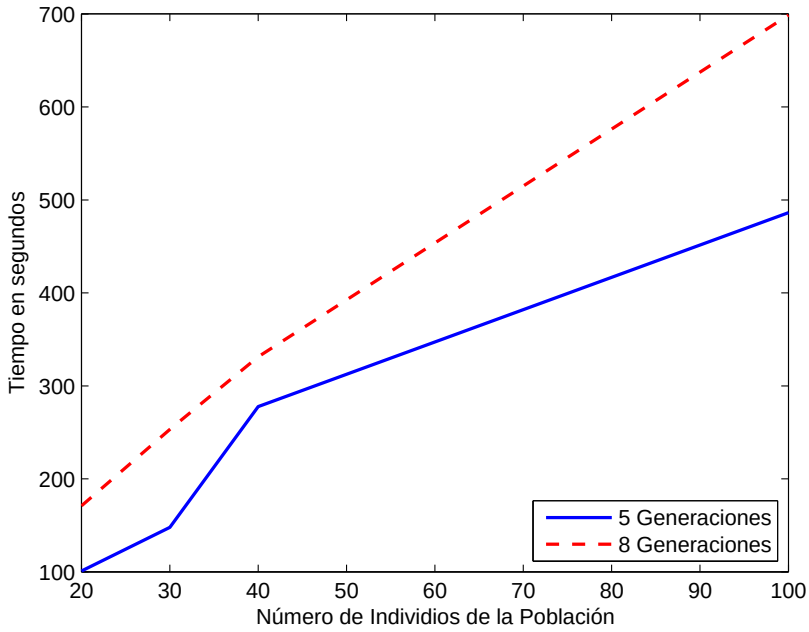


Figura 4.21: Gráfico relación de tiempo requerido por el algoritmo genético para distintos tamaños de población, distintas generaciones y M=50.

Otro análisis interesantes es el de conocer los valores de costos que toman los mejores individuos según la cantidad de población observada. El Gráfico 4.22 muestra en el eje X los distintos tamaños de población para el cual fue testeado, en el eje Y se muestra los costos asociados a cada individuo identificado. De acuerdo a lo muestreado podemos inferir que; a menor número de la población el costo asociado a cada individuo es mayor por lo cual, si bien es una solución aceptable, se podrían obtener mejores costos. A su vez, a mayor número de población observada el costo de los individuos identificados es menor, entregando una solución bastante mejor. Además

se observa que a mayores generaciones iteradas estas influyen en el costo asociado.

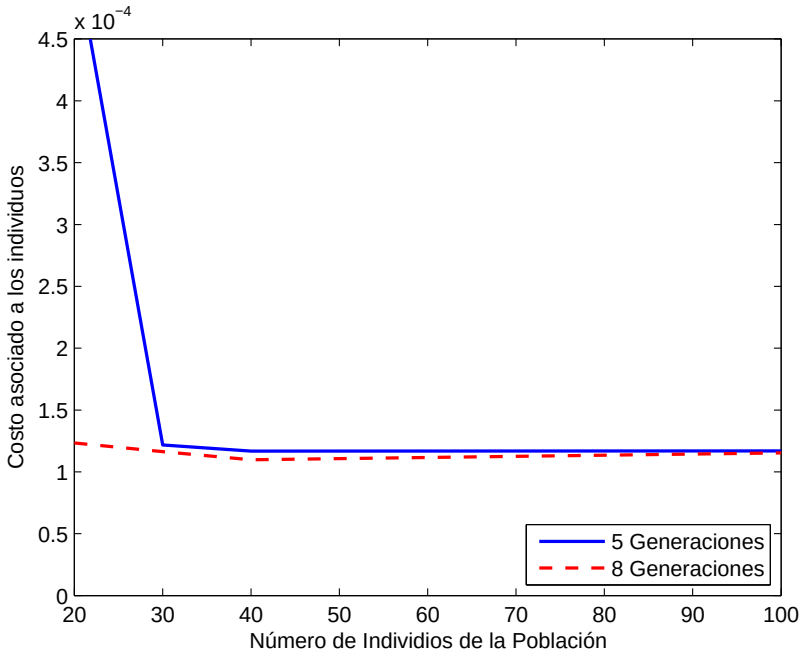


Figura 4.22: Gráfico que muestra los costos asociados a los mejores individuos identificados por el algoritmo genético, para distintos tamaños de población y generaciones.

Capítulo 5

Conclusiones y Perspectivas

En base a los objetivos planteados en la presente tesis se concluye los siguiente:

(a) En términos generales, en este trabajo se buscó calibrar numéricamente las leyes constitutivas que definen los coeficientes de una ecuación diferencial parcial, que modela la separación sólido-fluido, mediante la construcción y el análisis de un algoritmo genético apropiado. De acuerdo a la literatura analizada y expuesta en las Secciones 3.1, 3.2, 3.3 y 3.4 se puede deducir que las técnicas de programación mediante algoritmos genéticos son una buena herramienta de desarrollo para abordar la calibración de parámetros en la modelación del proceso de sedimentación sólido-fluido. Existen diversas aplicaciones de algoritmos genéticos a problemas de optimización, generados desde un problema inverso, las cuales han arrojado buenos resultados dadas las características propias de este tipo de algoritmo. La óptica para abordar el problema inverso consistió en formular el problema inverso como un problema de optimización y luego aproximarlo utilizando el método de volúmenes finitos.

(b) Se simuló el proceso de sedimentación utilizando una versión discreta para el modelo de Kynch, donde el flujo físico fue aproximado por el flujo de Godunov. Estas simulaciones permitieron crear observaciones, las cuales fueron tomadas como experimentos para el proceso de calibración.

(c) Se diseño un algoritmo genético basado en operadores y que se diferencia del usualmente utilizado en que utiliza números reales en vez de la representación binaria. Esta idea

básicamente, permite tener un algoritmo capaz de ser aplicado de manera dinámica a muchas situaciones análogas, por ejemplo los modelos de sedimentación utilizando fuerzas centrípetas o centrifugación.

(d) En el ámbito de la identificación de parámetros para la modelación del proceso de sedimentación, se encontraron trabajos que utilizan la técnica del gradiente, siendo este un método que tiene serias dificultades al buscar el óptimo global dentro de varios óptimos. Esta situación permite inferir que la utilización de estrategias evolutivas, como los algoritmos genéticos, son una buena opción para investigar y programar una solución algorítmica a este problema. Se creó un algoritmo genético acorde al problema planteado, con las características necesarias para abordar y entregar una solución al complejo problema, tarea no trivial pero finalmente exitosa pues, al realizar las pruebas con el algoritmo propuesto se logran obtener resultados satisfactorios. El algoritmo genético creado converge entregando valores más que razonable y admisibles para considerarlos como solución óptima.

Por otro lado, dentro de los posibles trabajos futuros están las siguientes tareas:

(a) Optimizar el algoritmo propuesto.

(b) Aplicar otras técnicas informáticas, como la de paralelización del algoritmo y/o buscar otras combinaciones de métodos que permitan crear un algoritmo híbrido entre el algoritmo presentado y alguna otra técnica ya investigada.

(c) Realizar nuevas simulaciones con valores experimentales observados en un laboratorio y que representen a distintos compuestos físicos.

(d) Realizar investigación de métodos alternativos como por ejemplo algoritmia asociada a Entropía Cruzada.

Bibliografía

[1] Becker R. *Espesamiento continuo: diseño y simulación de espesadores* Engineering Thesis, Dordrecht, Universidad de Concepción, Chile, 1982.

[2] S. Berres, R. Bürger, A. Coronel, and M. Sepúlveda. Numerical identification of parameters for a strongly degenerate convection-diffusion problem modelling centrifugation of flocculated suspensions. *Appl. Numer. Math.*, 52(4):311–337, 2005.

[3] Berres, S., Bürger, R., Coronel, A., Sepúlveda, M. *Numerical identification of parameters for a flocculated suspension from concentration measurements during batch centrifugation* Chemical Engineering Journal, Vol. 111, pp. 91-103., 2005.

[4] F. E. Benth and K. H Karlsen, *Note on Merton's Portfolio Selection Problem for the Schwartz Mean-Reversion Model* Stochastic Analysis and Applications, Vol. 23, pp. 687–704, 2005.

[5] M. C. Bustos, F. Concha, R. Bürger, and E. M. Tory. *Sedimentation and thickening*, volume 8 of *Mathematical Modelling: Theory and Applications*. Kluwer Academic Publishers, Dordrecht, 1999.

[6] R. Bürger, Aníbal Coronel, and Mauricio Sepúlveda. A numerical descent method for an inverse problem of a scalar conservation law modelling sedimentation. In *Numerical Mathematics and Advanced Applications: Numerical Mathematics and Advanced Applications: Proceedings ENUMATH 2007*, pages 225–232. Springer Verlag, Providence, RI, 2008.

[7] R. Bürger, Aníbal Coronel, and Mauricio Sepúlveda. Numerical solution of an inverse problem for a scalar conservation law modelling sedimentation. In *Hyperbolic problems:*

theory, numerics and applications, volume 67 of *Proc. Sympos. Appl. Math.*, pages 445–454. Amer. Math. Soc., Providence, RI, 2009.

[8] R. Bürger and W. L. Wendland. Sedimentation and suspension flows: historical perspective and some recent developments. *J. Engrg. Math.*, 41:101–116, 2001.

[9] Cerro J., Valero J. and Barrientos A. Identification of a small unmanned helicopter model using genetic algorithms. *IEEE/RSJ International Conference on Intelligent Robots and Systems.*, 3360-3365, 2005.

[10] Chadan, K., Célestin P. Inverse problems in quantum scattering theory. *Springer-Verlag.*, ISBN 0387080929, 1977.

[11] Chincao, J. *Metaheurística e Ingeniería del Software.*. Tesís Doctoral. Universidad de Málaga. 2007.

[12] Clercq, J. D., Nopens, I., Defrancq, J. y Vanrolleghem, P. A. Extending and calibrating a mechanistic hindered and compression settling model for activated sludge using in-depth batch experiments. *Water Research WATER RES.*, Vol. 42,pp. 781-791, 2008.

[13] F. Concha. *Manual de Filtración y Separación*. Centro de Tecnología Mineral, CETTEM, Concepción, Chile, 2001.

[14] F. Concha and R. Bürger. Thickening in the 20th century: a historical perspective. *Minerals & Metallurgical Process*, 20(2):57–67, 2003.

[15] Coronel, A., James, F., Sepúlveda, M. Numerical identification of parameters for a model of sedimentation processes *Inverse Problems*, Vol. 19, pp. 951-972, 2003.

[16] A. Coronel, N. Guerrero, R. Lagos, and E. Lozada. A continuous genetic algorithm for flux calibration in scalar conservation laws. *Canadian Journal on Computing in Mathematics, Natural Sciences, Engineering and Medicine*, 2(6):139–143, 2011.

[17] A. Coronel, R. Lagos, and E. Lozada. Numerical flux identification in scalar conservation laws by application of a continuous genetic algorithm. *Submitted*, 2011.

[18] A. Coronel. An application of elementary wave interaction to an inverse problem in scalar conservation laws. *Submitted*, 2011.

[19] Á. Fernández, J.P., et al. S.E.V y Optimización Global: una visión diferente de los modelos equivalentes en geofísica aplicada. *Boletín Geológico y Minero.*, ISSN: 0366-0176, 118 (1): 37-50, 2007.

[20] L.J. Fogel, A.J. Owens, and M.J. Walsh. *Artificial intelligence through simulated evolution*. Wiley, Chichester, WS, UK, 1966.

[21] R. Flecher. Practical methods of optimization *John Wiley & Sons.* New York, 1987.

[22] Gunzburger, M. D. Perspectives in flow and Optimization *Volume II of Advances in design and Control Series, SIAM* Philadelphia, 2003.

[23] F. Glover. Future paths for integer programming and links to artificial intelligence *Computers & Operations Research* 13:533-549, 1986.

[24] David E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1st edition, 1989.

[25] Randy L. Haupt and Sue Ellen Haupt. *Practical genetic algorithms*. Wiley-Interscience [John Wiley & Sons], Hoboken, NJ, second edition, 2004. With 1 CD-ROM (Windows).

[26] Harris, S.D., & Ingham, D.B Parameter identification within a porous medium using genetic algorithms, In *Handbook of Porous Media II*, pages 678–742. CRC Press 2005.

[27] A. Hoffman. *Arguments on Evolution: A Paleontologist Perspective*. Oxfor of Michigan Press, 1975.

[28] J.H. Holland. *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor, MI, USA, 1975.

[29] Hong, T., Xiao, X., Wen, D., Yang, X Parameter identification for modeling river network using a genetic algorithm. *Journal of Hydrodynamics* Ser. B, Vol. 22, Pp 246-253, No 2, 2010.

[30] V. Isakov. *Inverse problems for partial differential equations*, volume 127 of *Applied Mathematical Sciences*. Springer-Verlag New York, Inc., 1998.

[31] S. Izquierdo, Cubero A., Fueyo N. Diseño Inverso de Chimeneas Industriales Mediante Algoritmos Genéticos: Minimización del Impacto Ambiental y Reducción de Costes. *Ingeniería química* ISSN 0210-2064, Nº. 417, pags. 124-129, 2004.

[32] James, F., and Sepúlveda, M. Convergence result for the flux identification in a scalar conservation law *SIAM J. Control Optim. 37* Nº 3, 869-891, 1989.

[33] Johnson, R.W., Melich, M.E., Michalewicz, Z., Schmidt, M., Coevolutionary Optimization of Fuzzy Logic Intelligence for Strategic Decision Support. *IEEE Transactions on Evolutionary Computation*, Vol.9, No.6, pp.682-694, 2005.

[34] J. Kennedy, R. Eberhart. Swarm intelligence. *Morgan Kaufmann Publishers*, San Francisco, 2001.

[35] Kewley, Robert y Mark Embrechts Computational military tactical planning system. *IEEE Transactions on Systems, Man and Cybernetics, Part C - Applications and Reviews*, vol.32, no.2, p.161-171. 2002.

[36] A. Kirsch. *An introduction to the mathematical theory of inverse problems*, volume 120 of *Applied Mathematical Sciences*. Springer-Verlag New York, Inc., 1996.

[37] G. J. Kynch. A theory of sedimentation. *Trans. Faraday Soc*, 48:166–176, 1952.

[38] Mitchell, Melanie. *An Introduction to Genetic Algorithms*. MIT Press, p.2. , 1996.

[39] Nocedal, Jorge and Wright, Stephen J. *Numerical Optimization*. Springer, 2000.

[40] Obayashi, Shigeru, Daisuke Sasaki, Yukihiro Takeguchi, y Naoki Hirose. Multiobjective evolutionary computation for supersonic wing-shape optimization *IEEE Transactions on Evolutionary Computation*, vol.4, no.2, p.182-187. 2000.

[41] J. Pérez. Contribución a los métodos de optimización basados en procesos naturales y su aplicación a la medida de antenas en campo próximo. *Universidad de Cantabria.*, Tesis Doctoral, España 2005.

[42] A. Prilepko, D. G. Orlovskii, I. A. Vasin. *Methods for Solving Inverse Problems in Mathematical Physics*. Marcel Dekker, Inc., New York, 2000.

[43] Sato, S., K. Otori, A. Takizawa, H. Sakai, Y. Ando y H. Kawamura. Applying genetic algorithms to the optimum design of a concert hall *Journal of Sound and Vibration*, vol.258, no.3, p. 517-526 2002.

[44] P. H. Sanchez *Leyes de Conservación Escalares en Procesos de Separación* Memoria de Título, Universidad del Bío-Bío, 2009.

[45] S. N. Sivanandam and S. N. Deepa. *Introduction to genetic algorithms*. Springer, Berlin, 2008.

[46] M. Sipser. Introduction to the theory of computation. Course technology, 2005.

[47] M. Taranillla, Leguizamón, M., Gagliardi, E., Hernández P. Estudio de la aplicabilidad de un enfoque evolutivo para la descomposición en Suma de Minkowski. *Anales del X Congreso Argentino de Ciencias de la Computación, CACIC 2004*, La Matanza, Bs.As., 2004.

[48] E. M. Tory. Stochastic sedimentation and hydrodynamic diffusion. *Chemical Engineering Journal*, 80:8189, 2000.

[49] A. M. Turing. Computing Machinery and Intellence. *Mind Vol.59*, 94–101, 1950.

[50] B. Engquist and S. Osher. One-sided difference approximations for nonlinear conservation laws. *Math. Comput.*, 36:321–351, 1981.

[51] Zbigniew Michalewicz. *Genetic algorithms + data structures = evolution programs*. Artificial Intelligence. Springer-Verlag, Berlin, 1992.

[52] Robert Eymard, Thierry Gallouët, and Raphaèle Herbin. Finite volume methods. In *Handbook of numerical analysis, Vol. VII*, Handb. Numer. Anal., VII, pages 713–1020. North-Holland, Amsterdam, 2000.

[53] Randall J. LeVeque. *Numerical methods for conservation laws*. Lectures in Mathematics ETH Zürich. Birkhäuser Verlag, Basel, second edition, 1992.

[54] Eleuterio F. Toro. *Riemann solvers and numerical methods for fluid dynamics*. Springer-Verlag, Berlin, third edition, 2009. A practical introduction.

[55] John R. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, Cambridge, MA, USA, 1992.

[56] I Rechenberg. *Evolutionstrategie: Optimierung Technisher Systeme nach Prinzipien des Biologischen Evolution*. Fromman-Hozlboog Verlag, Stuttgart, 1973.

[57] Porto, Vincent, David Fogel y Lawrence Fogel. Alternative neural network training methods. *IEEE Expert*, vol.10, no.3, p.16-22. 1995.

Apéndice A

Anexo

A.1. Lista de Símbolos

J_i	:	Flujo de densidad.
$\mathbf{v}$	:	Velocidad promedio de la mezcla.
J_D	:	Flujo de difusión.
$\mathbf{v}_s$	:	Velocidad del sólido.
$\mathbf{v}_f$	:	Velocidad del fluido.
m_i	:	Masa.
$\mathbf{u}_i$	:	Velocidad de difusión.
H	:	Altura de la mezcla.
$\phi(x,t)$	:	Fracción volumétrica local de sólidos en el punto x y tiempo t.
$q(t)$	:	Velocidad volumétrica promedio.
f	:	Función de densidad de flujo de sólidos de Kynch.
L	:	Altura de la mezcla.
$I(t)$ y $D(t)$	:	Concentraciones de sólidos de la columna de sedimentación en tiempo t.
$t = T$	:	Tiempo fijo.
$\mathbf{e} \in \mathbb{R}^d$	:	Vector de parámetros.
$\phi(x,T;e)$	:	Perfil de simulación.
$\phi^{\exp(x)}$	:	Perfil experimental.