

Solución numérica de un problema inverso aplicando un algoritmo genético continuo

STEFAN BERRES^a, ANÍBAL CORONEL^{b*}, RICHARD LAGOS^c

^a Universidad Católica de Temuco, Departamento de Ciencias Matemáticas y Físicas, Facultad de Ingeniería, Temuco, Chile.

^b Universidad del Bío Bío, Departamento de Ciencias Básicas, Facultad de Ciencias, Chillán, Chile.

^c Universidad de Magallanes, Departamento de Matemática, Facultad de Ciencias, Punta Arenas, Chile.

Resumen. En este artículo se considera el problema de la determinación de la función de flujo en una ley de conservación escalar que modela el fenómeno de sedimentación. Los datos de la observación experimental utilizada para la calibración corresponden a un perfil de la concentración de sólidos en un tiempo fijo. El problema de identificación se formula como uno de optimización, donde la función objetivo es la de mínimos cuadrados que minimiza la distancia entre los perfiles solución del modelo y la observación. La solución del problema directo es aproximada por un esquema de volúmenes finitos monótono. La solución numérica del problema de calibración se obtiene mediante un algoritmo genético continuo. Se presentan resultados numéricos para validar la eficiencia del algoritmo propuesto.

Palabras clave: Algoritmo genético continuo, calibración, identificación de parámetros, sedimentación.

MSC2010: 90C59, 90C30, 49M20, 35L50, 35L65.

Numerical solution of an inverse problem by applying a continuous genetic algorithm

Abstract. In this paper we consider the problem of flux determination in a scalar conservation law modeling the phenomenon of sedimentation. The experimental observation data used for the calibration consist of a solid concentration profile at a fixed time. The identification problem is formulated as an optimization one, where the distance between the profiles of the model simulation and observation data is minimized by a least squares cost function. The direct problem is approximated by a monotone finite volume scheme.

* E-mail: acoronel@ubiobio.cl

Recibido: 2 de octubre de 2017, Aceptado: 1 de agosto de 2018.

Para citar este artículo: S. Berres, A. Coronel, R. Lagos, Solución numérica de un problema inverso aplicando un algoritmo genético continuo, *Rev. Integr. temas mat.* 36 (2018), No. 2, 67–81. doi: 10.18273/revint.v36n2-2018001.

The numerical solution of the calibration problem is obtained by a continuous genetic algorithm. Numerical results are presented in order to validate the efficiency of the proposed algorithm.

Keywords: Continuous genetic algorithm, calibration, parameter identification, sedimentation.

1. *Introducción*

La sedimentación es un proceso mecánico utilizado para la separación de una mezcla de tipo sólido-fluido. Su gran utilidad en los procesos industriales la convierte en un fenómeno relevante para la investigación científica (ver [5]). Los lineamientos de la teoría de sedimentación para materiales incompresibles fueron establecidos inicialmente por Kynch [16]. Esta teoría introduce las condiciones bajo las cuales los procesos de sedimentación pueden ser modelados utilizando una ecuación de transporte no lineal unidimensional, o más precisamente una ley de conservación escalar. Estas hipótesis, a pesar de ser ideales, fundamentan de manera muy razonable la explicación del fenómeno. Actualmente, existen diversas extensiones de esta teoría; los principales aspectos históricos de la evolución de esta teoría aparecen detallados en [4], [5], [6].

La componente principal de una ley de conservación es la función flujo, la cual describe las propiedades del material utilizado dependiendo de la concentración de sólidos como una variable desconocida. En el problema de calibración del modelo se inicia con un perfil de concentración de sólidos en un tiempo fijo, bajo el supuesto de una forma paramétrica del flujo. La parametrización de la función flujo es una consecuencia de las ecuaciones constitutivas comprobadas empíricamente. El problema de calibración se formula como un problema de optimización, donde se busca un vector de parámetros que minimiza la distancia entre el perfil solución del modelo y el perfil observado.

El problema de calibración ha recibido una atención notable (ver [1], [2], [3], [20], [22], [24]). En [1], [2], [3] el problema de optimización fue resuelto numéricamente por un método de gradiente. En cambio, en [20], [22], [24] se aplican otras técnicas para la solución del problema inverso. La rigurosidad de la aproximación por el método de gradiente no ha sido tratada, porque se sabe que las soluciones entrópicas de las leyes de conservación escalares son funciones discontinuas, incluso bajo condiciones de alta regularidad en la condición inicial y la función de flujo. La baja regularidad de la solución de la ecuación de estado implica una baja regularidad en la función objetivo. En particular, la diferenciabilidad de la función objetivo es un asunto complicado.

En este artículo el problema de optimización se resuelve aplicando un algoritmo genético continuo. La función objetivo es discretizada utilizando la solución numérica del problema directo. La aproximación numérica natural del problema directo se obtiene mediante un método de volúmenes finitos, donde el flujo numérico es el de Engquist and Osher [7]. La función objetivo construida numéricamente no corresponde a un problema de optimización discreta. Por lo tanto, la aplicación de un algoritmo genético codificado de manera binaria conduce a un proceso de cálculo muy lento, y por ello se opta por utilizar un algoritmo genético codificado de manera continua (para mayores detalles consultar el libro de Haupt and Haupt [13] y también las referencias [10], [11], [14], [18]). El algoritmo propuesto para la calibración es aplicado a datos sintéticos. Este artículo es una versión

sucinta de uno más extenso, donde el comportamiento del algoritmo es puesto a prueba en varios ejemplos.

El presente artículo está organizado en cuatro secciones. En la Sección 2 se presentan las hipótesis de modelamiento de Kynch, se establece el modelo matemático para la sedimentación y se esboza el algoritmo genético continuo. En la Sección 3 se presentan los resultados numéricos de la identificación de parámetros. Finalmente, en la Sección 4 se presentan algunas conclusiones.

2. Teoría

2.1. Balance de masa y momentum en una mezcla

La fundamentos de la teoría de mezclas dentro de la mecánica del medio continuo se establecen en los trabajos de Fick, Stefan y Maxwell. Sin embargo, la formalización en el marco teórico conocido actualmente fue establecida por Truesdell. Esta teoría supone que cada punto del espacio puede ser ocupado por un número finito de partículas diferentes, una para cada componente del sistema. En consecuencia, la mezcla puede ser representada como la superposición de varios medios continuos, cada uno de los cuales mantiene su propio movimiento junto con las restricciones impuestas por las interacciones. En la teoría de mezclas cada componente satisface una ecuación de equilibrio. Si se denota por G_i una propiedad extensiva por unidad de masa de la i -ésima componente de la mezcla con densidad ρ_i , entonces

$$\frac{\partial}{\partial t}(\rho_i G_i) + \nabla \cdot J_i - r_i = 0, \quad (1)$$

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 dividido en su parte convectiva y difusiva para establecer la ecuación

$$\frac{\partial}{\partial t}(\rho_i G_i) + \nabla \cdot (\rho_i G_i \mathbf{v}) + \nabla \cdot J_D - r_i = 0, \quad (2)$$

donde \mathbf{v} es la velocidad promedio de la mezcla y J_D es el flujo de difusión. La ecuación de equilibrio (2) se puede utilizar para derivar balances de masa y momento en varias aplicaciones.

2.2. Modelo matemático para la sedimentación

El modelo matemático de la sedimentación describe una mezcla de partículas sólidas sumergidas en un fluido (ver Figura 1 para una ilustración del proceso). Se asume que la mezcla satisface las siguientes propiedades:

- (A1) Todas las partículas sólidas tienen el mismo tamaño, forma y densidad.
- (A2) El sólido y el fluido de la mezcla son incompresibles. No hay transferencia de masa entre sus componentes.

- (A3) La velocidad relativa sólido-líquido 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 líquido, que depende solamente de la concentración local de sólidos u .
- (A4) Los efectos de pared son despreciables.

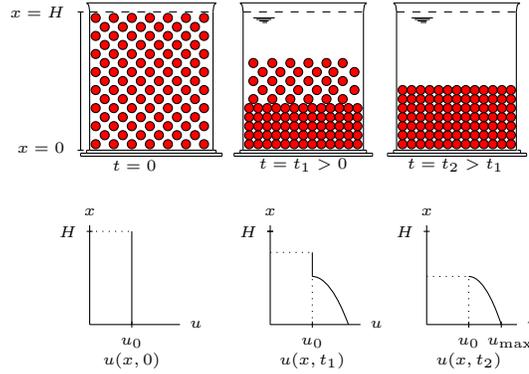


Figura 1. Columna y proceso de sedimentación.

Los balances de masa para la fase sólida y líquida son

$$u_t + \nabla \cdot (u\mathbf{v}_s) = 0, \quad (1-u)_t + \nabla \cdot ((1-u)\mathbf{v}_f) = 0, \quad (3)$$

respectivamente. Definiendo la velocidad volumétrica promedio $\mathbf{q}(\mathbf{x}, t) = u\mathbf{v}_s + (1-u)\mathbf{v}_f$ y sumando ambas ecuaciones se obtiene que $\nabla \cdot (\mathbf{q}(\mathbf{x}, t)) = 0$. A partir de las ecuaciones de balance de masa para las fases sólida y líquida se puede deducir que la ecuación diferencial del modelamiento de la sedimentación *bath* tiene la forma

$$u_t + (f_b(u))_x = 0, \quad x \in [0, H], \quad t > 0, \quad (4)$$

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

$$f_b(u) < 0, \quad u \in (0, u_{\max}), \quad f_b(0) = f_b(u_{\max}) = 0, \quad f'_b(0) < 0 \quad \text{y} \quad f'_b(u_{\max}) > 0. \quad (5)$$

En la práctica, esta restricción se impone de manera implícita mediante la elección de la forma paramétrica del flujo. En el ejemplo numérico se considera la función de flujo

$$f_b(u) = v_\infty u \left(1 - \frac{u}{u_{\max}} \right)^C, \quad (v_\infty, C) \in \mathbb{R}_- \times]1, \infty[. \quad (6)$$

Nótese que la negatividad $v_\infty < 0$ satisface la restricción $f_b(u) < 0$. Los tres parámetros identificados se establecen en el vector de parámetros

$$\mathbf{e} = (v_\infty, C, u_{\max}). \quad (7)$$

La ecuación (4) es hiperbólica no lineal de primer orden y debe ser complementada con condiciones iniciales y de frontera adecuadas. Bajo los supuestos de que la suspensión

tiene una concentración inicial homogénea u_0 y en la parte inferior del recipiente existe un incremento continuo pero rápidamente creciente de la concentración, iniciando desde u_0 hasta la concentración máxima u_{\max} , se tienen las siguientes condiciones inicial y de frontera:

$$u(x, 0) = u_0, \quad x \in [0, H], \quad u(0, t) = u_{\max}, \quad t > 0, \quad u(H, t) = 0, \quad t > 0, \quad (8)$$

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 (4) con condiciones inicial y de frontera dadas por (8) con la función de flujo satisfaciendo (6).

2.3. Problema inverso

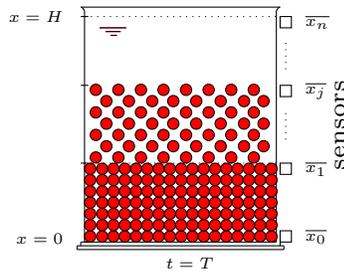


Figura 2. Forma esquemática para obtener la observación experimental.

Para la calibración del modelo (4),(5),(8) se dispone de un conjunto de datos experimentales medidos a lo largo de la columna en un tiempo fijo $t = T$ (ver Figura 2). Los n sensores son asociados a localizaciones específicas denotadas por x_i^{exp} . Los valores de la concentración de sólidos medidos en esos puntos son $\bar{\phi}_i$, lo cual significa que el conjunto de datos para la calibración del flujo es el siguiente:

$$\{(\bar{x}_i, \bar{\phi}_i) \in [0, H] \times [0, u_{\max}] : \bar{x}_0 = 0 < \bar{x}_1 < \dots < \bar{x}_{n-1} < \bar{x}_n = H\}.$$

En la práctica este conjunto de datos es ajustado por una curva, es decir se define $u^{\text{exp}} : [0, H] \rightarrow [0, u_{\max}]$ tal que $u^{\text{exp}}(\bar{x}_i) = \bar{\phi}_i$, que es denominado el perfil experimental. En términos de la información experimental el problema inverso de determinar la función de flujo es formulado en un ambiente abstracto como el problema de optimización

$$\begin{cases} \text{minimizar} & J(u, f_b) := \frac{1}{2} \int_0^H |(u - u^{\text{exp}})(x)|^2 dx, \\ \text{subjeto a} & E(u, p; f_b) = 0, \quad f \in U_{ad}, \end{cases} \quad (9)$$

donde U_{ad} es el conjunto admisible definido como

$$U_{ad} = \left\{ f_b : \mathbb{R} \rightarrow \mathbb{R} \mid f_b \in C^2(\mathbb{R}), f \text{ satisface (5)} \right\}$$

y $E(u, p; f_b)$ es la formulación débil del problema (4),(5),(8), más precisamente,

$$\begin{aligned} E(u, p; f_b) &= - \int_0^T \int_0^H \left\{ up_t + f_b(u)p_x \right\} dx dt \\ &+ \int_0^T \left\{ f_b(g_1(t))p(H, t) - f_b(g_0(t))p(0, t) \right\} dt \\ &- \int_0^H u_0(x)p(x, 0) dx, \quad \forall p \in C_0^1(Q_T), \end{aligned} \quad (10)$$

donde $g_0(t) = \phi(0, t) = 0$ y $g_1(t) = \phi(H, t) = u_{\text{máx}}$ son las soluciones en la frontera del intervalo $[0, H]$, y p es una función test [1]. La forma débil (10) es obtenida a partir de (4) multiplicando por la función test p , integrando sobre $[0, H] \times [0, T]$ y utilizando integración por partes.

El problema (9) se reduce al problema de identificación de parámetros, donde se supone que la función de flujo es parametrizada por el vector de parámetros $\mathbf{e} = (e_1, \dots, e_d) \in \mathbb{R}^d$, que significa que $f(\cdot) = f(\cdot; \mathbf{e})$. En consecuencia, la identificación de parámetros se formula como el siguiente problema de optimización en varias variables:

$$\left\{ \begin{array}{ll} \text{minimizar} & \mathcal{J}(\mathbf{e}), \quad \mathcal{J}(\mathbf{e}) := J(u(\mathbf{e}), f_b(u(\mathbf{e}); \mathbf{e})), \\ \text{sujeto a} & E(u(\mathbf{e}), p; f_b(u(\mathbf{e}); \mathbf{e})) = 0, \\ & \mathbf{e} \in D := \left\{ \mathbf{x} \in \mathbb{R}^d : f_b(\cdot, \mathbf{x}) \in U_{ad} \right\}. \end{array} \right. \quad (11)$$

2.4. Discretización de la función costo y del problema directo

La discretización de la función costo (11) es dada por

$$\min_{\mathbf{e} \in D} \mathcal{J}_\Delta(\mathbf{e}), \quad \mathcal{J}_\Delta(\mathbf{e}) := \frac{1}{2} \sum_{j=0}^M \left| u_j^N(\mathbf{e}) - u_j^{exp} \right|^2 \Delta x, \quad (12)$$

donde $u_j^N(\mathbf{e})$ para $j \in \{0, \dots, M\}$ es la solución numérica de (4),(5),(8) en $t = T$, y los valores de u_j^{exp} son los correspondientes datos experimentales calculados por promediar la función interpolada u^{exp} sobre los intervalos $K_j =]x_{j-1/2}, x_{j+1/2}[$:

$$u_j^{exp} = \frac{1}{\Delta x} \int_{K_j} u^{exp}(x) dx, \quad j = 0, \dots, M. \quad (13)$$

En efecto, en este trabajo, para determinar u_j^N se sigue lo detallado en [8], [17], [23] para la metodología de volúmenes finitos en leyes de conservación, y se considera el siguiente esquema de volúmenes finitos:

$$u_0^{n+1} = u_0^n - \lambda \left\{ g(u_0^n, u_1^n) - f_b(u_0^n) \right\}, \quad (14a)$$

$$u_j^{n+1} = u_j^n - \lambda \left\{ g(u_j^n, u_{j+1}^n) - g(u_{j-1}^n, u_j^n) \right\}, \quad (14b)$$

$$u_M^{n+1} = u_M^n - \lambda \left\{ f_b(u_M^n) - g(u_{M-1}^n, u_M^n) \right\}, \quad (14c)$$

definida para $n = 0, \dots, N$ y $j = 1, \dots, M - 1$, donde

$$\begin{aligned}\lambda &= \Delta t / \Delta x, \quad \Delta t = T / N, \quad \Delta x = H / (M + 1), \\ u_j^0 &= \frac{1}{\Delta x} \int_{K_j} u_0(x) dx, \quad K_j = (x_{j-1/2}, x_{j+1/2}), \\ x_{j+1/2} &= (x_j + x_{j+1}) / 2, \quad x_{j-1/2} = (x_{j-1} + x_j) / 2, \\ x_j &= j \Delta x, \quad x_{-1/2} = x_0, \quad x_{M+1/2} = x_M, \quad t_n = n \Delta t,\end{aligned}$$

y g es un flujo numérico de Engquist-Osher [7], es decir,

$$g(u, v) = \int_u^v |f'_b(s)| ds. \quad (15)$$

Además, por razones de estabilidad y convergencia del esquema numérico (14) es necesario considerar la condición de Courant-Friedrichs-Lewy (abreviadamente condición CFL)

$$\Delta t \leq \frac{\Delta x}{\max_{u \in [0, u_{\max}]} |f'(u)|},$$

la cual limita el tamaño del paso de tiempo para el esquema explícito.

2.5. Algoritmo genético

Las técnicas de computación evolutiva imitan los principios de la selección natural (o la supervivencia de los más aptos) y la evolución. La base de la computación evolutiva está en los siguientes cuatro paradigmas: algoritmos genéticos [14], programación genética [15], estrategias evolutivas [19] y programación evolutiva [9]. Entre estas técnicas, los algoritmos genéticos son los más populares porque son, computacionalmente, más fáciles de implementar. Complementariamente, bajo ciertas condiciones, proporcionan convergencia global y otras ventajas detalladas en [21].

En lo que sigue se utilizará la terminología estándar de algoritmos genéticos. Para completar, presentamos algunas definiciones básicas; cromosoma, gen, población y generación. Mayores detalles se pueden consultar [13], [21].

Definición 2.1. Un cromosoma es un arreglo de los parámetros que deben ser identificados, es decir, donde la función objetivo es evaluada.

Definición 2.2. Un gen es cada una de las componentes del vector de parámetros.

Definición 2.3. Una población es un conjunto de cromosomas.

Definición 2.4. Una generación es la población que existe al final de cada iteración del algoritmo genético.

Los algoritmos genéticos más utilizados seleccionan entre una representación binaria o una continua (punto flotante) de los parámetros a identificar. En este artículo se opta por una representación continua, principalmente, porque son más rápidos cuando la función objetivo y las variables son continuas. Mayores detalles respecto de ventajas y desventajas de ambas representaciones se pueden encontrar, por ejemplo, en [13].

En este artículo se considera el siguiente algoritmo genético:

- (a) **Problación inicial.** Se define la matriz que representa a la población aleatoria inicial $\mathbb{E} = [\mathbf{e}_1 | \mathbf{e}_2 | \cdots | \mathbf{e}_n]^T$, donde cada fila $\mathbf{e}_j^T \in \Omega := \prod_{i=1}^d [l_i, u_i] \subset D \subset \mathbb{R}^d$, con $l_i < u_i$ para $i = 1, \dots, d$. Sea G el número máximo de iteraciones o de generaciones. Se inicializa el contador $q \in \mathbb{Z}_0^+$ en $q = 0$ y comienza una iteración de los pasos siguientes:
- (b) **Costo de la población.** Se define el vector $\mathbf{cost} \in \mathbb{R}^n$ mediante la evaluación de la función objetivo para cada cromosoma de la población \mathbb{E} , es decir $\mathbf{J} := (\mathcal{J}_\Delta(\mathbf{e}_1), \dots, \mathcal{J}_\Delta(\mathbf{e}_n))^T \in \mathbb{R}^n$. Se define la matriz $\hat{\mathbb{E}} = [\mathbb{E} | \mathbf{J}]$. Dado J_{\min} como una tolerancia preestablecida para la evaluación de la función objetivo, si existe $\ell \in \{1, \dots, n\}$ tal que $\mathbf{J}_\ell \leq J_{\min}$, entonces la solución del problema de minimización discreto es aproximada por \mathbf{e}_ℓ y se detiene la iteración.
- (c) **Selección de los padres.** Se escogen los padres en tres etapas. Primero, se actualiza $\hat{\mathbb{E}}$ permutando sus columnas hasta satisfacer la propiedad $\hat{\mathbb{E}}_{1,d+1}^q \leq \hat{\mathbb{E}}_{2,d+1}^q \leq \dots \leq \hat{\mathbb{E}}_{n,d+1}^q$. Segundo, si $s \in (0, 1]$ denota la tasa de selección, entonces se seleccionan las primeras $\lceil ns \rceil$ columnas de $\hat{\mathbb{E}}$ y se almacena la submatriz en la denominada *mating pool matrix* \mathbb{F} . Aquí, $\lceil \cdot \rceil$ denota al mayor entero más cercano. Tercero, aplicando la regla de la ruleta, se seleccionan los padres desde los cromosomas de \mathbb{F} .
- (d) **Apareamiento.** Se define la regla algebraica para el cruzamiento de los padres. En este artículo, se obtienen los nuevos cromosomas aplicando una combinación convexa aleatoria de los genes de los padres seleccionados mediante un punto de cruzamiento. El proceso de apareamiento se detiene cuando $n - \lceil ns \rceil$ cromosomas son generados. Aquí, en este paso, la matriz de población \mathbb{E} se actualiza considerando que los padres se almacenan en las primeras $\lceil ns \rceil$ filas y los nuevos en las siguientes.
- (e) **Mutación.** Si $\mu \in [0, 1]$ denota la tasa de mutación, se define el número total de mutaciones como $m := \lceil \mu(n-1)d \rceil$. Se repite m veces el siguiente proceso de mutación: el gen aleatorio \mathbb{E}_{ij} se reemplaza por un número aleatorio perteneciente al intervalo $[l_j, u_j]$. Nótese que después de que finaliza la mutación, la matriz de población \mathbb{E} se encuentra actualizada. Se incrementa $q = q + 1$. Si $q \leq G$, entonces ir al ítem (b), si no, calcular el vector \mathbf{J} , y la solución es el cromosoma \mathbf{e}_ℓ tal que $\mathbf{J}_\ell \leq \mathbf{J}_j$ para $j \in \{1, \dots, n\}$.

El hipercono Ω considerado en el ítem (a) se puede reemplazar por un conjunto convexo tal que $\Omega \subset D$. La hipótesis de convexidad se requiere para la combinación convexa utilizada en el ítem (d).

3. Resultados de la identificación

En esta sección se presenta la aplicación del algoritmo genético continuo para la identificación de parámetros de la función flujo f_b . Para la identificación se utilizó la configuración

de parámetros sugerida por Haupt y Haupt en [13]:

| Parámetro | Valor |
|----------------------------|-------|
| Población (n) | 20 |
| Tasa de selección (s) | 0,5 |
| Tolerancia (J_{\min}) | 1e-6 |
| Tasa de mutación (μ) | 0,37 |
| Máximo de iteraciones | 20 |

Se consideran dos ejemplos. En el Ejemplo 1 se estudian algunas propiedades del algoritmo genético, y en el Ejemplo 2 se identifica una función de flujo con mayor complejidad.

3.1. Ejemplo 1

Se supone que los sensores están ubicados en $\bar{x}_i = \ell/20, i = 0, \dots, 20$. En el ejemplo se considera la función de flujo definida en (6), donde se identifican los parámetros v_∞ , u_{\max} y C , de acuerdo con (7). Para obtener los datos experimentales sintéticos se considera una simulación del problema directo con $T = 3500$, $u_0(x) = 0,05$, $M = 800$, $v_\infty = -2,7 \times 10^{-4}$, $u_{\max} = 0,5$ y $C = 5$. El conjunto de valores numéricos para la concentración que se obtiene mediante la simulación del problema directo son

$$\bar{\phi}_0 = 0,49759914, \quad \bar{\phi}_1 = 0,31229822, \quad \bar{\phi}_2 = 0,25193333, \quad (16)$$

$$\bar{\phi}_\ell = 0,05 \quad \text{para } \ell = 3, \dots, 9, \quad \bar{\phi}_\ell = 0 \quad \text{para } \ell = 10, \dots, 20. \quad (17)$$

Así se obtiene la siguiente función ajustada como perfil de observación en $T = 3500$:

$$u^{exp}(x) = \begin{cases} \frac{b^2 - \sqrt{b^2 - 4a(c - x/T)}}{2a} + d, & x \in [0; 0,1], \\ 0,05, & x \in [0,1; 0,45], \\ 0, & \text{en otro caso,} \end{cases} \quad (18)$$

con $a = 8,29285e - 4$, $b = 7,52058e - 4$, $c = 1,69437$ y $d = 4,40647e - 1$. Los valores numéricos de a, b, c, d se eligen de tal manera que la función (18) sea el mejor ajuste a los datos (16)-(17). Para el algoritmo genético se consideró

$$\Omega = [-5e - 4, -5e - 5] \times [2, 7] \times [0, 1, 0, 6]$$

como el rango de búsqueda para el vector de parámetros a identificar $\mathbf{e} = (v_\infty, C, u_{\max})$. En lo que sigue se reportan seis experimentos (desde (i) hasta (vi)) que muestran varias propiedades del algoritmo.

(i) Identificación de $\mathbf{e} = (v_\infty, C, u_{\max})$

En este ejemplo se identifica la función de flujo definida en (6). Para ser precisos se considera la identificación de $\mathbf{e} = (v_\infty, C, u_{\max})$ a partir de los datos experimentales dados en (18). Los resultados fueron obtenidos con una malla de tamaño espacial $M = 200$, el valor del vector de parámetros identificado es $\mathbf{e} = (-2,6e - 4; 4,910062; 0,483)$ y el valor de la función costo es $9,7161e - 5$. En la Figura 3-(a) se muestra la comparación del perfil obtenido con el mejor cromosoma y el perfil observado.

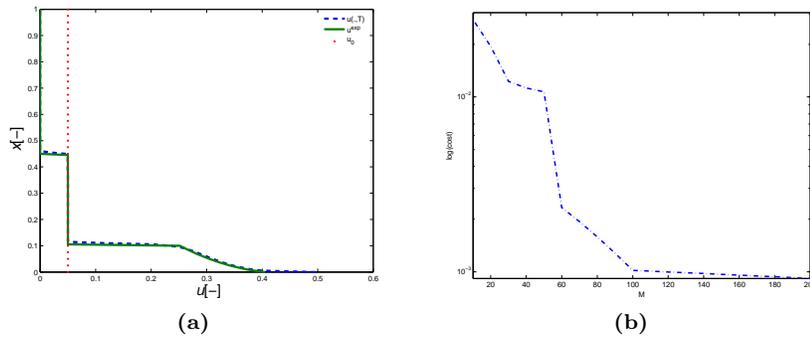


Figura 3. Experimentos numéricos (i) y (ii). En (a) se muestra el perfil identificado, la observación y la condición inicial, y en (b) la convergencia en términos del refinamiento.

(ii) Evolución de la función costo en términos de las generaciones

Se estudia la convergencia en términos de las generaciones con una longitud de espacio fija $M = 50$. Se encuentran los cromosomas de las primeras 100 iteraciones y se grafica la generación vs. la función de costo del mejor cromosoma en la Figura 3-(b).

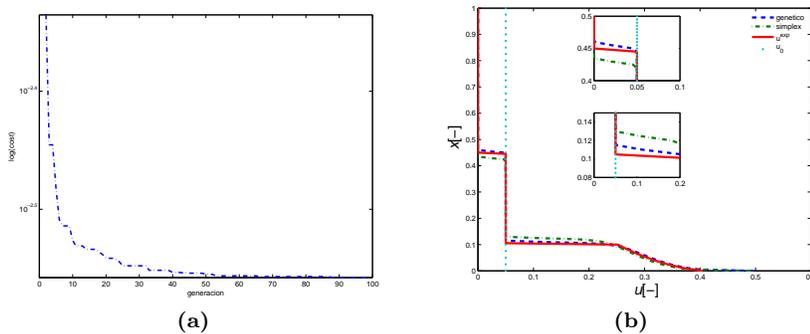


Figura 4. Experimentos numéricos (iii) y (iv). En (a) se muestra la evolución de la función costo en las distintas generaciones y en (b) la comparación de los perfiles identificados por el algoritmo genético y por el método simplex.

(iii) Convergencia con respecto al refinamiento de la malla

En este experimento se estudia la convergencia del algoritmo en términos del refinamiento. En este caso, se considera que el número total de iteraciones es fijo e igual a 200 y se desarrolla la identificación con tamaños de malla $M_i = 10i$ para $i \in \{1, \dots, 20\}$. Se encuentran los mejores cromosomas en términos del tamaño de malla y se grafica el tamaño de malla M_i vs. la evaluación de la función de costo de esos cromosomas. Los resultados se muestran en la Figura 4-(a).

(iv) Comparación con el método simplex

En este experimento se compara el algoritmo genético con el método de búsqueda llamado Símplex, que es implementado en la función `fminsearch` de Matlab. La selección

del método Simplex es hecha al seguir las sugerencias de Hansen y colaboradores [12]. Para el método simplex el punto inicial es $(-2, 3400e - 4; 4, 4191; 4, 6)$. Otros puntos iniciales también fueron considerados, pero la convergencia fue muy lenta y la ejecución fue abortada cuando el tiempo cpu superó las 11 horas. Los valores de la función costo obtenidas por el método simplex y el algoritmo genético, después de 50 iteraciones, son $1, 94322e - 4$ y $6, 43343e - 4$, respectivamente. Los valores correspondientes del tiempo cpu fueron de 8,56 horas en el caso del algoritmo genético, y en el caso del simplex de 9,32 horas. La comparación de los perfiles se muestra en la Figura 4-(b). De hecho, en las pruebas, el método simplex es muy lento cuando el punto inicial se escoge lejos del óptimo.

(v) Tiempo cpu del algoritmo.

En este experimento numérico se consideran medidas del tiempo CPU. Las simulaciones fueron hechas utilizando Matlab R(2010a) en un computador personal Dell Vostro 3500 con las siguientes propiedades: Intel Core I5 M520 con un procesador 2.40Ghz, 4Gb de RAM, 500 Gb de disco duro con 720 rpm, y con Windows 7 como sistema operativo. La identificación con el algoritmo genético es hecho con un cantidad de pasos de espacio $M = 30$ y con diferentes tamaños de población $n = 10i$ para $i = 1, \dots, 10$. El tiempo CPU es medido en generación 10 y 15. Los resultados se muestran en la Figura 5-(a).

(vi) Identificación con ruido

En este ejemplo se muestra la identificación con el algoritmo genético en la presencia de un ruido en la observación. Se consideró que la observación es perturbada con 0,5 % de ruido aleatoriamente distribuido. Los resultados fueron obtenidos con una malla de tamaño espacial $M = 200$. El valor del vector de parámetros identificados para la observación sin el ruido es $\mathbf{e} = (-2, 6e - 4; 4, 910062; 0, 483)$, y el valor de la función costo es $9, 7161e - 5$. El valor del vector de parámetros identificados para la observación con el ruido es $\mathbf{e} = (-2, 91e - 4; 5, 501051; 0, 494)$ y el valor de la función costo es $3, 5463e - 04$. En la Figura 3-(b) se muestra la comparación del perfil obtenido para el mejor cromosoma con ruido, el perfil para el mejor cromosoma sin ruido y el perfil observado perturbado.

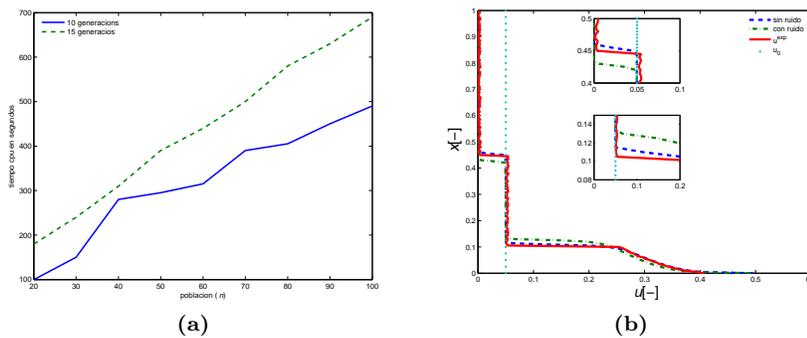


Figura 5. Experimentos numéricos (v)-(vi). En (a) se muestra la medida del tipo CPU en segundos y en (b) la comparación de los perfiles identificados para la observación sin ruido y la observación con ruido.

| i | \bar{x}_i | ϕ_i | i | \bar{x}_i | ϕ_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 |

Tabla 1. Ubicación de los sensores y valores de la concentración observada para el Ejemplo 2.

3.2. Ejemplo 2

En este ejemplo la función de flujo que se identificará es definida por

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

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 $\mathbf{e} = (v_0, v_1, C)$. Para obtener el perfil observado se consideró una simulación del problema directo con $T = 235$, $M = 200$, el flujo y la condición inicial definidas por

$$f_b(u) = \begin{cases} 3,38433u - 13,7672u^2 + 162275u^3 \\ \quad + 1,1264u^4 - 9,02235u^5, & u \in (0, u_{\text{máx}}), \\ 0, & \text{en otro caso,} \end{cases}$$

$$u_0(x) = \begin{cases} 0,34, & x \in]0,08; 1[, \\ 0, & \text{en otro caso.} \end{cases}$$

Se supone que los sensores están ubicados en $\bar{x}_i = \ell/20$, $i = 0, \dots, 20$. El conjunto de valores para la concentración que se obtiene mediante la simulación del problema directo se muestra en la Tabla 1. Luego, utilizando un ajuste de los datos se obtiene como perfil de observación en $T = 235$, el cual es dado por la siguiente función:

$$u^{\text{exp}}(x) = \begin{cases} ax^2 + bx + c, & x \in [0; 0,85[, \\ 0, & \text{en otro caso,} \end{cases}$$

donde $a = 0,1796$, $b = 0,08979$ y $c = -0,002797$.

Para el algoritmo genético se consideró: $\Omega = [-5e - 3, -1e - 3] \times [1e - 4, 5e - 4] \times [1.1, 4]$. Los resultados de la identificación con $M = 100$ se muestran en la Figura 6, y los valores 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 $1,8238e - 4$.

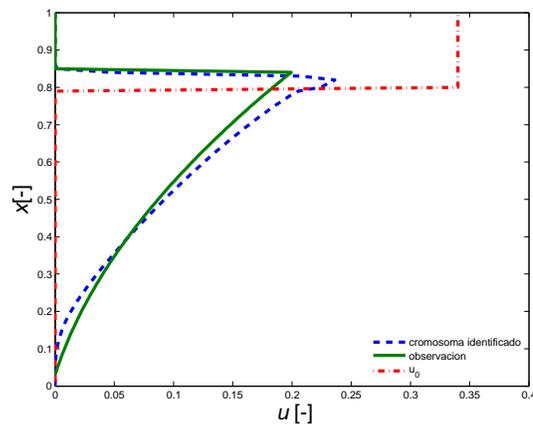


Figura 6. Perfiles para la condición inicial, el cromosoma identificado y observado en el Ejemplo 2.

4. Conclusiones

En esta contribución se ha validado exitosamente un algoritmo genético continuo para resolver numéricamente el problema de calibración de una función de flujo en una ley de conservación que modela el fenómeno físico de la sedimentación de partículas sólidas inmersas en un fluido, donde la función de flujo satisface las consideraciones físicas introducidas por Kynch en [16]. La distancia entre perfiles observados e identificados permite concluir que el algoritmo genético resuelve adecuadamente el problema de identificación de parámetros para leyes de conservación escalares no convexas.

Además, se observa que este nuevo algoritmo supera las conocidas desventajas de los métodos de tipo gradiente aplicadas en [1], [2], [3], entre las cuales se destacan; (a) El algoritmo genético elige aleatoriamente el punto de partida para la iteración, y (b) El algoritmo genético no necesita ninguna hipótesis de diferenciabilidad de la función objetivo.

Finalmente, el algoritmo genético de identificación considerado resultó ser estable a pesar de los datos de observación sintéticos utilizados. El problema inverso estudiado es sobredeterminado, ya que, en el ejemplo numérico, así como en situaciones prácticas relevantes, existen más datos de observación (aquí 20, en aplicaciones industriales hasta 1000) que parámetros libres para ajustarse (normalmente nunca mayores de 5).

Agradecimientos

Los autores agradecen al revisor anónimo que observó aspectos débiles y permitió la mejora del manuscrito. AC agradece a los proyectos de investigación DIUBB 172409 GI/C, DIUBB 103309 4/R y Fondo de Apoyo a la Participación a Eventos Internacionales (FAPEI) de la Universidad del Bío-Bío de Chile. RL agradece el apoyo de los proyectos de investigación PY-F1-01MF16 y PY-02-MATFIS-18 de la Universidad de Magallanes de Chile.

Referencias

- [1] Berres S., Bürger R., Coronel A. and Sepúlveda M., “Numerical identification of parameters for a strongly degenerate convection-diffusion problem modelling centrifugation of flocculated suspensions”, *Appl. Numer. Math.* 52 (2005), No. 4, 311–337.
- [2] Bürger R., Coronel A. and Sepúlveda M., “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*, Springer Verlag, Providence, RI (2008), 225–232.
- [3] Bürger R., Coronel A. and Sepúlveda M., “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.*, Amer. Math. Soc., Providence, RI (2009), 445–454.
- [4] Bürger R. and Wendland W. L., “Sedimentation and suspension flows: historical perspective and some recent developments”, *J. Engrg. Math.* 41 (2001), No. 2-3, 101–116.
- [5] Bustos M. C., Concha F., Bürger R. and Tory E. M., *Sedimentation and thickening*, vol 8 of *Mathematical Modelling: Theory and Applications*, Kluwer Academic Publishers, Dordrecht, 1999.
- [6] Concha F. and Bürger R., “Thickening in the 20th century: a historical perspective”, *Minerals & Metallurgical Process* 20 (2003), No. 2, 57–67.
- [7] Engquist B. and Osher S., “One-sided difference approximations for nonlinear conservation laws”, *Math. Comput.* 36 (1981), 321–351.
- [8] Eymard R., Gallouët T. and Herbin R., “Finite volume methods”, in *Handbook of numerical analysis*, Volume VII: Handb. Numer. Anal. North-Holland, Amsterdam, (2000), 713–1020.
- [9] Fogel L.J., Owens A.J. and Walsh M.J., *Artificial intelligence through simulated evolution*, Wiley, Chichester, WS, UK, 1966.
- [10] Gaffney J., Pearce C. and Green D., “Binary versus real coding for genetic algorithms: A false dichotomy?”, *ANZIAM J.* 51 (EMAC2009), C347–C359, 2010.
- [11] Goldberg D.E., *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1st edition, 1989.
- [12] Hansen N., Auger A., Ros R., Finck S. and Pošík P., “Comparing results of 31 algorithms from the black-box optimization benchmarking bbob-2009”, in *Proceedings of the 12th Annual Conference Companion on Genetic and Evolutionary Computation*, GECCO '10, New York, NY, USA, (2010), 1689–1696.
- [13] Haupt R.L. and Haupt S.E., *Practical genetic algorithms*, Wiley-Interscience [John Wiley & Sons], Hoboken, NJ, second edition, 2004.
- [14] Holland J.H., *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, MI, USA, 1975.
- [15] Koza J.R., *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, MIT Press, Cambridge, MA, USA, 1992.
- [16] Kynch G.J., “A theory of sedimentation”, *Trans. Faraday Soc* 48 (1952), 166–176.

- [17] LeVeque R.J., *Numerical methods for conservation laws*, Lectures in Mathematics ETH Zürich, Second edition, Basel: Birkhäuser Verlag, 1992.
- [18] Michalewicz Z., *Genetic algorithms + data structures = evolution programs*, Artificial Intelligence, Springer-Verlag, Berlin, 1996.
- [19] Rechenberg I., *Evolutionstrategie: Optimierung Technischer Systeme nach Prinzipien der Biologischen Evolution*, Fromman-Holzboog Verlag, Stuttgart, 1973.
- [20] Sweeney M.W. and Kabouris J.C., “Modeling, Instrumentation, Automation, and Optimization of Water Resource Recovery Facilities”, *Water Environment Research* 88 (2016), 1279–1298.
- [21] Sivanandam S. N. and Deepa S. N., *Introduction to genetic algorithms*, Springer, Berlin, 2008.
- [22] Tan Ch. K., Bao J. and Bickert G., “A study on model predictive control in paste thickeners with rake torque constraint ”, *Mineral Engineering* 105 (2017), 52–62.
- [23] Toro E.F., *Riemann solvers and numerical methods for fluid dynamics: A practical introduction*, Third edition, Springer-Verlag, Berlin, 2009.
- [24] Xu N., Wang X., Zhou J., Wang Q., Feng W. and Peng X., “An intelligent control strategy for thickening process”, *International Journal of Mineral Processing* 142 (2015), 56–62.