



Calibración numérica de leyes de conservación escalar vía un algoritmo genético continuo.

A continuous genetic algorithm for the numerical calibration in scalar conservation laws.

Stefan Berres*, Aníbal Coronel[†], and Richard Lagos[‡]

Received, Jan. 05, 2017

Accepted, May. 15, 2017

DOI: <http://dx.doi.org/10.17268/sel.mat.2017.01.02>

Resumen

En este artículo se investiga sobre el problema de la identificación de parámetros del flujo en una ley de conservación escalar. El problema es formulado como un problema de optimización, donde la función objetivo compara la solución del problema directo con el perfil observado en un tiempo fijo. Un esquema de volúmenes finitos resuelve el problema directo y un algoritmo genético continuo resuelve el problema inverso. El método numérico es verificado con datos experimentales sintéticos. Los parámetros simulados se recuperan aproximadamente. La técnica de optimización heurística probada resulta ser más robusta que las técnicas clásicas de optimización.

Palabras clave. Volumen finito, algoritmo genético, identificación del flujo, ley de conservación.

Abstract

Our work deals with the flux identification problem for scalar conservation laws. The problem is formulated as an optimization problem, where the objective function compares the solution of the direct problem with observed profiles at a fixed time. A finite volume scheme solves the direct problem and a continuous genetic algorithm solves the inverse problem. The numerical method is tested with synthetic experimental data. Simulation parameters are recovered approximately. The tested heuristic optimization technique turns out to be more robust than classical optimization techniques.

Keywords. Finite volume, genetic algorithm, flux identification, conservation law.

1. Introducción. El problema de la identificación del flujo en leyes de conservación escalares ha sido extensivamente estudiado en la literatura especializada porque tiene relevancia en varios campos de las ciencias e ingeniería. Por ejemplo; la identificación del flujo en problemas de sedimentación relacionados con el tratamiento de aguas residuales [3], el flujo bifásico a través de un medio poroso está relacionado con las funciones de permeabilidad relativa y la determinación de éstas no cuenta con una teoría debida a principios fundamentales de la física y por ello es y será una fuente de motivación de problemas de éste tipo [11], la identificación de parámetros también aparece como una herramienta útil en el modelamiento matemático del tráfico vehicular [7].

Frecuentemente, en las aplicaciones a ingeniería donde aparece la necesidad de usar éstas herramientas, las condiciones iniciales y de contorno se estiman vía mediciones empíricas. Las relaciones constitutivas

*Departamento de Ciencias Matemáticas, Facultad de Ingeniería, Universidad Católica de Temuco, Rudecindo Ortega 02950, Temuco-Chile. sberres@uct.cl.

[†]Departamento de Ciencias Básicas, Facultad de Ciencias, Universidad del Bío Bío, Avda. Andrés Bello 720, Chillán-Chile. acoronel@ubiobio.cl.

[‡]Departamento de Matemática y Física, Facultad de Ciencias, Universidad de Magallanes, Av. Bulnes 01855, Punta Arenas-Chile. richard.lagos@umag.cl.

This work is licensed under the [Creative Commons Attribution-NoComercial-ShareAlike 4.0](https://creativecommons.org/licenses/by-nc-sa/4.0/).

están determinadas por la información disponible a partir de datos experimentales. Así, las ecuaciones constitutivas dependen de parámetros desconocidos que no son accesibles desde procedimientos experimentales. En los casos en que sean accesibles, se necesitarían pruebas de laboratorio sofisticadas y costosas para determinarlas experimentalmente.

En el contexto de la calibración del flujo, consideramos un problema inverso donde el perfil de observación es dado en un tiempo fijo. El objetivo es determinar el flujo de manera que la solución entrópica del problema de valor inicial de una ley de conservación es lo más cercana posible a los datos observados. Las técnicas de optimización permiten obtener una solución a este problema inverso, donde la función objetivo cuantifica la diferencia entre los datos observados y las soluciones simuladas.

En general, la solución de una ley de conservación no lineal depende de manera no lineal de los parámetros. Independientemente de la regularidad de los coeficientes y las condiciones iniciales y de contorno dicha solución desarrolla discontinuidades en tiempo finito [2]. En la mayoría de los casos, este tipo de comportamiento del problema directo implica que la función objetivo resulta ser una función no convexa y no diferenciable, Figura(4.2). Así, el análisis del problema inverso y la elección de los métodos adecuados deben considerar la capacidad de minimizar funciones no convexas y no diferenciables.

En el excelente trabajo realizado por Rocca, Benedetti, Donelli, Franceschini y Massa [16] se muestra una visión unificada de los algoritmos evolutivos como métodos de optimización aplicados a resolver un problema inverso de dispersión. En particular, se sugiere que los algoritmos genéticos usualmente funcionan bien en situaciones en las que los métodos del tipo gradiente resultan fallar en la convergencia local debido a la no diferenciable de la función objetivo.

En el presente artículo se muestran dos ejemplos numéricos con el objetivo de investigar respecto de la viabilidad y aplicabilidad de un algoritmo genético continuo como método de optimización para resolver un problema inverso en leyes de conservación escalares, otros ejemplos se pueden ver en [1]. Aunque los resultados son preliminares, podemos adelantar las siguientes consideraciones que sirven de guía para un desarrollo posterior; los parámetros identificados por el algoritmo representan buenas aproximaciones a sus valores numéricos verdaderos y el óptimo numérico de la función objetivo depende del orden del método de volúmenes finito utilizado para la discretización del problema directo.

Bajo las consideraciones anteriores, creemos que la utilización de un método numérico de alta resolución, para resolver el problema directo, contribuye positivamente en el desempeño de la identificación del flujo numérico en leyes de conservación. Los ejemplos numéricos dados en este artículo sugieren que el uso de un algoritmo genético es una poderosa herramienta numérica para la identificación de parámetros en las leyes de conservación.

Sin embargo, antes de la aplicación de la técnica que se ha descrito a situaciones de mayor complejidad que sean de interés teórico o de ingeniería, se deben mejorar algunos aspectos, por ejemplo: considerar algoritmos híbridos que puedan combinar automáticamente el algoritmo genético continuo con otras estrategias evolutivas e incluso con algoritmos de tipo determinísticos.

2. Problemas directo e inverso continuo y discretización. En esta sección, después de dar una descripción breve de los problemas directo e inverso, se define el problema de identificación de parámetros, la discretización del problema directo y la función objetivo discreta.

2.1. El problema de la identificación de parámetros. El problema directo viene dado por el siguiente problema de valor inicial y de frontera

$$(2.1) \quad u_t + (f(u))_x = 0, \quad (x, t) \in Q_T := \mathcal{I} \times \mathcal{T},$$

$$(2.2) \quad u(x, 0) = u_0(x), \quad x \in \mathcal{I} := (0, 1),$$

$$(2.3) \quad u(\ell, t) = g_\ell(t), \quad \ell \in \{0, 1\}, \quad t \in \mathcal{T} := [0, T],$$

donde t denota al tiempo, x la variable espacial, u la variable de estado, f la función flujo, u_0 la condición inicial, y g_ℓ , $\ell \in \{0, 1\}$ las condiciones de frontera del tipo Dirichlet. Normalmente, se asume que u_0 , f , g_0 y g_1 son funciones que están dadas y se requiere determinar $u(\cdot, T)$ para un tiempo finito $T > 0$. Sin embargo, el flujo f a veces es desconocido y se puede determinar resolviendo un problema inverso. Dicho problema inverso es conocido como problema de calibración, donde se considera un conjunto de experimentos para tener un problema sobredeterminado. En particular, si se asume que $\hat{u}(x)$ es un perfil experimental dado en un tiempo fijo $t = T$, entonces el problema inverso de la identificación del flujo se puede formular como el siguiente problema de optimización

$$(2.4) \quad \begin{array}{ll} \text{minimizar} & J(u) := \frac{1}{2} \int_{\mathcal{I}} |(u - \hat{u})(x)|^2 dx, \\ \text{sujeto a:} & E(u, p; f) = 0, \quad f \in \mathcal{U}_{\text{ad}}, \end{array}$$

donde la restricción $E(u, p; f)$ es la formulación débil del problema directo(2.1)–(2.3),

$$E(u, p; f) = - \int \int_{Q_T} \{ u p_t + f(u) p_x \} dx dt + \int_{\mathcal{T}} \{ f(g_1(t)) p(1, t) - f(g_0(t)) p(0, t) \} dt - \int_{\mathcal{I}} u_0(x) p(x, 0) dx,$$

para todo $p \in C_0^1(Q_T)$ y \mathcal{U}_{ad} es el conjunto de los flujos admisibles definido como

$$\mathcal{U}_{\text{ad}} = \left\{ f : \mathbb{R} \rightarrow \mathbb{R} \mid f \in C^2(\mathbb{R}), f(0) = f'(0) = 0, f''(u) > 0 \quad \forall u \in I_{\text{máx}} \right\},$$

con el intervalo máximo $I_{\text{máx}}$ definido como

$$I_{\text{máx}} = \left[\text{mín}(A), \text{máx}(A) \right] \quad \text{con} \quad A = \left\{ \|u_0\|_{L^\infty(I)}, \|g_1\|_{L^\infty(\mathcal{T})}, \|g_2\|_{L^\infty(\mathcal{T})} \right\}.$$

Para el problema de identificación de parámetros, se asume que la función flujo depende de un número finito de parámetros, denotados por $\mathbf{e} = (e_1, \dots, e_d) \in \mathbb{R}^d$: es decir, $f(\cdot) = f(\cdot; \mathbf{e})$. Usualmente, la dependencia paramétrica analítica del flujo f sobre el conjunto de parámetros \mathbf{e} se deriva de una relación constitutiva entre una velocidad y una densidad. En particular, en el caso del modelo del flujo del tráfico, se supone que el flujo está definido por $f(u) = uv(u)$, donde v es la velocidad. Por ejemplo, en el modelo para el tráfico de Lighthill-Whitham, la velocidad está dada por $v(u) = v_* (1 - u/\rho^*)$, tal que v_* y ρ^* son los parámetros para la calibración ([12, 13]). Por lo tanto, la formulación general del problema de optimización (2.4) se reduce a un problema de optimización con respecto a d parámetros como sigue

$$(2.5) \quad \begin{cases} \text{minimizar} & \mathcal{J}(\mathbf{e}) := J(u(\mathbf{e})), \\ \text{sujeto a:} & E(u(\mathbf{e}), p; f(u(\mathbf{e}); \mathbf{e})) = 0, \\ & \mathbf{e} \in D := \{ \mathbf{x} \in \mathbb{R}^d : f(u; \mathbf{x}) \in \mathcal{U}_{\text{ad}} \text{ para toda } u \in I_{\text{máx}} \}. \end{cases}$$

Los elementos de $D \subset \mathbb{R}^d$ son aquellos parámetros admisibles sobre el cual el algoritmo de optimización buscará los parámetros óptimos.

2.2. Solución numérica del problema directo. Una descripción más detallada del método numérico que se emplea para determinar una aproximación de la solución del problema directo se puede encontrar en [4, 18]. Sin embargo, a modo de esbozo, el método de volúmenes finitos considera una partición del dominio espacial \mathcal{I} en M subintervalos K_j de largo $\Delta x = 1/M$ centrados en los nodos $x_j = j\Delta x$, para $j = 0, \dots, M$. Se definen los intervalos por $K_j := (x_{j-1/2}, x_{j+1/2})$, donde $x_{j\pm 1/2} = x_j \pm \Delta x/2$ se refiere a la posición entre los nodos. Aquí, $x_{j+1/2} = (x_j + x_{j+1})/2$ para $j = 0, \dots, M-1$, $x_{-1/2} = x_0 - \Delta x/2$ y $x_{M+1/2} = x_M + \Delta x/2$. Los conjuntos K_j son llamados células o volúmenes de control y sus fronteras son llamadas interfaces. Análogamente, el dominio temporal \mathcal{T} se particiona en N subintervalos de largo $\Delta t = T/N$ definido por $R_n = [t_n, t_{n+1})$, donde $t_n = n\Delta t$, para $n = 0, \dots, N$. Para simplificar la notación se usa $Q_j^n := K_j \times R_n$. La solución numérica de (2.1)–(2.3) sobre Q_j^n se denota por u_j^n . Con esta notación, se discretizan las ecuaciones (2.1)–(2.3).

En efecto, se comienza con la discretización de la condición inicial (2.2) estableciendo

$$(2.6) \quad u_j^0 = \frac{1}{\Delta x} \int_{K_j} u_0(x) dx, \quad j = 0, \dots, M.$$

Luego, siguiendo las ideas de la técnica de los volúmenes finitos, se integran las ecuaciones (2.1)–(2.3) sobre Q_j^n , se denota el radio de la malla por $\lambda = \Delta t/\Delta x$, y se deduce el esquema numérico siguiente

$$(2.7) \quad u_j^{n+1} = u_j^n - \lambda \left\{ f_{j+1/2}^n - f_{j-1/2}^n \right\}, \quad f_{j+1/2}^n \approx \frac{1}{\Delta t} \int_{R_n} f(u(x_{j+1/2}, t)) dt,$$

para todo $n = 0, \dots, N$ y $j = 0, \dots, M$, donde $f_{j+1/2}^n$, para un trazado (stencil) con $2p+1$ puntos, depende de los valores en los nodos como $f_{j+1/2}^n = g(u_{j-p}^n, \dots, u_{j+p}^n)$. Se asume que la función de flujo numérico $g \in \text{Lip}(\mathbb{R}^{2p}, \mathbb{R})$ es Lipschitz continua y asegura la consistencia del esquema de volúmenes finitos (2.7) con la ecuación diferencial discretizada satisfaciendo $g(u, \dots, u) = f(u)$. En las simulaciones numéricas de este artículo, se consideró la función de flujo numérico de Godunov [12], definida explícitamente para

$f \in \mathcal{U}_{ad}$ por

$$(2.8) \quad g^{\text{Godunov}}(u, v) := \begin{cases} f(u) & \text{para } u, v > 0, \\ f(v) & \text{para } u, v < 0, \\ \min \{f(u), f(v)\} & \text{para } u \leq 0 \leq v, \\ \max \{f(u), f(v)\} & \text{para } v \leq 0 \leq u. \end{cases}$$

Naturalmente, el esquema (2.7) se escribe como sigue

$$(2.9) \quad u_0^{n+1} = u_0^n - \lambda \{g^{\text{Godunov}}(u_0^n, u_1^n) - f(u_0^n)\},$$

$$(2.10) \quad u_j^{n+1} = u_j^n - \lambda \{g^{\text{Godunov}}(u_j^n, u_{j+1}^n) - g^{\text{Godunov}}(u_{j-1}^n, u_j^n)\},$$

$$(2.11) \quad u_M^{n+1} = u_M^n - \lambda \{f(u_M^n) - g^{\text{Godunov}}(u_{M-1}^n, u_M^n)\}.$$

Para la estabilidad y convergencia del esquema Godunov (2.9)–(2.11), se asegura que la condición CFL

$$(2.12) \quad \lambda \cdot \max \{|f'(u)| : u \in I_{\max}\} \leq 1/2$$

se satisface.

2.3. El problema de la identificación de parámetros discreto. La observación continua \hat{u} se discretiza mediante

$$(2.13) \quad \hat{u}_j = \frac{1}{\Delta x} \int_{K_j} \hat{u}(x) dx, \quad j = 0, \dots, M.$$

La discretización natural del problema de la identificación de parámetros (2.5) es

$$(2.14) \quad \begin{cases} \text{minimizar} & \mathcal{J}_\Delta(\mathbf{e}), \quad \mathcal{J}_\Delta(\mathbf{e}) := \frac{1}{2} \sum_{j=0}^M |u_j^N(\mathbf{e}) - \hat{u}_j|^2 \Delta x, \\ \text{sujeto a:} & u_j^N(\mathbf{e}) \text{ obtenida en (2.9)–(2.11) para } \mathbf{e} \in D. \end{cases}$$

Notar que la restricción continua

$$E(u(\mathbf{e}), p; f(u(\mathbf{e}); \mathbf{e})) = 0$$

se sustituye por la restricción discreta que $u_j^N(\mathbf{e})$ sea la solución numérica (2.1–2.3) que se obtiene del esquema de volúmenes finitos (2.9)–(2.11). La consistencia entre ambas restricciones se debe a la convergencia del método de volúmenes finitos.

3. Algoritmo genético continuo para la estimación del flujo. Las técnicas de computación evolutiva imitan los principios de La selección natural (o la supervivencia de los más aptos) y evolución. La base de la computación evolutiva está en los siguientes cuatro paradigmas;

- (A) algoritmos genéticos [8],
- (B) programación genética [9],
- (C) estrategias evolutivas [15] y
- (D) programación evolutiva [5].

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 [17].

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 [6, 17].

Definición 1. *Un cromosoma es un arreglo de los parámetros que deben ser identificados, es decir, donde la función objetivo es evaluada.*

Definición 2. *Un gen es cada una de las componentes del vector de parámetros.*

Definición 3. *Una población es un conjunto de cromosomas.*

Definición 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 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 [6].

En este artículo se considera el siguiente algoritmo genético:

- (a) **Población 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 $\mathbf{J} := (\mathcal{J}_\Delta(\mathbf{e}_1), \dots, \mathcal{J}_\Delta(\mathbf{e}_n))^T \in \mathbb{R}^n$. Para la evaluación de la función objetivo, $u_j^N(\mathbf{e})$ se calcula según las ecuaciones (2.9), (2.10) y (2.11). 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 de (2.14) es \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 vía 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 $[l_j, u_j]$. Notar que después 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), sino 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 hipercubo 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).

4. Resultados numéricos. En esta sección, consideramos dos ejemplos de la aplicación del algoritmo genético continuo utilizado para la identificación del flujo en leyes de conservación. En ambos ejemplos, el algoritmo genético se ejecuta con la siguiente configuración; tamaño de población inicial $n = 20$, tasa de selección $s = 0,5$ y tolerancia $J_{\min} = 1,0 \times 10^{-6}$ para la evaluación de la función objetivo. La tasa de mutación para el Ejemplo 1 es $\mu = 0,2$ y $\mu = 0,37$ para el Ejemplo 2. El número máximo de iteraciones para los Ejemplos 1 y 2 es $\text{iter} = 100$.

Ésta configuración en los valores de los parámetros para la ejecución del algoritmo genético es la que sugieren Haupt y Haupt [6]. Cabe señalar que en el libro se presenta un conjunto de pruebas que permiten identificar que los parámetros asociados a la población y tasa de mutación son los más cruciales, mientras que la tasa de cruzamiento, el método de selección y el tipo de cruzamiento son menos relevantes para el rendimiento del algoritmo. La versión del algoritmo genético que se utiliza "funciona mejor (optimiza más rápidamente) con tamaños de población pequeños y tasas de mutación relativamente altas. A medida que aumenta el tamaño de la población, la tasa de mutación óptima disminuye". Una validación propia de los parámetros numéricos sugeridos confirmó que los valores elegidos parecen ofrecer una buena tasa de convergencia.

Con respecto del método numérico para el problema directo, el número máximo de pasos de tiempo en el método de volúmenes finito para los dos ejemplos se selecciona mediante la siguiente relación

$$N = \left\lceil T(0,45\Delta x)^{-1} \max_{u \in I_{\max}} |f'(u)| \right\rceil,$$

donde el intervalo I_{\max} se detalla en la descripción de cada ejemplo.

4.1. Ejemplo 1: Identificación de un único parámetro (d=1). En este ejemplo, se considera una función de flujo (2.1) del tipo Burgers, $f(u) = u^\alpha/\alpha$, y condición inicial y frontera dadas por

$$u_0(x) = \begin{cases} 0, & (4x - 1)(4x - 3) > 0, \\ 1, & \text{en otro caso,} \end{cases} \quad g_0(t) = g_1(t) = 0 \quad \forall t > 0.$$

Notar que $f''(u) = (\alpha - 1)u^{\alpha-2}$, lo cual implica que f pertenece al conjunto admisible \mathcal{U}_{ad} cuando $\alpha - 1 > 0$, puesto que $I_{m\acute{a}x} = [0, 1]$. Así el conjunto restricción en (2.5) se define por $D = (1, \infty)$.

Se consideran dos observaciones analíticas obtenidas con $\alpha = 2$ en $T_1 = 1/4$ y $T_2 = 1/2$, denotadas por \hat{u}_1 y \hat{u}_2 , respectivamente. Mediante el método de las características, se deduce que $\hat{u}_i : I \rightarrow \mathbb{R}$, para $i = 1, 2$, están definidas como

$$\hat{u}_1(x) = u(x, 1/4) = \begin{cases} 0, & (4x - 1)(8x - 7) > 0, \\ 4x - 1, & (4x - 1)(2x - 1) \leq 0, \\ 1, & \text{en otro caso,} \end{cases}$$

$$\hat{u}_2(x) = u(x, 1/2) = \begin{cases} 0, & 4x - 1 < 0, \\ 2x - 1/2, & (4x - 1)(4x - 3) \leq 0, \\ 1, & \text{en otro caso.} \end{cases}$$

Las funciones objetivo están dadas por

$$(4.1) \quad \mathcal{J}_1(\alpha) = \frac{1}{8} \left(\frac{\alpha - 1}{\alpha + 1} - \frac{2(\alpha - 1)}{2\alpha - 1} + \frac{1}{3} \right) + \frac{\alpha - 2}{16\alpha} \text{sgn}(\alpha - 2),$$

$$(4.2) \quad \mathcal{J}_2(\alpha) = \frac{1}{4} \left(\frac{\alpha - 1}{\alpha + 1} - \frac{2(\alpha - 1)}{2\alpha - 1} + \frac{1}{3} \right) + \frac{\alpha - 2}{16\alpha} (1 + \text{sgn}(\alpha - 2)),$$

donde sgn denota a la función signo.

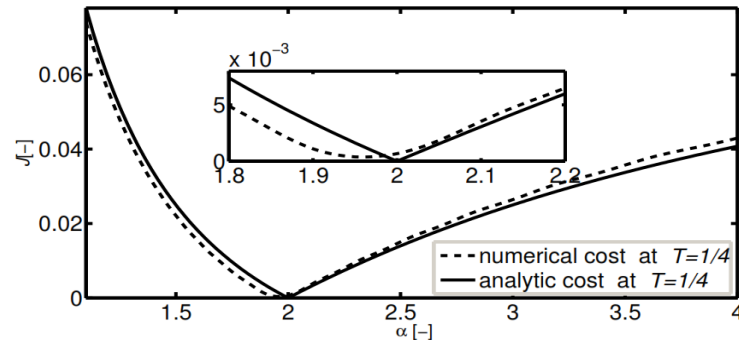


FIGURA 4.1. Función objetivo analítica (línea sólida) y función objetivo numérica (línea punteada)

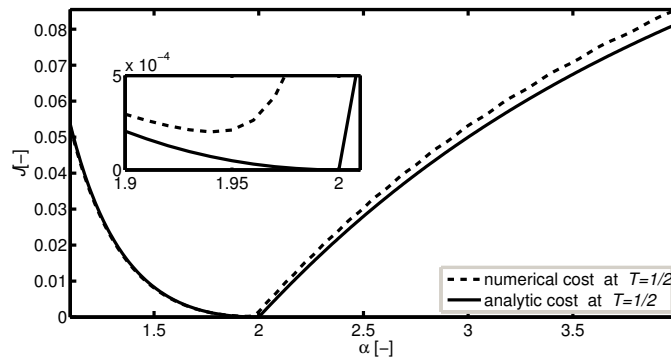


FIGURA 4.2. Función objetivo analítica (línea sólida) y Función objetivo numérica (línea punteada).

Para el algoritmo genético, se considera $\Omega = [1, 1, 4] \subset D$. Los gráficos de las funciones objetivo analítica (línea sólida) y numérica (línea punteada) sobre Ω se muestran en la Figuras (4.1) y (4.2). En

ambos gráficos se observa que el óptimo de las funciones objetivos analítica y numérica no coinciden. Éste comportamiento es una consecuencia natural del método numérico utilizado para la simulación del problema directo, puesto que los métodos de volúmenes finitos de primer orden tienen una gran difusión numérica, [4, 18].

CUADRO 4.1

El mejor cromosoma de las primeras 11 generaciones y su correspondiente evaluación en la función objetivo.

q	Observación \hat{u}_1		Observación \hat{u}_2	
	α	cost	α	cost
0	1,9375	0,00046090	1,7865	0,00106196
1	1,9375	0,00046090	1,9605	0,00026737
2	1,9375	0,00046090	1,9110	0,00025926
3	1,9675	0,00037939	1,9383	0,00020224
4	1,9575	0,00036810	1,9415	0,00020118
5	1,9575	0,00036810	1,9415	0,00020118
6	1,9612	0,00036794	1,9415	0,00020118
7	1,9594	0,00036738	1,9415	0,00020118
8	1,9594	0,00036738	1,9415	0,00020118
9	1,9594	0,00036738	1,9415	0,00020118
10	1,9594	0,00036738	1,9415	0,00020118

Para la simulación del problema directo, se usó $M = 100$ intervalos de espacio y un número variable de pasos de tiempo satisfaciendo la condición CFL.

Los mejores cromosomas de cada iteración se muestran en la Tabla 4.1. Aquí, notamos una rápida convergencia del algoritmo genético implementado. El número máximo de generaciones y la tolerancia considerados fueron 1.0×10^2 y 1.0×10^{-6} , respectivamente. Sin embargo, el algoritmo encontró un óptimo aceptable con cuatro generaciones y se detuvo cuando se alcanzó el criterio de parada dado por la tolerancia preestablecida, la iteración número 10.

CUADRO 4.2

Comparación de la identificación vía el algoritmo genético y el método simplex Nelder-Mead como función del paso en la discretización espacial (M). J_{ga} y J_{sNM} son los valores numéricos de la función objetivo en el punto de convergencia del algoritmo genético y del método simplex Nelder-Mead, respectivamente.

M	J_{sNM}	J_{ga}	$J_{sNM} - J_{ga}$
10	1.1421E-2	1.1421E-2	2.7347E-10
20	6.8933E-3	6.8933E-3	1.3056E-08
30	3.7897E-3	3.7897E-3	-5.6871E-12
40	4.0919E-3	4.0918E-3	1.3905E-07
50	2.4407E-3	2.4407E-3	8.4222E-10
60	1.9710E-3	1.9710E-3	-1.1750E-10
70	1.5684E-3	1.5684E-3	9.4597E-10
80	1.6682E-3	1.6682E-3	-1.5565E-11
90	1.2927E-3	1.2927E-3	4.4827E-12
100	1.1096E-3	1.1068E-3	2.7854E-06

En la Tabla 4.2 se presenta una comparación entre la identificación del parámetro mediante el algoritmo genético y el método simplex Nelder-Mead [10] vía la función predefinida `fminsearch` de Matlab. Se consideró la función observación \hat{u}_1 y varios pasos temporales (M). Para el método simplex Nelder-Mead se consideró como aproximación inicial a 1,1. Adicionalmente, se puede observar que los valores numéricos del óptimo global de la función objetivo encontrado por ambos métodos coinciden, en general, hasta la cuarta cifra significativa. Sin embargo, al chequear, el resto de la cifra, en cada resultado, se observa que el valor numérico de J_{ga} es inferior J_{sNM} en el 70 % de los casos, lo que demuestra que el algoritmo genético supera al método simplex de Nelder-Mead en la tarea de minimizar incluso en la situación en que la aproximación inicial se elige cerca del óptimo para una función objetivo convexa. Tal comparación sería aún más a favor del algoritmo genético en situaciones generales, donde la función de objetivo pierde convexidad y el óptimo no se conoce de antemano para garantizar la cercanía de la aproximación inicial.

4.2. Ejemplo 2: Identificación de un flujo del tráfico vehicular ($d=2$). La función básica usada como modelo del flujo del tráfico vehicular es del tipo siguiente: $\hat{f}(u) = v_* u (1 - u/\rho^*)$, la cual es una

función cóncava para v_* , y ρ^* en \mathbb{R}^+ , es decir $\hat{f} \notin \mathcal{U}_{ad}$. Se sabe que el análisis de funciones de flujo cóncavo es completamente análogo al caso de funciones de flujo convexo ([2, 12] para detalles). Con el objetivo de realizar el análisis de la función de flujo vehicular mediante la teoría de las funciones de flujo convexa, se redefine el flujo como $\hat{f}_1(u) = -\hat{f}(u) + \hat{f}'(0)u + \hat{f}(0)$, entonces se tiene que $\hat{f}_1(u) = -v_*u^2/\rho^* \in \mathcal{U}_{ad}$ (ver [14]).

En este ejemplo, se considera un flujo ligeramente más general que $\hat{f}_1(u) = -v_*u^2/\rho^*$, a saber,

$$f(u) = \alpha u^\beta, \quad u_0(x) = \begin{cases} 2, & x \in [0, 1/3), \\ 3, & x \in [1/3, 2/3), \\ 1, & x \in [2/3, 1], \end{cases} \quad g_0(t) = 0, \quad \text{y} \quad g_1(t) = 1.$$

Notar que $f \in \mathcal{U}_{ad}$ si $\alpha\beta(\beta - 1) > 0$, puesto que $I_{m\acute{a}x} = [0, 3]$. Así, el conjunto de restricciones definido en (2.5) es

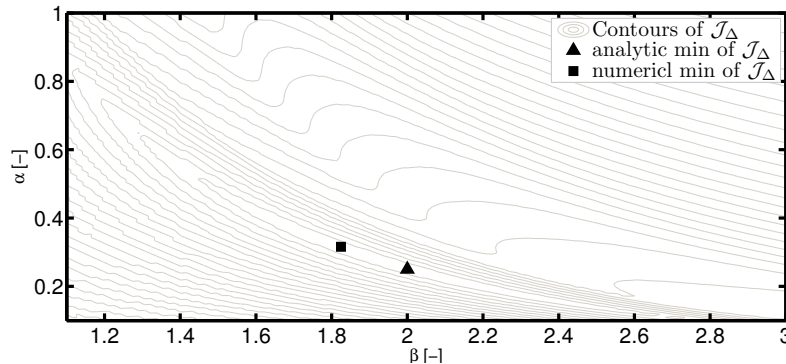
$$D = \left\{ \mathbf{e} = (\alpha, \beta) \in \mathbb{R}^2 : \alpha\beta(\beta - 1) > 0, \quad f(\cdot; \mathbf{e}) \in \mathcal{U}_{ad} \right\}.$$

Ahora, se considera una observación analítica obtenida con $\alpha = 0,25$ y $\beta = 2$ en $T = 1/4$. Por el método de las características se deduce que $\hat{u} : I \rightarrow \mathbb{R}$ se define como

$$\hat{u}(x) := u(x, 1/4) = \begin{cases} 8x, & x(4x - 1) \leq 0, \\ 2, & (4x - 1)(12x - 7) < 0, \\ 8x - 8/3, & (12x - 7)(24x - 17) \leq 0, \\ 3, & (24x - 17)(11x - 12) < 0, \\ 1, & (11x - 12)(x - 1) \leq 0. \end{cases}$$

Para el algoritmo genético se elige $\Omega = [0,1, 3] \times [1,1, 4] \subset D$. El gráfico de la función objetivo sobre Ω se muestra en la Figura 4.3. Al igual que en el ejemplo anterior, se observa que la difusión numérica del método de Godunov implica que el óptimo para la función objetivo numérica es levemente diferente del óptimo de la función objetivo analítica.

FIGURA 4.3. Curvas de nivel de la función objetivo discreta $\mathcal{J}_\Delta : \Omega \rightarrow \mathbb{R}$ con $M = 200$ intervalos espaciales.



5. Conclusiones. El problema de la identificación de parámetros para las leyes de conservación escalar se formula como un problema de optimización y se resuelve mediante un algoritmo genético de tipo continuo. El algoritmo genético continuo se ha aplicado a dos identificaciones de parámetros. La concordancia entre los parámetros observados y los parámetros identificados por el algoritmo genético continuo es de alta calidad. Además, se observó que el óptimo de la función objetivo numérica depende de la discretización del problema directo. Por lo tanto, creemos que la aplicación del algoritmo genético continuo con un método numérico de alta resolución, para determinar la aproximación a la solución del problema directo, contribuye de manera natural a la mejora en la identificación del flujo numérico en leyes de conservación escalares.

La identificación del flujo es importante, porque la calibración de modelos matemáticos basados en leyes de conservación permite una mejor comprensión de los procesos físicos inherentes de la situación en estudio. Por lo tanto, se necesitan métodos robustos de optimización. Los métodos clásicos tienen una tasa de convergencia baja, o incluso a veces no convergen. Los métodos heurísticos son una alternativa alentadora.

Agradecimientos. El autor S. Berres agradece el apoyo de Conicyt (Chile) a través del proyecto Fondecyt 1120587. El autor A. Coronel agradece el apoyo de los proyectos de investigación DIUBB GI 153209/C y DIUBB GI 153109/EF de la Universidad del Bío-Bío de Chile. El autor R. Lagos reconoce y agradece el apoyo del proyecto de investigación PY-F1-01MF16 de la Universidad de Magallanes de Chile.

Referencias

- [1] S. Berres, A. Coronel, R. Lagos and M. Sepúlveda. Performance of a real coded genetic algorithm the calibration of scalar conservation laws. *Anziam J.*, 58:51–77, 2016.
- [2] C.M. Dafermos. *Hyperbolic Conservation Laws in Continuum Physics*, volume 325 of Grundlehren der Mathematischen Wissenschaften (Fundamental Principles of Mathematical Sciences). Springer-Verlag, Berlin, third edition, 2010.
- [3] J. De Clerq, I. Nopens, J. Defrancq and Pa. Vanrolleghem. Extending and calibrating a mechanistic hindered and compression settling model for activated sludge using in-depth batch experiments. *Water Research*, 42(3):781–791, 2008.
- [4] R. Eymard, T. Gallouët and R. Herbin. *Finite Volume Methods*. In Handbook of numerical analysis, Vol. VII, Handb. Numer. Anal., VII, pages 713–1020. North-Holland, Amsterdam, 2000.
- [5] L.J. Fogel, A.J. Owens and M.J. Walsh. *Artificial Intelligence Through Simulated Evolution*. Wiley, Chichester, WS, UK, 1966.
- [6] R. L. Haupt and S. E. Haupt. *Practical Genetic Algorithms*. Wiley-Interscience [John Wiley & Sons], Hoboken, NJ, second edition, 2004.
- [7] H. Holden, F.S. Priuli, and N.H. Risebro. On an inverse problem for scalar conservation laws. *Inverse Problems*, 30(3):035015, 2014.
- [8] J.H. Holland. *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor, MI, USA, 1975.
- [9] J.R. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, Cambridge, MA, USA, 1992.
- [10] J.C. Lagarias, J. A. Reeds, M. H. Wright, and P. E. Wright, Convergence Properties of the Nelder-Mead Simplex Method in Low Dimensions, *SIAM Journal of Optimization*, 9(1):112–147, 1998.
- [11] R. Lagos. *Estudio analítico y numérico de un problema inverso originado en la extracción secundaria de petróleo*. Tesis de Magíster, Universidad del Bío-Bío, 2015.
- [12] R.J. LeVeque. *Numerical methods for conservation laws*. Lectures in Mathematics ETH Zürich. Birkhäuser Verlag, Basel, second edition, 1992.
- [13] M. J. Lighthill and G. B. Whitham. On kinematic waves. II. A theory of traffic flow on long crowded roads. *Proc. Roy. Soc. London. Ser. A.*, 229:317–345, 1955.
- [14] H. Liu and T. Pan. Interaction of elementary waves for scalar conservation laws on a bounded domain. *Math. Methods Appl. Sci.*, 26(7):619–632, 2003.
- [15] I. Rechenberg. *Evolutionstrategie: Optimierung Technischer Systeme nach Prinzipien des Biologischen Evolution*. Frommann-Holzboog Verlag, Stuttgart, 1973.
- [16] P. Rocca, M. Benedetti, M. Donelli, D. Franceschini and A. Massa. Evolutionary optimization as applied to inverse scattering problems. *Inverse Problems*, 25(12):123003, 2009.
- [17] S.N. Sivanandam and S. N. Deepa. *Introduction to Genetic Algorithms*. Springer, Berlin, 2008.
- [18] E.F. Toro. *Riemann Solvers and Numerical Methods for Fluid Dynamics : A practical introduction*. Springer-Verlag, Berlin, third edition, 2009.