



UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE INGENIERÍA ELÉCTRICA

OPTIMIZACIÓN MULTIOBJETIVO DE UN MODELO FENOMENOLÓGICO PARA EL
EMPAQUETAMIENTO DE BATERÍAS DE LITIO MEDIANTE PROGRAMACIÓN
GENÉTICA

MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL ELÉCTRICO

NICOLÁS MANUEL AGUILAR AGUILERA

PROFESOR GUÍA:
PABLO ESTÉVEZ VALENCIA
PROFESOR CO-GUÍA:
JORGE VERGARA QUEZADA
COMISIÓN:
WILLIAMS CALDERÓN MUÑOZ

SANTIAGO DE CHILE
2019

RESUMEN DE LA MEMORIA PARA OPTAR
AL TÍTULO DE INGENIERO CIVIL ELÉCTRICO
POR: NICOLÁS MANUEL AGUILAR AGUILERA
FECHA: 2019
PROF. GUÍA: PABLO ESTÉVEZ VALENCIA

OPTIMIZACIÓN MULTIOBJETIVO DE UN MODELO FENOMENOLÓGICO PARA EL EMPAQUETAMIENTO DE BATERÍAS DE LITIO MEDIANTE PROGRAMACIÓN GENÉTICA

Actualmente las baterías de ion-litio son utilizadas en una gran variedad de aplicaciones, tales como: dispositivos electrónicos, herramientas, vehículos eléctricos y sistemas de almacenamiento de energía. El masivo uso de estas baterías ha despertado el interés de optimizar el diseño de su empaquetamiento, siendo relevantes factores como: costos, temperaturas máximas, área, ventilación, etc. Una forma de abordar este problema es evaluando una gran cantidad de configuraciones, mediante simulaciones en software de multi-física Computational Fluid Dynamics (CFD). Sin embargo, una simulación toma varios minutos, por lo que probar una gran cantidad de configuraciones requiere de mucho tiempo. Una alternativa es formular un modelo fenomenológico, el cual permite obtener una aproximación de la simulación del CFD en un tiempo mucho menor a éste.

En la memoria de título de Francisco Villa los ajustes al modelo fenomenológico se realizaron mediante programación genética en un contexto de optimización multiobjetivo. En esta memoria se propone continuar este trabajo, analizando la validez de las soluciones encontradas, aumentando el número de celdas estudiadas y buscando nuevas soluciones que se ajusten de mejor manera a las condiciones del nuevo modelo.

Para lograr lo anterior, se construyó una base de datos de un modelo de 102 celdas con distintas configuraciones utilizando el software CFD ANSYS. Las estradas son: la corriente en las celdas, la separación entre celdas, el diámetro de las celdas, el flujo de aire entrante y la temperatura ambiente. Posteriormente, se utilizó programación genética multiobjetivo para evolucionar y encontrar simultáneamente funciones de los siguientes parámetros de interés del modelo fenomenológico: coeficiente de arrastre, factor de fricción y número de Nusselt. Con esta metodología se pudo construir un frente de Pareto en la optimización del modelo fenomenológico, ajustando la velocidad y presión de fluido y la temperatura de las celdas centrales.

Se obtuvieron conjuntos de funciones que representan de mejor manera el comportamiento de los parámetros analizados, logrando un mayor ajuste del modelo fenomenológico en relación a las desarrolladas en trabajos anteriores. En referencia al modelo original, se pudo disminuir el RMSE global de $3,9361[m/s]$, $215,1223[Pa]$ y $7,4035[K]$ a $0,954[m/s]$, $64,4[Pa]$ y $6,5051[K]$ con el modelo más general encontrado sobre la configuración de 102 celdas. Además, se observó que los modelos encontrados pierden validez al extrapolar a un mayor número de celdas, siendo necesario realizar nuevos ajustes.

Tabla de Contenido

1. Introducción	1
1.1. Objetivos	2
1.1.1. Objetivos específicos	2
2. Antecedentes	3
2.1. Baterías de ion-litio	3
2.2. Computational Fluid Dynamics (CFD)	4
2.3. Modelo fenomenológico	4
2.3.1. Ecuaciones	4
2.3.2. Cálculo de variables de salida	5
2.3.3. Parámetros de interés	6
2.4. Programación genética	8
2.4.1. Control de bloat	11
2.5. Optimización multiobjetivo	11
2.6. Trabajo anterior	12
3. Metodología	13
3.1. Obtención de Datos	13
3.1.1. Modelo ANSYS	13
3.1.2. Simulaciones	13
3.2. Modelo fenomenológico	15
3.3. Evaluación de expresiones encontradas en trabajos anteriores	15
3.4. Programación genética	17
3.4.1. Fitness	17
3.4.2. Parámetros	18
3.4.3. Variación de parámetros	19
3.4.4. Comparación con frentes de Pareto obtenidos con expresiones anteriores	19
3.4.5. Variación en cálculo de Nusselt	20
3.4.6. Configuraciones de 53 y 151 celdas	20
4. Análisis y resultados	21
4.1. Evaluación de modelos anteriores	21
4.2. Programación genética	24
4.2.1. Adición de nuevo terminal	24
4.2.2. Variación de parámetros	26
4.2.3. Comparación de frentes obtenidos con expresiones anteriores	29

4.2.4. Variación en forma del número de Nusselt	34
4.2.5. Validez de modelos sobre configuraciones de 53 y 151 celdas	37
Conclusión	39
Bibliografía	41
A. Árboles de individuos estudiados	44
A.1. $best(Vf)$	44
A.2. $best(Pf)$	46
A.3. $best(Tc)$	47
A.4. $best(prod)$	49
A.5. $best(prod)_2$	51
A.6. $best(prod)_3$	52

Índice de Tablas

2.1. Densidad del aire en función de la temperatura	6
2.2. Factor de corrección del número de Nusselt en función de la columna de fluido evaluada	8
3.1. $A(S)$ y $B(S)$ en función de la separación entre celdas para el modelo fenomenológico original.	16
4.1. Error de los individuos elegidos para los tres conjuntos de datos.	31

Índice de Ilustraciones

2.1. Diseño de un banco de baterías. Fuente [1]	3
2.2. Diagrama de flujo convencional de algoritmos genéticos. Fuente [2]	9
2.3. Ejemplo de frente de Pareto. Fuente [3]	12
3.1. Ejemplo de resultado de simulación en ANSYS.	14
3.2. Lugares de "medición" de salidas en ANSYS.	14
3.3. Esquema banco de baterías. Fuente [4]	15
4.1. RMSE y ME por columna para los modelos de Reyes, Villa y Garcia-Tapia.	22
4.2. Histogramas de errores globales para los modelos de Reyes, Villa y Garcia-Tapia para el caso de 102 celdas.	22
4.3. Salidas de modelos vs salida ANSYS en bloques extremos (14 para V_f y T_c , 1 para P_f) para los modelos de Reyes, Villa y Garcia-Tapia en el conjunto de prueba.	23
4.4. Salidas de modelos vs separación entre celdas en bloque 10 de la configuración.	24
4.5. Frentes de Pareto original (azul) y cono índice de columna (naranja), para el conjunto de entrenamiento.	25
4.6. Frentes de Pareto original (azul) y con índice de columna (naranja), calculados sobre el conjunto de validación.	25
4.7. Promedio del tamaño de los árboles en función del número de generación para cada una de las 3 funciones objetivo, variando α y el tamaño máximo permitido.	26
4.8. Frentes de Pareto en el conjunto de entrenamiento con diferentes α y tamaño de árboles máximo.	27
4.9. Frentes de Pareto en conjunto de validación con diferentes α y tamaño de árboles máximo.	28
4.10. Frentes de Pareto obtenidos en este trabajo versus modelos anteriores en conjunto de validación.	29
4.11. RMSE y ME por columna para los puntos elegidos sobre el conjunto de prueba.	32
4.12. Histogramas de errores globales para los individuos elegidos sobre el conjunto de prueba.	32
4.13. Salidas predichas vs salidas ANSYS en bloque extremos para los individuos elegidos, sobre conjunto de prueba.	33
4.14. Frentes de Pareto variando la forma de calcular el número de Nusselt.	35
4.15. RMSE y ME por columna para los puntos $best(prod)$, $best(prod)_2$ y $best(prod)_3$ sobre el conjunto de prueba.	36
4.16. Frentes de Pareto sobre base de datos de 53 celdas.	37

4.17. Frentes de Pareto sobre base de datos de 151 celdas.	38
4.18. RMSE y ME por columna para el individuo $best(prod)$ sobre bases de datos de 53, 102 y 151 celdas.	39
A.1. Árbol de coeficiente de arrastre, individuo $best(Vf)$	44
A.2. Árbol de factor de fricción, individuo $best(Vf)$	44
A.3. Árbol de Nusselt, individuo $best(Vf)$	45
A.4. Árbol de coeficiente de arrastre, individuo $best(Pf)$	46
A.5. Árbol de factor de fricción, individuo $best(Pf)$	46
A.6. Árbol de Nusselt, individuo $best(Pf)$	46
A.7. Árbol de coeficiente de arrastre, individuo $best(Tc)$	47
A.8. Árbol de factor de fricción, individuo $best(Tc)$	47
A.9. Árbol de Nusselt, individuo $best(Tc)$	48
A.10.Árbol de coeficiente de arrastre, individuo $best(prod)$	49
A.11.Árbol de factor de fricción, individuo $best(prod)$	49
A.12.Árbol de Nusselt, individuo $best(prod)$	50
A.13.Árbol de coeficiente de arrastre, individuo $best(prod)_2$	51
A.14.Árbol de factor de fricción, individuo $best(prod)_2$	51
A.15.Árbol de Nusselt, individuo $best(prod)_2$	51
A.16.Árbol de coeficiente de arrastre, individuo $best(prod)_3$	52
A.17.Árbol de factor de fricción, individuo $best(prod)_3$	52
A.18.Árbol de Nusselt, individuo $best(prod)_3$	53

Capítulo 1

Introducción

Actualmente las baterías de ion-litio son usadas en una gran variedad de aplicaciones, tales como dispositivos electrónicos portátiles, herramientas eléctricas, vehículos híbridos/eléctricos y sistemas de almacenamiento de energía. Esto se debe a sus favorables características como alta densidad energética y bajo efecto memoria, que hace que su desempeño sea superior a otros tipos de baterías en las aplicaciones dadas. Debido a lo anterior surge el interés de optimizar su diseño.

En esta memoria se aborda el problema del empaquetamiento de baterías, en el cual se deben optimizar varios factores simultáneamente tales como: costo, temperaturas máximas, área, entre otros. Esto hace que el problema de empaquetamiento sea un problema multiobjetivo de gran interés.

El problema de empaquetamiento se modela con ecuaciones parciales, que son simuladas en un software multi-física CFD (Computational Fluid Dynamics). Sin embargo, los tiempos de simulación son excesivamente altos, motivo por el cual se ha formulado un modelo fenomenológico que permite obtener en principio soluciones rápidas para las diferentes configuraciones. Sin embargo, este modelo entrega errores importantes en algunas variables de salida con respecto al software CFD, lo que puede deberse a estimaciones de coeficientes no necesariamente correctas en todas las situaciones. Debido a esto, el modelo requiere ajustarse para el problema de diseño del empaquetamiento óptimo de baterías de litio.

El ajuste del modelo fenomenológico se realiza optimizando simultáneamente, a través de programación genética, las ecuaciones del coeficiente de arrastre, coeficiente de fricción y el número de Nusselt, las cuales son usadas en el cálculo de las salidas del modelo, correspondientes a: velocidad de flujo, presión de fluido y temperatura de las celdas centrales. Se utiliza como estándar dorado las simulaciones en el software de multi-física ANSYS CFD.

El problema descrito anteriormente fue abordado en forma simplificada en la memoria de título de Francisco Villa[1], donde se consideraron solo 5 celdas y además se tuvo que descomponer el problema de optimización en tres partes separadas debido a las limitaciones del software ocupado (MATLAB). Posteriormente el código fue portado a Java, disminuyendo notablemente el tiempo necesario por el algoritmo de programación genética y permitiendo

optimizar las 3 ecuaciones al mismo tiempo, destacando los aportes del trabajo de Germán García y Nicolás Tapia[4].

En esta memoria se propone expandir la configuración original al caso real de decenas de celdas, analizando la validez de las soluciones encontradas sobre la nueva configuración y considerando la optimización simultánea del coeficiente de arrastre, factor de fricción y número de Nusselt.

1.1. Objetivos

El objetivo general de esta memoria es la optimización de un modelo fenomenológico de bancos de celdas de litio utilizando programación genética multiobjetivo.

1.1.1. Objetivos específicos

1. Construir una base de datos con diferentes configuraciones de hasta 102 celdas simuladas en el software CFD ANSYS.
2. Diseñar una metodología para lograr un modelo secuencial de n bloques.
3. Ajustar las ecuaciones del coeficiente de arrastre, el factor de fricción y el número de Nusselt del modelo fenomenológico mediante programación genética multiobjetivo.
4. Analizar las soluciones encontradas (frente de Pareto) para las 3 ecuaciones incluyendo rango de validez, expresión mínima y sentido físico.

Capítulo 2

Antecedentes

2.1. Baterías de ion-litio

Las baterías de ion-litio son un tipo de baterías recargables usadas en una gran variedad de aplicaciones, tales como dispositivos electrónicos portables, herramientas eléctricas, vehículos híbridos/eléctricos y sistemas de almacenamiento de energía. Sus favorables características hacen que sean consideradas como una tecnología clave de muchos sistemas eléctricos presentes y futuros [5], por lo que son de gran interés para la industria y la investigación.

Los estándares de baterías de litio distinguen distintos niveles de organización. Las unidades más básicas son llamadas “celdas”, las cuales pueden agruparse para formar módulos para producir voltajes medios. Una agrupación de módulos a su vez es llamada “pack” o baterías [6]. Estos son empaquetados junto con controladores eléctricos, térmicos y mecánicos, formando un banco de baterías. Las etapas del diseño pueden observarse en la figura 2.1.

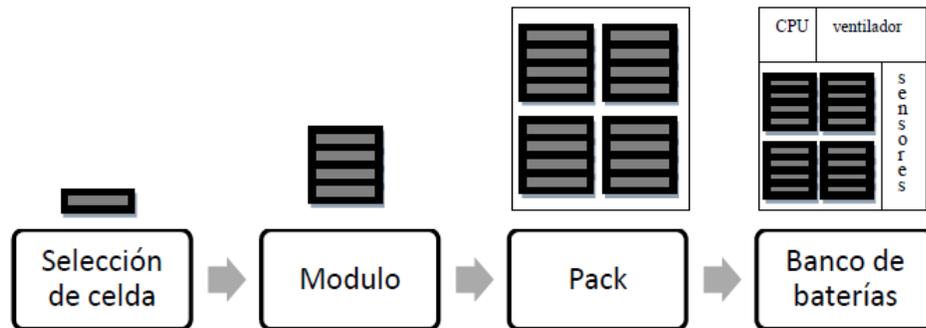


Figura 2.1: Diseño de un banco de baterías. Fuente [1]

En el diseño inicial de un pack de baterías es necesario considerar una variedad de aspectos, tales como seguridad, costo del empaquetamiento del sistema, vida útil y durabilidad, costo de manufactura, consideraciones de mantenimiento, manejo de temperaturas, componentes electrónicos para la supervisión y control, capacidad, tamaño, entre otros [6]. Esto hace que en el problema de la optimización del empaquetamiento de baterías se deban optimizar

simultáneamente varios criterios, algunos de los cuales son opuestos entre ellos, por lo que el problema se trabaja como una optimización multiobjetivo, teniendo de esta forma no una sino un conjunto de soluciones posibles.

2.2. Computational Fluid Dynamics (CFD)

Computational Fluid Dynamics es una rama de la mecánica de fluidos que se basa en resolver los aspectos físicos del flujo de fluidos, que se rigen por los siguientes principios fundamentales: conservación de masa, de momentum y de energía. Estos principios pueden expresarse en términos de ecuaciones matemáticas, que en su forma general son ecuaciones integrales o diferenciales parciales. Computational Fluid Dynamics busca resolver estas ecuaciones mediante el uso de métodos numéricos, reemplazando las derivadas parciales o integrales por formas algebraicas discretizadas, que permiten obtener valores numéricos para el flujo de fluido en el tiempo y/o espacio [7].

2.3. Modelo fenomenológico

Debido a los altos tiempos de simulación utilizando softwares multi-física convencionales, se ha formulado un modelo fenomenológico [8], que ha sido utilizado en trabajos anteriores [1]. Este modelo busca obtener la temperatura, densidad, presión y velocidad de fluido y la temperatura de las celdas centrales de un banco de baterías organizado como grilla escalonada.

2.3.1. Ecuaciones

Este modelo se basa en los siguientes principios:

1. Caída de presión en intercambiadores de calor para bancos de tubos:

$$\Delta P = (N_l f \chi \rho \frac{V^2}{2}), \quad (2.1)$$

donde ΔP es la diferencia de presión en $[Pa]$, N_l es el número de celdas enfrentadas al fluido, f es el factor de fricción del pack (adimensional), χ es el factor de forma de la configuración (adimensional), ρ la densidad del fluido en $[kg/m^3]$ y V la velocidad del fluido en $[m/s]$.

2. Gas ideal:

$$P = \rho RT, \quad (2.2)$$

donde P es la presión del fluido en $[Pa]$, ρ es la densidad en $[kg/m^3]$, R la constante de gases ideales $[J/kgK]$ y T la temperatura en $[K]$.

3. Conservación de momento:

$$P_i A_i + \dot{m}_i V_i = P_{i+1} A_{i+1} + \dot{m}_{i+1} V_{i+1} + F_{ext}, \quad (2.3)$$

donde P es la presión en $[Pa]$, A el área por donde pasa el flujo en $[m^2]$, \dot{m} el flujo másico en $[kg/s]$, V la velocidad del fluido en $[m/s]$, F_{ext} las fuerzas externas que afectan al fluido en $[N]$ e i es el índice correspondiente al lugar en el espacio donde se evalúa la variable.

En este caso F_{ext} es la fuerza de arrastre que ejerce el cuerpo al fluido, dada por la ecuación:

$$F_{arraastre} = \frac{1}{2} A \cdot \rho \cdot V^2 \cdot C_{drag}, \quad (2.4)$$

donde A es el área proyectada del cuerpo y C_{drag} es el coeficiente de arrastre del cuerpo.

4. Conservación de energía:

$$\dot{m} \left(c_p \Delta T + \frac{\Delta V^2}{2} + g \Delta z \right) = \dot{Q}_{fluido} + \dot{W}_{eje}, \quad (2.5)$$

donde \dot{m} es el flujo másico del fluido en $[kg/s]$, c_p es el calor específico del fluido $[J/kgK]$, ΔT es la diferencia de temperatura en $[K]$, ΔV el cambio en velocidad $[m/s]$, g la aceleración de gravedad $[m/s^2]$, Δz la diferencia en altura $[m]$, \dot{Q}_{fluido} es la tasa de energía transferida hacia el fluido $[J/s]$ y \dot{W}_{eje} es la tasa de trabajo ejercido hacia el fluido $[J/s]$.

5. Transferencia de energía:

$$\dot{q} = -k_{celda} \frac{\Delta T}{L}, \quad (2.6)$$

$$\dot{q} = h_{fluido} \Delta T, \quad (2.7)$$

donde \dot{q} es el calor transferido por unidad de área en $[W/m^2]$, ΔT es la diferencia de temperatura en $[K]$, k_{celda} es el coeficiente conductivo de la celda $[W/mK]$, L el largo de la celda en $[m]$ y h_f es el coeficiente convectivo del fluido $[W/m^2K]$.

2.3.2. Cálculo de variables de salida

Las variables de salida del modelo son calculadas por medio de ecuaciones fenomenológicas, cuyos parámetros provienen de relaciones obtenidas experimentalmente, como el factor de fricción o la correlación de Nusselt-Reynolds, y otros son valores teóricos como el número de Reynolds. Las expresiones para las variables de salida del modelo son las siguientes:

1. Presión de fluido:

$$P(i) = P(i+1) + \frac{N_i f \chi \rho V(i)^2}{2}, \quad (2.8)$$

donde $P(i)$ y $P(i+1)$ son las presiones de aire en la columna de fluido i e $i+1$, respectivamente, en $[Pa]$, N_i es el número de celdas enfrentadas al fluido, f es el coeficiente de fricción (adimensional), χ es el factor de forma (adimensional), ρ es la densidad del fluido en $[kg/m^3]$ y $V(i)$ es la velocidad del fluido en la columna i en $[m/s]$.

2. Velocidad del fluido:

$$V(i+1) = \frac{A(P(i) - P(i+1)) - F_{ext}}{\dot{m}} + V(i), \quad (2.9)$$

donde $P(i)$ es la presión en $[Pa]$ en la columna i , A es el área por donde pasa el flujo en $[m^2]$, \dot{m} es el flujo másico en $[kg/s]$, V es la velocidad en $[m/s]$ y F_{ext} las fuerzas externas (de arrastre) en $[N]$.

3. Temperatura del fluido:

$$T_f(i+1) = \frac{\left(\frac{\dot{q}}{\dot{m}} - \frac{V(i)^2}{2}\right)}{c_p} + T_f(i), \quad (2.10)$$

donde \dot{m} es el flujo másico en $[kg/s]$, T_f es la temperatura del fluido en $[K]$, c_p es el calor específico del fluido en $[J/kgK]$, V es la velocidad del fluido en $[m/s]$ y \dot{q} es la energía transferida hacia el fluido en $[W]$.

4. Densidad del fluido:

La densidad del fluido es calculada en función de la temperatura en el mismo punto. Para esto se realiza una interpolación lineal según datos de temperatura/densidad de aire conocidos previamente, mostrados en la tabla 2.1.

Tabla 2.1: Densidad del aire en función de la temperatura

Temperatura $[^{\circ}C]$	Densidad $[kg/m^3]$
0	1.293
20	1.205
40	1.127

5. Temperatura de la celda:

$$T_c(i) = \frac{\dot{q}}{A_{sup}h_f} + \frac{T_f(i) + T_f(i+1)}{2}, \quad (2.11)$$

donde \dot{q} es el calor transferido por unidad de área en $[W]$, A_{sup} es el área de una celda sin contar sus caras circulares, T_c es la temperatura de la celda en $[K]$, T_f es la temperatura del fluido en $[K]$, h_f es el coeficiente de convección del fluido en $[W/m^2K]$ e i es el bloque en que se está evaluando.

2.3.3. Parámetros de interés

El modelo descrito anteriormente requiere ajustarse para el problema de diseño del empaquetamiento óptimo de baterías de litio. Para esto se pretende ajustar las expresiones del

coeficiente de fricción (y forma), el coeficiente de arrastre y el número de Nusselt, de modo de disminuir el error en las variables de salida con respecto a las simulaciones de ANSYS.

Estos coeficientes son ajustados debido a que sus ecuaciones en el modelo fenomenológico original se encuentran basadas en resultados experimentales, para un flujo de fluido sobre tubos singulares o bancos de tubos [9]. Sin embargo, las condiciones sobre los bancos de baterías que se quieren simular no son necesariamente las mismas que en esos estudios, por lo que requerirían optimización para el caso particular estudiado en este trabajo.

1. **Número de Nusselt:** parámetro adimensional que relaciona las propiedades térmicas del fluido:

$$Nusselt = \frac{h_f L}{k_f} \quad (2.12)$$

Donde h_f es el coeficiente convectivo del fluido en $[W/m^2K]$, L el largo característico del objeto por donde circula el fluido en $[m]$ y k_f la conductividad del fluido en $[W/mK]$. El número de Nusselt es ocupado para calcular h_f , y con éste las temperaturas de las celdas según la ecuación 2.11.

El número de Nusselt se puede relacionar en función del número de Reynolds y el número de Prandtl:

$$Reynolds = \frac{VL}{\nu}, \quad (2.13)$$

$$Prandtl = \frac{\nu}{\alpha}, \quad (2.14)$$

donde V es la velocidad de fluido, ν es la viscosidad cinemática del fluido y α es la difusividad térmica del fluido.

La correlación del número de Nusselt con el número de Reynolds y el número de Prandtl es de la forma

$$Nu = C Re^m Pr^n, \quad (2.15)$$

donde C , m y n so constantes que dependen del tipo de configuración y rango de aplicación del modelo. Se han obtenido varias correlaciones empíricamente, y en particular Žkauskas [9] postula lo siguiente para configuraciones escalonadas de tubos:

$$Nu \begin{cases} 0,9Re^{0,4}Pr^{0,36}\left(\frac{Pr}{Pr_s}\right)^{0,25} & \text{para } 10 < Re < 100 \\ 0,51Re^{0,5}Pr^{0,36}\left(\frac{Pr}{Pr_s}\right)^{0,25} & \text{para } 100 < Re < 1000 \\ C'0,35Re^{0,6}Pr^{0,36}\left(\frac{Pr}{Pr_s}\right)^{0,25} & \text{para } 1000 < Re < 2 \cdot 10^5, \end{cases} \quad (2.16)$$

donde Pr_s es el número de Prandtl para la superficie de un sólido, y C' es un factor de corrección del número de Nusselt en función de en que columna de fluido se está evaluando:

2. **Factor de fricción y factor de forma:** estos dos factores son utilizados para calcular el valor de la presión del aire ($f\chi$ en la ecuación 2.8). Ambos son reajustados de forma conjunta bajo el nombre de factor de fricción.

Tabla 2.2: Factor de corrección del número de Nusselt en función de la columna de fluido evaluada

Columna	1	2	3	4	5	7	10	13	16	≥ 20
C'	0,64	0,76	0,84	0,89	0,92	0,95	0,97	0,98	0,99	1

Se han obtenido valores aproximados para estos coeficientes de forma experimental [9], y la curva obtenida tiene la forma general:

$$f_{friccion} * f_{forma} = A(S) \cdot Re^{B(S)} \quad (2.17)$$

3. **Coefficiente de arrastre:** es utilizado para calcular las fuerzas externas de arrastre (C_{drag} en la ecuación 2.4), que influyen en el cálculo de la velocidad del fluido. Se han obtenido valores para este coeficiente de forma empírica [9] sobre un flujo perpendicular a tubos cilíndricos, obteniendo relaciones con el número de Reynolds [9][10].

2.4. Programación genética

La programación genética es una técnica de la computación evolutiva que busca resolver problemas sin que el usuario deba conocer o especificar la forma o estructura específica de la solución de forma previa [11]. Esta técnica puede buscar en el espacio de posibles programas computacionales un individuo que sea apto para resolver el problema propuesto. Dependiendo del problema abordado, un programa computacional puede ser una fórmula matemática, un plan, una estrategia de control, un procedimiento, etc. [2].

La programación genética es una variante del algoritmo genético convencional, que se basa en el proceso de evolución natural. Este algoritmo consiste en transformar un conjunto de objetos matemáticos individuales o población. Típicamente un individuo se representa por una cadena de caracteres simulando el funcionamiento de un cromosoma, con un valor de fitness asociado. Se obtiene una nueva población usando operaciones basadas en el proceso de evolución Darwiniana y operaciones genéticas de ocurrencia natural, tales como mutación o recombinación. Con esto, en cada generación los individuos más aptos tienen una mayor probabilidad de sobrevivir y reproducirse para formar la próxima generación.

En la programación genética los individuos o programas son representados generalmente por estructuras en árbol, las cuales tienen la propiedad de poder ser evaluadas recursivamente. Así los individuos representan programas computacionales jerárquicos. El tamaño, la forma y el contenido de estos programas varían en cada generación.

Las variables y constantes del programa son denominadas “terminales”, y se encuentran en los nodos exteriores u “hojas” de los árboles que representan los individuos. Por otra parte, en los nodos internos se encuentran las “funciones”, que operan sobre los terminales u otras funciones para crear funciones.

La programación genética sigue los mismos pasos que los algoritmos genéticos, como se observa en la figura 2.2, para transformar la población inicial y buscar mejores soluciones al

problema propuesto. El algoritmo de programación genética consiste en los siguientes pasos:

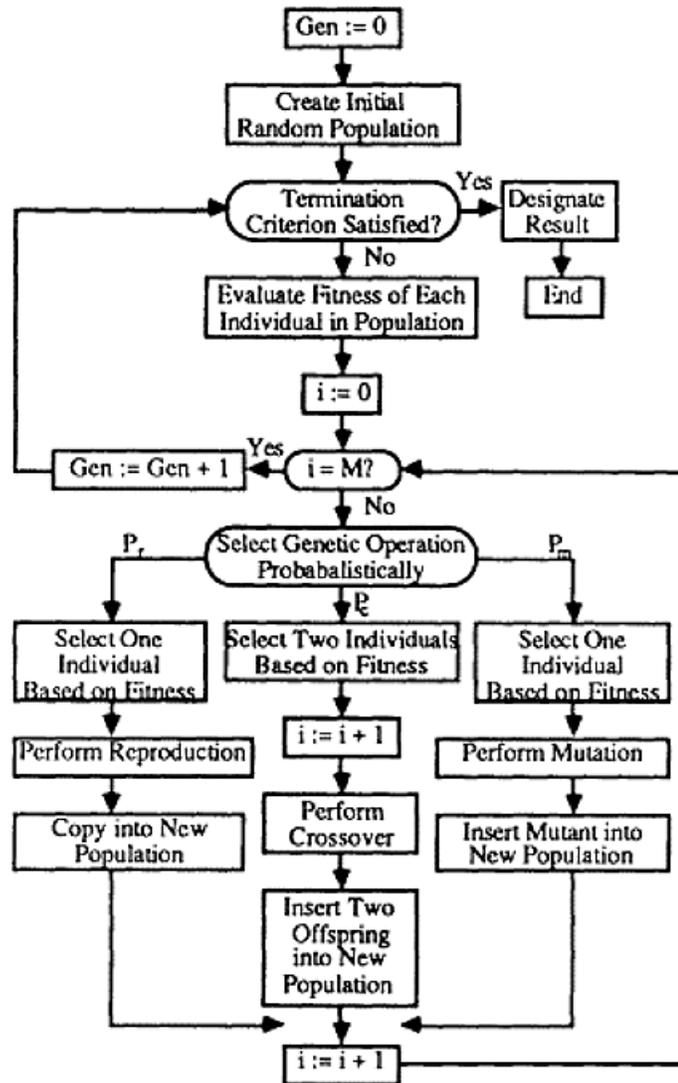


Figura 2.2: Diagrama de flujo convencional de algoritmos genéticos. Fuente [2]

1. Inicialización del algoritmo: Hay cinco grandes pasos en la preparación para usar programación genética en la resolución de un problema [2]. Estos consisten en determinar:
 - (a) Conjunto de terminales: consisten en las variables de entrada o constantes del programa.
 - (b) Conjunto de funciones: corresponden a las funciones que operan sobre los terminales u otras funciones.
 - (c) Medida de fitness: consiste en identificar la forma en que se evalúa el desempeño de un programa para resolver el problema. Generalmente consiste en una expresión matemática, comúnmente una medida de error o función de costo.
 - (d) Parámetros y variables de control: entre los cuales se encuentran el número de generaciones y de individuos. Éstos deben elegirse teniendo en consideración la

complejidad del problema a resolver, ya que los recursos computacionales requeridos aumentan junto con ellos.

- (e) Designación de resultado y criterio de término: consiste en especificar el criterio para finalizar el algoritmo, entre los cuales se encuentran la convergencia de la mejor solución en sucesivas generaciones, un número máximo de generaciones desde el inicio, etc. Junto con esto se designa al mejor individuo encontrado, que generalmente pertenece a la última generación calculada.

Junto con estos pasos también se escogen otros parámetros necesarios para el funcionamiento del algoritmo:

- (a) Generación de población inicial: existen varios métodos, entre los cuales se encuentran:
 - i. Método “full”: consiste en generar árboles en que cada camino directo entre una hoja y la raíz sea de igual largo a la altura máxima de los árboles.
 - ii. Método “grow”: consiste en crear árboles de tamaño variable aleatorio, sin sobrepasar el tamaño máximo.
 - iii. Método “ramped half-and-half”: incorpora los métodos “full” y “grow”. Se crean un mismo número de árboles para cada altura entre 2 y el máximo, y para cada valor se genera la mitad de los árboles con el método “full” y la otra mitad con el método “grow”.
- (b) Tamaño máximo de los árboles.
- (c) Selección: consiste en elegir el método de selección de individuos a los que se les aplicará operadores genéticos. Generalmente se busca que sean los individuos más aptos. Entre algunos métodos se encuentran:
 - i. Ruleta o proporcional: en que cada individuo tiene una probabilidad de selección proporcional a su fitness en relación a la población.
 - ii. Torneo: en que se selecciona al individuo con mayor fitness de un subconjunto aleatorio de la población.
- (d) Operadores genéticos: consiste en elegir los operadores genéticos a usar y su probabilidad de ocurrencia. Entre estos se encuentran:
 - i. Crossover o recombinación: se seleccionan dos padres y un punto de crossover aleatorio para cada uno de ellos. Luego los sub-árboles que tienen como raíz los puntos de crossover son intercambiados entre los padres.
 - ii. Mutación: se selecciona un punto de mutación aleatorio en el árbol. Luego se cambia el sub-árbol que tiene como raíz el punto de mutación por otro generado aleatoriamente.
 - iii. Permutación: se selecciona un nodo función y se permuta aleatoriamente el sub-árbol del que es raíz.
 - iv. Edición: que busca simplificar las expresiones donde sea posible.

Posteriormente a la inicialización comienza un proceso iterativo de búsqueda de la mejor solución:

2. Evaluación: se calcula el fitness de los individuos que conforman la población.
3. Selección: se seleccionan los individuos para crear la siguiente generación, de acuerdo a los criterios de selección escogidos en la inicialización.

4. Operadores genéticos: Se crean nuevos individuos mediante la aplicación de los operadores genéticos elegidos con la probabilidad determinada en la inicialización.
5. Evaluación criterio de término: se evalúa si se debe finalizar el algoritmo según los criterios determinados durante la inicialización. De no cumplirse algún criterio, se vuelve al proceso de evaluación y se itera nuevamente.

2.4.1. Control de bloat

Uno de los principales problemas de la programación genética es la de bloat o inflación de código [2]. Esto es, el crecimiento de los individuos sin una mejora significativa en el fitness, causando un mayor uso de recursos computacionales innecesariamente. Entre los métodos para controlarlo se encuentran [12]:

- Imponer límites de tamaño e incluir la complejidad estructural en la función de costo [2].
- Operadores de edición: pueden ser operadores que periódicamente simplifiquen los árboles, eliminando subárboles que no contribuyan a la solución final.
- Inclusión de presión de parsimonia en el método de selección: ya sea seleccionando individuos basados en su tamaño o realizando dos torneos, uno de fitness y otro de tamaño.
- Remover individuos demasiado grandes de la población.
- Operador ‘prune and plant’: se ‘poda’ una ramas del individuo elegido y se reemplaza por un terminal. Luego la rama podada se ‘planta’ en la población como un nuevo árbol. Con esto se obtienen dos individuos de menor tamaño que su ancestro.

2.5. Optimización multiobjetivo

Un problema de optimización multiobjetivo es aquel que involucra múltiples funciones objetivo generalmente opuestas entre sí $f_j, j = 1, \dots, k$. Matemáticamente, un problema con k objetivos a minimizar se formula como:

$$\min_{x \in X} (f_1(x), \dots, f_k(x)), \quad (2.18)$$

donde x corresponde a un vector de n dimensiones de variables de entrada $x = (x_1, \dots, x_n)$ y X contiene todos los posibles x que pueden ser usados en las ecuaciones.

Debido a que no es una única función que se optimiza, frecuentemente no hay una única posible solución sino un set de soluciones [13]. Luego el problema consiste en encontrar las llamadas soluciones óptimas de Pareto, que corresponden a aquellas soluciones x_i en que no exista un x_j que disminuya el valor de alguna función objetivo sin disminuir el de otra. La figura 2.3 ilustra un ejemplo de frente de Pareto.

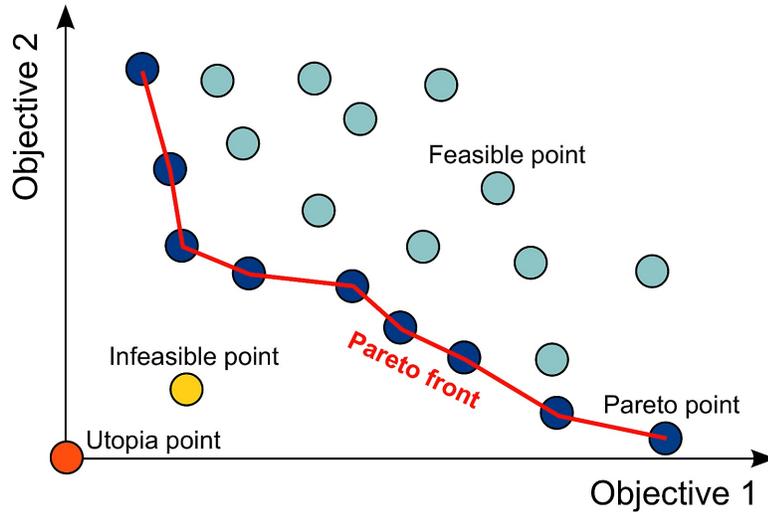


Figura 2.3: Ejemplo de frente de Pareto. Fuente [3]

2.6. Trabajo anterior

El ajuste del modelo fenomenológico descrito anteriormente fue abordado en forma simplificada en la memoria de título de Francisco Villa [1], donde se consideraron solo 5 celdas y se descompuso el problema en tres partes, optimizando por separado el coeficiente de arrastre, factor de fricción y número de Nusselt. La implementación utilizada entonces se hizo en MATLAB, lo que provocó problemas debido a la baja eficiencia computacional, necesitando grandes cantidades de tiempo en cada iteración del algoritmo de programación genética.

En trabajos posteriores a la memoria de Francisco Villa, el código fue portado a Java utilizando el toolkit ECJ [14], aumentando la velocidad de simulación notablemente. En esta plataforma también fue implementado el algoritmo evolutivo multiobjetivo NSGA2, permitiendo minimizar el funcional completo del problema, buscando el frente de Pareto.

Dentro de las mejoras del código original de Francisco Villa se destaca el trabajo de Germán García y Nicolás Tapia [4], donde fueron corregidas algunas ecuaciones del modelo fenomenológico implementado y se limitó el rango de las entradas, logrando encontrar soluciones con un error significativamente menor en este rango de operación.

Capítulo 3

Metodología

En este capítulo se presenta la metodología a utilizar en la presente memoria.

3.1. Obtención de Datos

3.1.1. Modelo ANSYS

Los datos utilizados como estándar dorado para la creación de los conjuntos de entrenamiento, validación y prueba fueron obtenidos usando el software CFD ANSYS con un modelo proporcionado por un experto. La geometría básica del modelo es de grilla escalonada con 102 celdas, intercalando columnas de 4 y 3 unidades, tal como puede verse en el ejemplo de la figura 3.1. Los bordes del banco siempre se encuentran a una distancia de 15 [mm] de las celdas externas.

Como condiciones de borde en el borde izquierdo se encuentra la entrada de aire, en que el flujo másico y temperatura se mantienen constantes durante la simulación a F_{in} y T_{in} , respectivamente. El borde derecho corresponde a la salida de aire y tiene como condición una presión igual a la presión atmosférica y es ocupado como referencia (presión parcial = 0). Los bordes superior e inferior, junto con las paredes de las celdas son estacionarias sin deslizamiento (velocidad de fluido = 0) y sin rugosidad. El material de los bordes superior e inferior es aluminio y son paredes adiabáticas, por lo que no permiten la transferencia de calor de un lado a otro. El material de las celdas es litio y pueden transferir calor hacia/desde el aire.

3.1.2. Simulaciones

Se realizaron simulaciones con distintos valores de las variables de diseño: diámetro de celda, espaciado entre celdas, flujo volumétrico en la entrada, corriente de uso y temperatura

inicial del aire. Estas variables de interés tienen valores en los siguientes rangos:

- Corriente I : 0 a 15 [A]
- Diámetro D : 18, 22, 26, 32 [mm]
- Separación S : 0,3 a 1,5 [veces D]
- Flujo de entrada F_{in} : 1 a 200 [CFM]
- Temperatura inicial de fluido T_{in} : 10 a 30 [°C]

Debido a que no fue posible parametrizar el tiempo de simulación en función de la corriente (siendo la intención inicial simular una descarga completa), se decidió simular para todos los casos un tiempo de 960 [s], correspondiente al tiempo mínimo de descarga con la máxima corriente y suponiendo una capacidad de 4 [Ah] para la batería.

Los resultados obtenidos de las simulaciones de ANSYS son: velocidad de flujo (V_f), presión de fluido (P_f) y temperatura de la celda central de cada bloque (T_c). Las salidas correspondientes al fluido son medidas como el promedio del parámetro en una línea entre las dos celdas centrales de cada columna de cuatro celdas (líneas amarillas en la figura 3.2). Por otro lado, la temperatura de las celdas centrales es medida en las columnas de tres unidades (celdas en azul en la figura 3.2).

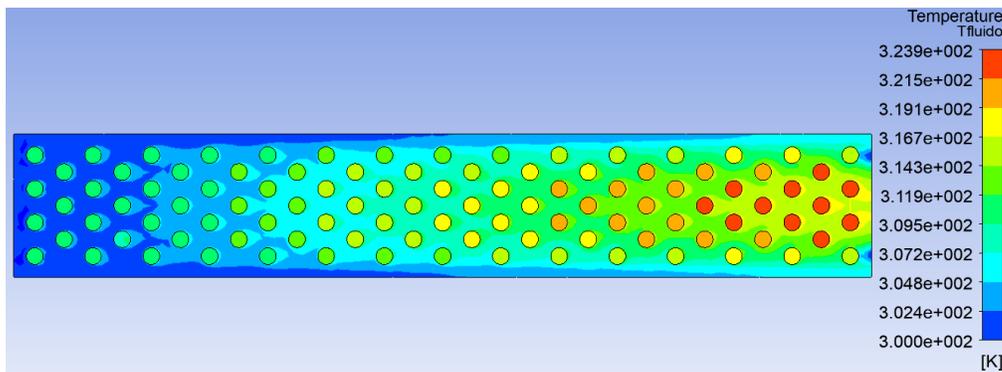


Figura 3.1: Ejemplo de resultado de simulación en ANSYS.

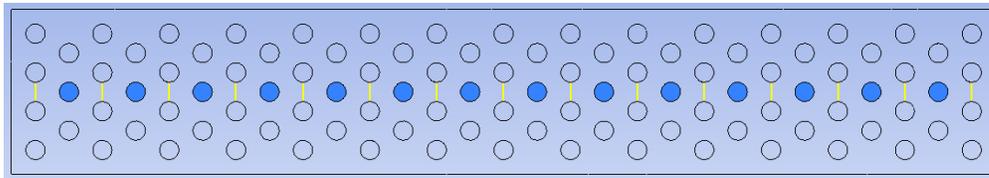


Figura 3.2: Lugares de "medición" de salidas en ANSYS.

Las simulaciones fueron realizadas en dos computadores, uno con un procesador Intel Core i5-7300 HQ y otro con un Intel Core i7-4790. En ambos fue notado que la mayoría de las simulaciones terminaban en un tiempo menor a cinco minutos en los computadores ocupados, pero que aquellas que no terminaban en este tiempo generalmente llegaban a varias horas o incluso días. Debido a esto se decidió limitar el tiempo máximo de simulación a 7,5 minutos, descartando de la base de datos las configuraciones que excedían este tiempo de simulación.

Con el objetivo de poder comparar los resultados del nuevo modelo ANSYS con los obtenidos en el trabajo de Francisco Villa [1], se decidió realizar simulaciones con las mismas entradas ocupadas en ese trabajo para crear la base de datos. Considerando los nuevos rangos más acotados, se realizaron un total de 1256 simulaciones, de las cuales solo 1077 terminaron dentro del tiempo establecido. Divididas en los mismos conjuntos en que se encontraban inicialmente, esto resultó en: 344 datos de entrenamiento (32 %), 356 de validación (33 %) y 377 de prueba (35 %).

3.2. Modelo fenomenológico

Para realizar el cálculo de las salidas del modelo fenomenológico internamente el espacio es dividido en ‘bloques’ compuestos de 5 celdas, como se muestra en la figura 3.3. En el caso de un banco de 102 celdas como el ocupado en este trabajo corresponden 14 bloques posicionados horizontalmente, representando las celdas centrales del banco (ver figura 3.2).

Luego de establecer las condiciones iniciales se resuelve iterativamente hasta que las salidas convergen o se llega a un número máximo de iteraciones preestablecido. Entonces para cada bloque i se rescata la velocidad de fluido (V_f) en la columna $i + 1$, la presión de fluido (P_f) en la columna i y la temperatura de la celda central (T_c).

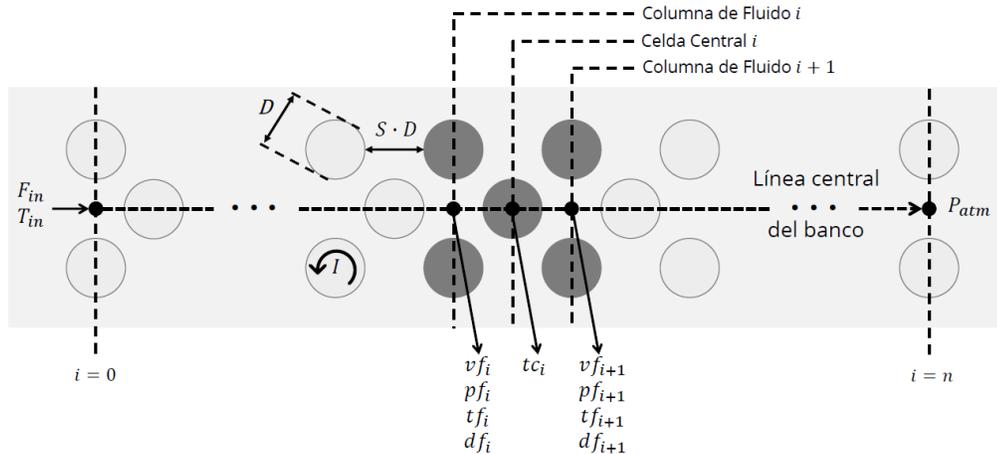


Figura 3.3: Esquema banco de baterías. Fuente [4]

3.3. Evaluación de expresiones encontradas en trabajos anteriores

Una vez obtenida la base de datos se procedió a evaluar el error de salida del modelo fenomenológico ocupando las expresiones encontradas en trabajos anteriores para coeficiente de arrastre, factor de fricción y número de Nusselt, con el objetivo de establecer su desempeño

al aumentar el número de columnas de celdas y tener una base con la cual comparar resultados posteriores. Estas mediciones fueron realizadas sobre el conjunto de prueba.

- Modelo fenomenológico original (Reyes [8]): Ajuste de coeficientes en modelo fenomenológico basado en literatura. Tiene expresiones de la forma:

$$Coef\ arrastre = 3,3481Re^{0,179} \quad (3.1)$$

$$Factor\ friccion = 0,7 \cdot A(S)Re^{B(S)} \quad (3.2)$$

Donde $A(S)$ y $B(S)$ son funciones lineales por tramo en función de la separación entre celdas, dados por la tabla 3.1.

Tabla 3.1: $A(S)$ y $B(S)$ en función de la separación entre celdas para el modelo fenomenológico original.

Separación	0,25	0,5	1	1,5
A	82,188	38,446	11,728	1,2095
B	-0,605	-0,54	-0,402	-0,211

$$Nusselt = C'(Re, indCol) Re^{0,653} \quad (3.3)$$

Donde C' varía en función del tramo en que se encuentre el número de Reynolds y de la columna en el banco en que se evalúa, según lo descrito en la ecuación 2.15 y la tabla 2.2.

- Modelo Villa [1]: Búsqueda individual de ecuaciones de interés mediante programación genética en modelo de 5 celdas.

$$Coef\ arrastre = 4,78 \left(0,636(4,81Re^{2,3 \cdot 0,49An}) Dfn \frac{An}{Re} \right)^{0,0148} \quad (3.4)$$

$$Factor\ friccion = 4 \left(1,021 * (9S)^{(2,81*S)^{-2}} + \left(\frac{0,3021}{S^2} \right)^2 - 0,6 \right) Re^{-0,2655} \quad (3.5)$$

$$Nusselt = Re^{0,677} \quad (3.6)$$

- Modelo Garcia-Tapia [4]: Búsqueda simultánea de ecuaciones de interés mediante programación genética en modelo de 5 celdas.

$$Coef\ arrastre = 0,891An - An^2 + Re^{0,1452} \quad (3.7)$$

$$Factor\ friccion = Dfn * S^{(24,5261Re^{-0,3891} - 2)} \quad (3.8)$$

$$Nusselt = 2,0232Re^{0,5528} \quad (3.9)$$

Como principal forma de visualización del error se decidió ocupar histogramas del error para las variables de salida en los distintos bloques, además del RMSE (*Root-mean -square error*), definido por la expresión 3.10:

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (\varepsilon_i)^2}, \quad (3.10)$$

donde N corresponde al número de datos evaluados y ε es el error entre el modelo fenomenológico y la simulación en ANSYS.

3.4. Programación genética

Se procedió a utilizar programación genética con el objetivo de encontrar expresiones para el coeficiente de fricción, el de arrastre y el número de Nusselt, que mejor se adapten a la nueva base de datos. La evolución de los tres coeficientes se realizó de forma simultánea utilizando el algoritmo evolutivo multiobjetivo NSGA2 en el toolbox ECJ en Java.

El programa utilizado para esto se encuentra implementado en Java, utilizando el toolkit de computación evolutiva ECJ[14] para el algoritmo de programación genética.

3.4.1. Fitness

Se utilizan tres funciones objetivo para evaluar el fitness de los individuos en cada iteración del algoritmo. Estas corresponden a métricas de error más un parámetro de regularización en las siguientes variables de salida:

1. Velocidad de fluido (V_f)
2. Presión del fluido (P_f)
3. Temperatura de la celda central (T_c)

Debido a que se pretendía que la expresión obtenida fuera generalizable para un número arbitrario de celdas, en primera instancia se consideró como métrica el error cuadrático medio, considerando los errores en los 14 bloques de la configuración evaluada, según la expresión 3.11:

$$RMSE_{14}(Y) = \sqrt{\frac{1}{N} \sum_{k=1}^N \left(\frac{1}{14} \sum_{i=1}^{14} (Y_{ANSYS,k}(i) - Y_{PG,k}(i))^2 \right)}, \quad (3.11)$$

donde Y corresponde a la salida evaluada (V_f , P_f o T_c), N es el número de datos en el conjunto de entrenamiento, k es la muestra evaluada, i es el índice del bloque evaluado, $Y_{ANSYS,k}(i)$ e $Y_{PG,k}(i)$ son la salida deseada y la salida del individuo evolucionado, respectivamente, en la muestra k y el bloque i .

A cada métrica de error se le suma un término de regularización con el objetivo de controlar el crecimiento de los árboles, obteniendo las siguientes expresiones para fitness:

$$f_1(\theta) = RMSE_x(V_f) + \alpha\sqrt{s(\theta_1)} \quad (3.12)$$

$$f_2(\theta) = RMSE_x(P_f) + \alpha\sqrt{s(\theta_2)} \quad (3.13)$$

$$f_3(\theta) = RMSE_x(T_c) + \alpha\sqrt{s(\theta_3)}, \quad (3.14)$$

donde α corresponde a un factor de regularización que controla el tamaño del árbol, θ al conjunto de árboles que conforman a un individuo ($[\theta_1, \theta_2, \theta_3]$) y $s(\theta_i)$ corresponde al número de nodos de los árboles correspondientes a cada función objetivo (árboles de coeficiente de arrastre (θ_1), factor de fricción (θ_2) y número de Nusselt (θ_3)). Entonces el problema de minimización puede formularse como:

$$F(\theta) = \min_{\theta} (f_1(\theta), f_2(\theta), f_3(\theta)). \quad (3.15)$$

3.4.2. Parámetros

Los parámetros de configuración utilizados fueron los siguientes:

- Inicialización de población
 - Método: Ramped Half and Half
 - Profundidad máxima inicial: 5
 - Individuos: 1000
- Tamaño de árboles máximo: 50 nodos
- Set de funciones: $\{+, -, *, /, \wedge, \wedge^2\}$
- Terminales:
 - Coeficiente de arrastre: $\{rand, \frac{A_{sup}}{A_{ctrl}}, \frac{d_f(i)}{d_{fref}}, S, Rem(i)\}$
 - Factor de fricción: $\{rand, \frac{Vm_f}{V_{fref}}, \frac{d_f}{d_{fref}}, S, Rem(i)\}$
 - Nusselt: $\{rand, Rem(i)\}$

Donde $rand$ es un número aleatorio entre -1 y 1, A_{sup} es el área de las celdas menos las caras circulares en $[m^2]$, A_{ctrl} es el área que hay entre dos celdas de una columna de fluido en $[m^2]$, $d_f(i)$ es la densidad del fluido en la posición i en $[kg/m^3]$, $d_{fref} = 1, 205$ es la densidad de referencia en $[kg/m^3]$, S es la separación entre celdas (adimensional), $Rem(i)$ es el número de Reynolds en la posición i (adimensional), $Vm_f(i)$ es la velocidad media del fluido en la posición i en $[m/s]$ y V_{fref} es una velocidad de referencia en $[m/s]$.

- Operadores genéticos:
 - Mutación: probabilidad 20 %
 - Crossover: probabilidad 80 %
- Selección: “lexicographic parsimony pressure” [15]

- Supervivencia: mantener el mejor
- Condición de término: 1000 generaciones

3.4.3. Variación de parámetros

Según lo sugerido en la literatura [16] [17], se consideró la adición del índice de la columna evaluada como un terminal para la expresión del número de Nusselt. Para evaluar esto se realizó una corrida de programación genética añadiendo el terminal ‘ColIndex’, correspondiente al número de la columna en que se está evaluando, y se comparó con una corrida con los parámetros originales.

Obtenidos los individuos de ambas corridas se procedió a comparar los frentes de Pareto de los fitness finales de los individuos en el conjunto de entrenamiento y los frentes de Pareto obtenidos calculando $RMSE_{14}$ sobre Vf, Pf y Tc en el conjunto de validación.

Realizada la comparación anterior y escogidos los terminales a ocupar se procedió a variar los parámetros de factor de regularización (α) y el tamaño máximo de los árboles, con el objetivo de controlar la complejidad de las expresiones y así facilitar la interpretación. Se realizaron pruebas con los siguientes valores:

- Factor de regularización α : {0.1, 0.25, 0.5}
- Tamaño máximo de árboles: {20, 50} nodos

Para cada caso se compararon los frentes de Pareto obtenidos sobre el conjunto de validación, además del tamaño promedio de los árboles en cada generación, con el objetivo de obtener una descripción del comportamiento al variar los parámetros.

3.4.4. Comparación con frentes de Pareto obtenidos con expresiones anteriores

De los frentes de Pareto obtenidos en la parte anterior se procedió a realizar una comparación con los modelos desarrollados por Reyes, Villa y Garcia-Tapia, con el objetivo de visualizar si existe una mejora al utilizar las nuevas expresiones encontradas. Para esto se calculó el $RMSE_{14}$ según la expresión 3.11 sobre el conjunto de validación de los modelos anteriores y se representó gráficamente con respecto al frente de Pareto obtenido.

Posteriormente se escogieron 4 individuos para representar el conjunto de Pareto obtenido, correspondientes a:

- $best(Vf)$: Mínimo $RMSE_{14}(V_f)$
- $best(Pf)$: Mínimo $RMSE_{14}(P_f)$
- $best(Tc)$: Mínimo $RMSE_{14}(T_c)$
- $best(prod)$: Mínimo producto entre $RMSE_{14}(V_f)$, $RMSE_{14}(P_f)$ y $RMSE_{14}(T_c)$

Se procedió a visualizar su desempeño graficando el RMSE y error promedio por columna, histogramas de error en todas las salidas y mostrando las salidas de cada modelo vs las salidas en ANSYS en los bloques externos de la configuración.

3.4.5. Variación en cálculo de Nusselt

Se realizaron dos corridas de programación genética, manteniendo los mejores parámetros obtenidos anteriormente, pero variando la forma de obtener el número de Nusselt. En el primer caso se separaron los terminales 'Re' e 'indCol' en dos subárboles diferentes que son multiplicados entre si para obtener el resultado final. Con esto se buscó simplificar la expresión final y así simplificar su comprensión.

En el segundo caso se trabajó asumiendo que la dependencia del número de Nusselt con la columna tiene el mismo comportamiento que el modelo original utilizado [9] (denotado por $Z(indCol)$) y multiplicando este factor por la expresión obtenida del árbol θ_3 .

Obtenidos los individuos de las generaciones finales se procedió a graficar los frentes de Pareto obtenidos sobre el conjunto de validación, para así observar el desempeño comparativo obtenido. Junto con lo anterior, de cada corrida se escogió un individuo de cada corrida para comparar, siguiendo el mismo procedimiento de $best(prod)$ que en la sección anterior, obteniendo los modelos $best(prod)_2$ en el caso de separación de subárboles y $best(prod)_3$ en el otro caso.

3.4.6. Configuraciones de 53 y 151 celdas

Con el objetivo de determinar si los modelos obtenidos mantienen su validez y desempeño al variar el número de celdas en la configuración ocupada, se crearon dos nuevas bases de datos de prueba utilizando configuraciones de 53 y 151 celdas, ordenadas en 7 y 21 bloques '4-3-4' respectivamente. En ambos casos las entradas (I , D , S , F_{in} y T_{in}) fueron las mismas que en el conjunto de prueba original de 102 celdas.

Obtenidos los datos se procedió a simular sobre las nuevas bases de datos utilizando las expresiones de los individuos obtenidos en la sección 3.4.3, calculando el error en las salidas Vf, Pf y Tc para todos los bloques. Obtenidos los errores de cada individuo se calculó el $RMSE_7$ (caso 53 celdas) y el $RMSE_{21}$ (caso 151 celdas) y se graficaron los frentes de Pareto obtenidos, con el objetivo de analizar las posibles diferencias al caso de 102 celdas.

Junto a lo anterior se graficó el error por columna en las 3 bases de datos de prueba (53, 102 y 151 celdas) para el individuo $best(prod)$ obtenido, con el objetivo de ver las diferencias en la evolución del error por columna evaluada.

Capítulo 4

Análisis y resultados

4.1. Evaluación de modelos anteriores

Para conocer la generalización de 5 a 102 celdas de los trabajos mostrados previamente, se evaluaron tres modelos (ecuaciones 3.1 a 3.9) y se calcularon las salidas de cada uno de ellos dadas las entradas del conjunto de prueba, comparando los resultados con las salidas de ANSYS.

En la figura 4.1 se observa el RMSE (fila superior) y el error promedio (fila inferior) por columna de la configuración de 102 celdas. En la figura 4.2 se observan los histogramas de errores sobre las salidas de todas las configuraciones de entrada. En la figura 4.3 se muestran las salidas predichas y esperadas en los bloques ‘extremos’ del modelo (correspondientes al primer bloque para V_f/T_c y el último para P_f), en los cuales se esperaría encontrar los errores más altos debido a la acumulación del error. En las tres figuras los gráficos de la izquierda corresponden a velocidad de flujo, los centrales a presión de fluido y los de la derecha a la temperatura de la celda central.

En la figura 4.1 se observa que para la velocidad de fluido el modelo original (Reyes) tiende a ser el con menor RMSE en la mayoría de las bloques, solo siendo peor que el modelo de Villa a partir del bloque 12. Esto puede deberse a las diferentes medias que tienen a lo largo de las columnas, ya que en el modelo de Reyes en promedio se sobreestiman las velocidades de salida, haciendo que las de entrada en los bloques posteriores sean en promedio mayores a las reales, produciendo un mayor error en el cálculo resultante respecto a los datos esperados.

Por otra parte, el modelo García-Tapia muestra un RMSE mayor en todas las columnas del modelo, debido a que presenta algunos errores de gran magnitud relativa, los que se concentran en las configuraciones con una separación entre celdas baja ($0,3 - 0,4[\text{veces } D]$), tal como puede verse en la figura 4.4. Estos errores que no se ven en los otros modelos probablemente se deben a que este modelo fue entrenado con un rango de entradas más acotado que el de Villa, haciendo que el rango de operación sea menor. Luego al aumentar el número de celdas pueden producirse conjuntos de entradas fuera del rango original.

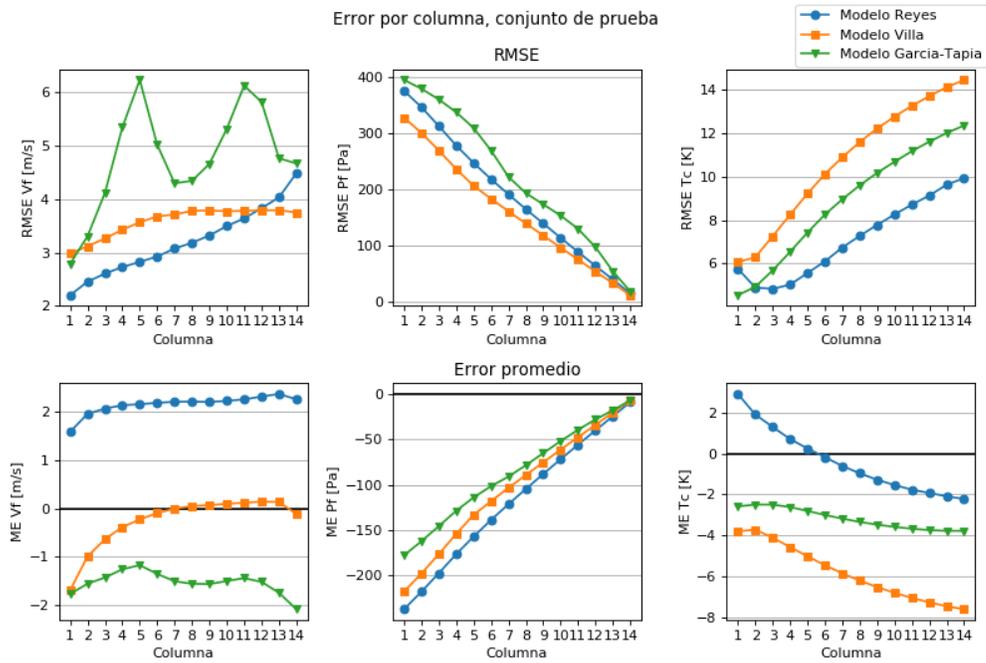


Figura 4.1: RMSE y ME por columna para los modelos de Reyes, Villa y Garcia-Tapia.

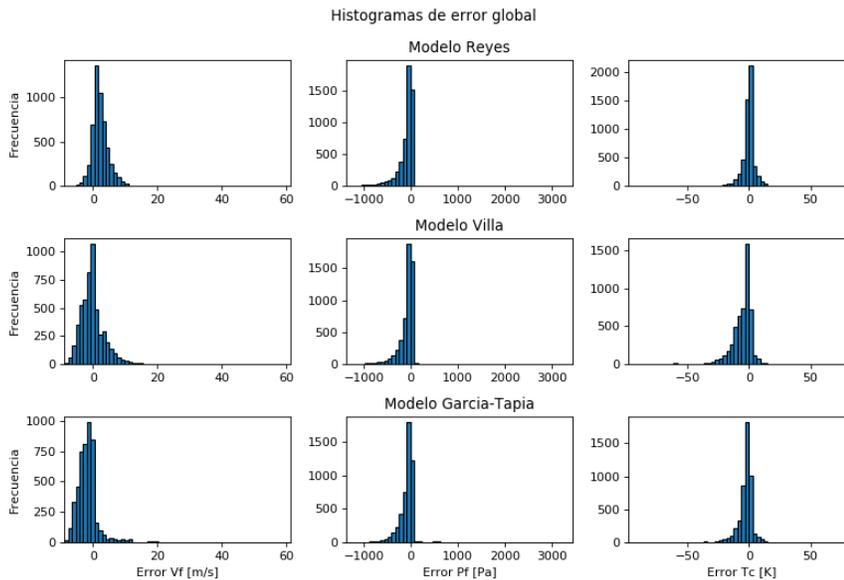


Figura 4.2: Histogramas de errores globales para los modelos de Reyes, Villa y Garcia-Tapia para el caso de 102 celdas.

En la figura 4.1, columna central, se observa que todos los modelos subestiman la presión resultante en cada bloque, haciendo crecer el error absoluto hacia el primer bloque del arreglo, en que la presión corresponde a la suma de las presiones parciales en los demás bloques. El modelo de García-Tapia nuevamente presenta los mayores errores para separaciones cercanas a $0,3[veces D]$, que corresponderían a los mismos puntos en que el error en Vf también es elevado.

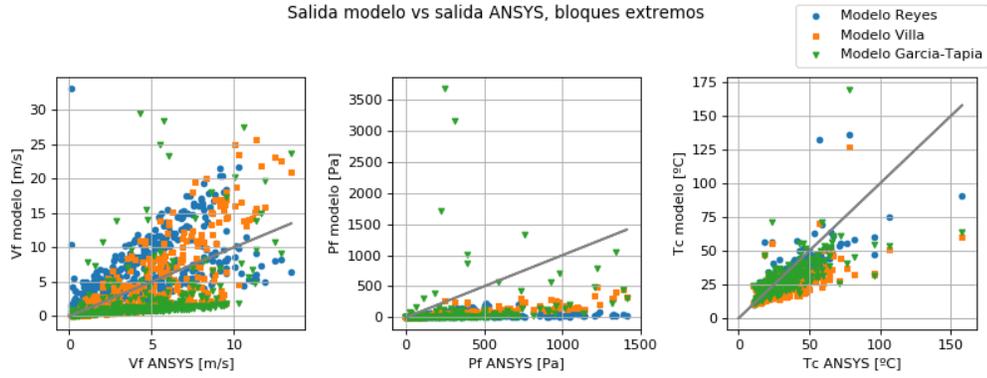


Figura 4.3: Salidas de modelos vs salida ANSYS en bloques extremos (14 para V_f y T_c , 1 para P_f) para los modelos de Reyes, Villa y Garcia-Tapia en el conjunto de prueba.

En cuanto a la temperatura de las celdas centrales (figuras 4.1 y 4.2, columna derecha), se observa que tanto el modelo de Villa como el de García-Tapia subestiman la temperatura esperada, pero este último errores más pequeños y por lo tanto un menor RMSE. Por otra parte, el modelo original tiene un mejor desempeño a partir del segundo bloque, con un RMSE menor y un error promedio más cercano a cero.

Es importante notar que tanto el modelo de Villa como el de García-Tapia fueron entrenados minimizando el error sobre una configuración con un solo bloque de 5 celdas, por lo que el mayor error obtenido por estos modelos al expandir la configuración a un mayor número de celdas se debe a que las expresiones obtenidas sobre la configuración de 5 celdas no permiten describir el comportamiento del nuevo caso con un modelo más grande.

Error Vf y Pf modelos vs separación, bloque 10

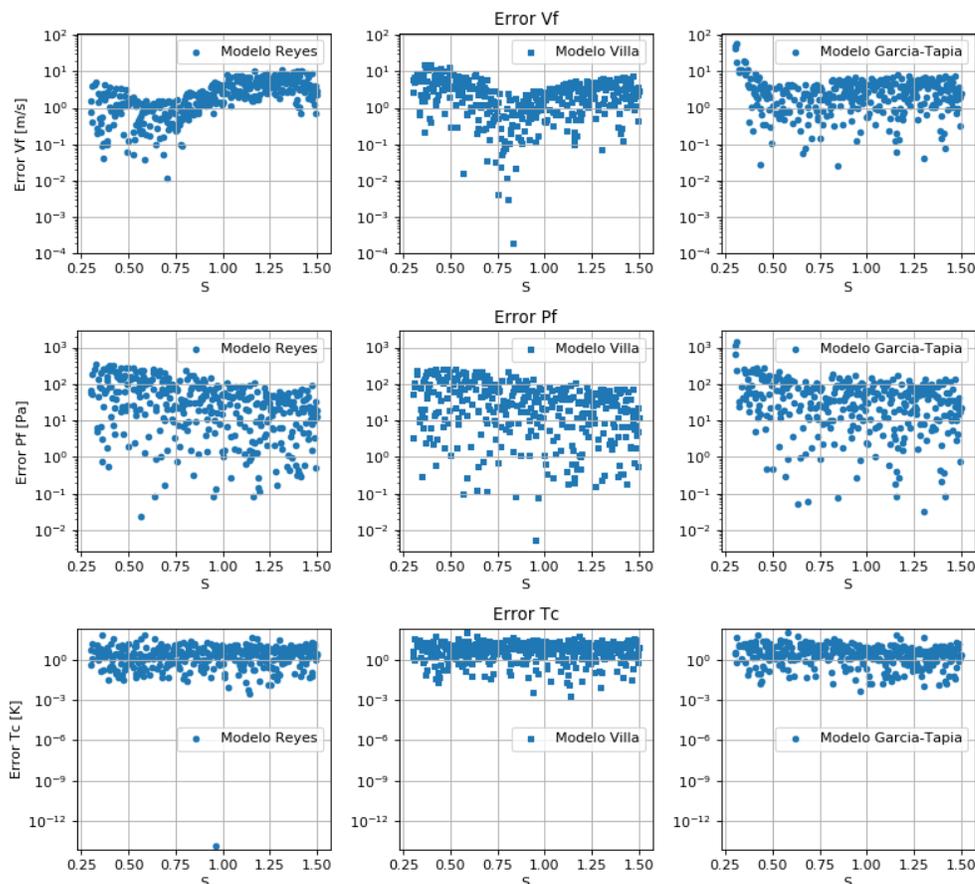


Figura 4.4: Salidas de modelos vs separación entre celdas en bloque 10 de la configuración.

4.2. Programación genética

4.2.1. Adición de nuevo terminal

Se realizó el entrenamiento del modelo propuesto de 102 celdas con los parámetros originales (ver sección 3.4.2) y otra agregando el índice de columna evaluada como terminal a la expresión del número de Nusselt, obteniendo los frentes de Pareto en entrenamiento observados en la figura 4.5. Los frentes de Pareto obtenidos calculando $RMSE_{14}$ sobre el conjunto de validación se muestran en la figura 4.6.

Tal como puede observarse, la adición del nuevo terminal principalmente afecta el Fitness encontrado para F3, correspondiente a la expresión de Nusselt. Se observa una mejora tanto en el Fitness en el conjunto de entrenamiento como en el $RMSE_{14}(Tc)$ sobre el conjunto de validación, acercando más a cero el frente de Pareto en esa dimensión.

Estos resultados son consistentes con la literatura revisada, ya que se han encontrado correlaciones entre el número de Nusselt y el número de columnas enfrentadas al fluido en

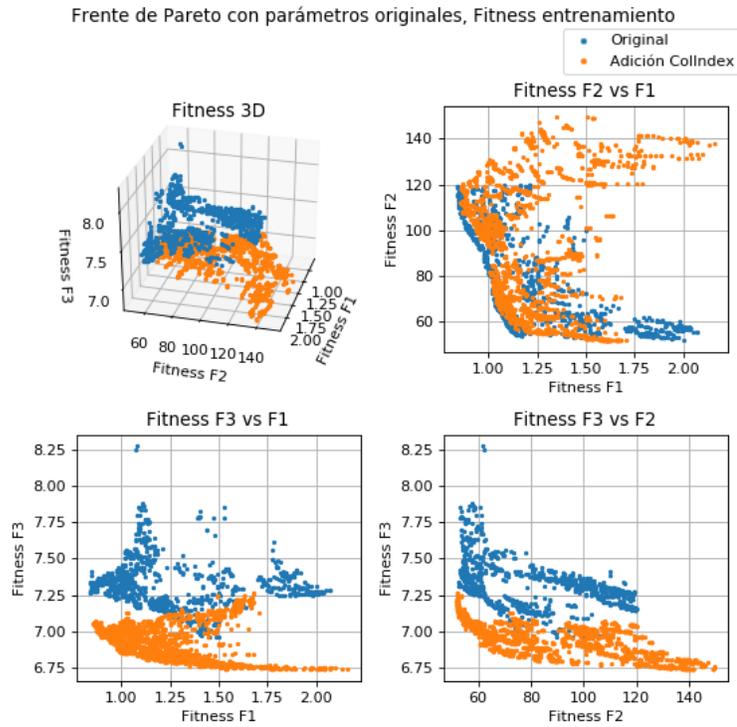


Figura 4.5: Frentes de Pareto original (azul) y como índice de columna (naranja), para el conjunto de entrenamiento.

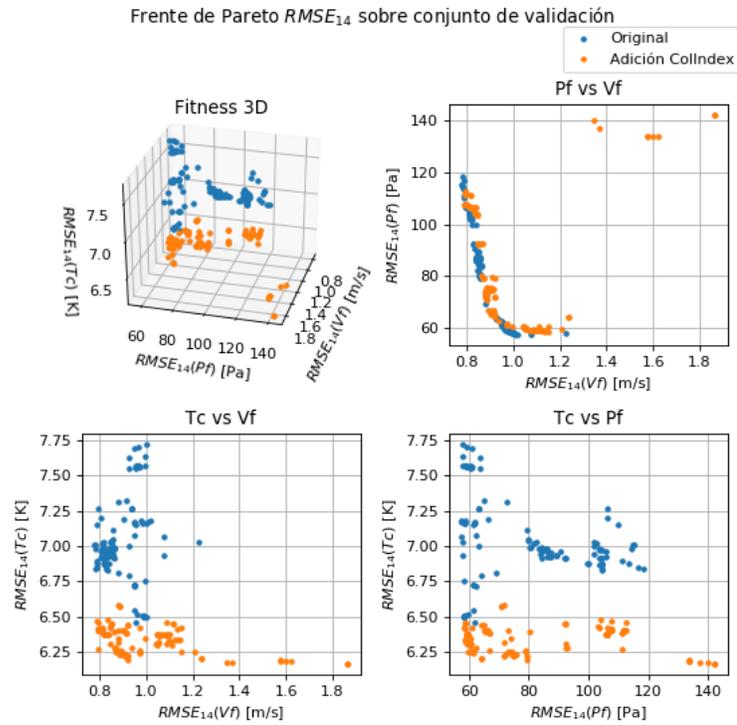


Figura 4.6: Frentes de Pareto original (azul) y con índice de columna (naranja), calculados sobre el conjunto de validación.

problemas de naturaleza similar [9].

De acuerdo a los resultados obtenidos en esta prueba, se decidió realizar las siguientes corridas del algoritmo de programación genética utilizando el modelo con el terminal nuevo de índice de columna.

4.2.2. Variación de parámetros

Se realizaron 6 corridas independientes del algoritmo de programación genética fijando el factor de regularización α en los valores 0.1, 0.25 y 0.5, y limitando el tamaño máximo de los árboles a 20 y 50 nodos. Cada corrida tardó un tiempo aproximado de 15 horas. Finalizadas las pruebas se obtuvo el promedio del tamaño de los árboles cada 10 generaciones, que se muestra en la figura 4.7.

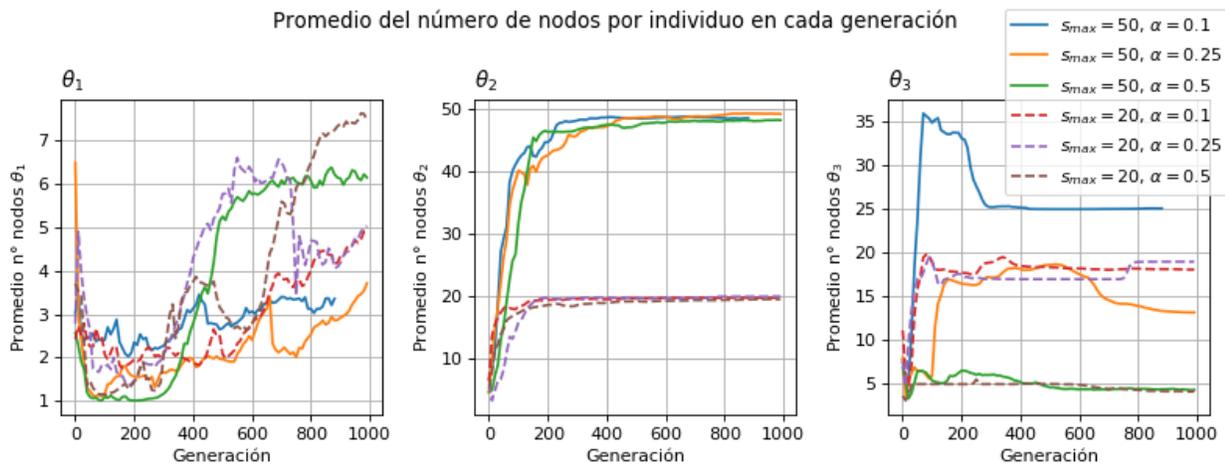


Figura 4.7: Promedio del tamaño de los árboles en función del número de generación para cada una de las 3 funciones objetivo, variando α y el tamaño máximo permitido.

Puede verse que para θ_3 (gráfico derecho) el tamaño promedio de los árboles crece rápidamente en las primeras generaciones, llegando al máximo permitido alrededor de la generación 100 para los casos con $s_{max} = 20$ y $\alpha = 0,1$ y $0,25$. Para los casos con $s_{max} = 50$, alrededor de las generaciones 300-400 los tamaños promedio comienzan a solo disminuir, debido a que se encuentran expresiones con errores equivalentes pero menor tamaño de árbol, teniendo mejor fitness.

Se observa que para θ_2 el tamaño de los árboles crece hasta llegar al máximo permitido, mientras que θ_3 tiende a estabilizarse entre las generaciones 300 y 400 y θ_1 tiene una mayor variación a partir de estos puntos, con las primeras generaciones presentando árboles con tamaño promedio pequeño.

Los diferentes comportamientos observados en los 3 tipos de árboles puede deberse a las diferentes escalas en que se encuentran los 3 errores RMSE asociados a cada árbol (θ_1/V_f , θ_2/P_f y θ_3/T_c), con V_f teniendo los valores más pequeños y P_f los más grandes. Luego el

tamaño relativo del parámetro de regularización con respecto al RMSE calculado es diferente en cada caso.

Lo anterior se podría corregir llevando los cálculos de error a una escala más similar, como por ejemplo calcular el error porcentual y a partir de éste una medida de error, tal como RMSPE (*Root Mean Square Percentage Error*). Otra opción sería utilizar diferentes factores de regularización para cada uno de las funciones de fitness, pero esto aumentaría la complejidad del algoritmo al ingresar más parámetros.

La figura 4.8 muestra que los individuos obtenidos con $\alpha = 0,5$ muestran una mayor dispersión en el espacio, principalmente en la dimensión $F1$. Esto probablemente se debe a que los cambios en el término de regularización al variar el número de nodos fácilmente llegan a ser mayores que los cambios en el $RMSE_{14}$ calculado, por lo que el algoritmo encuentra nuevos individuos pertenecientes al frente de Pareto más fácilmente al variar el número de nodos.

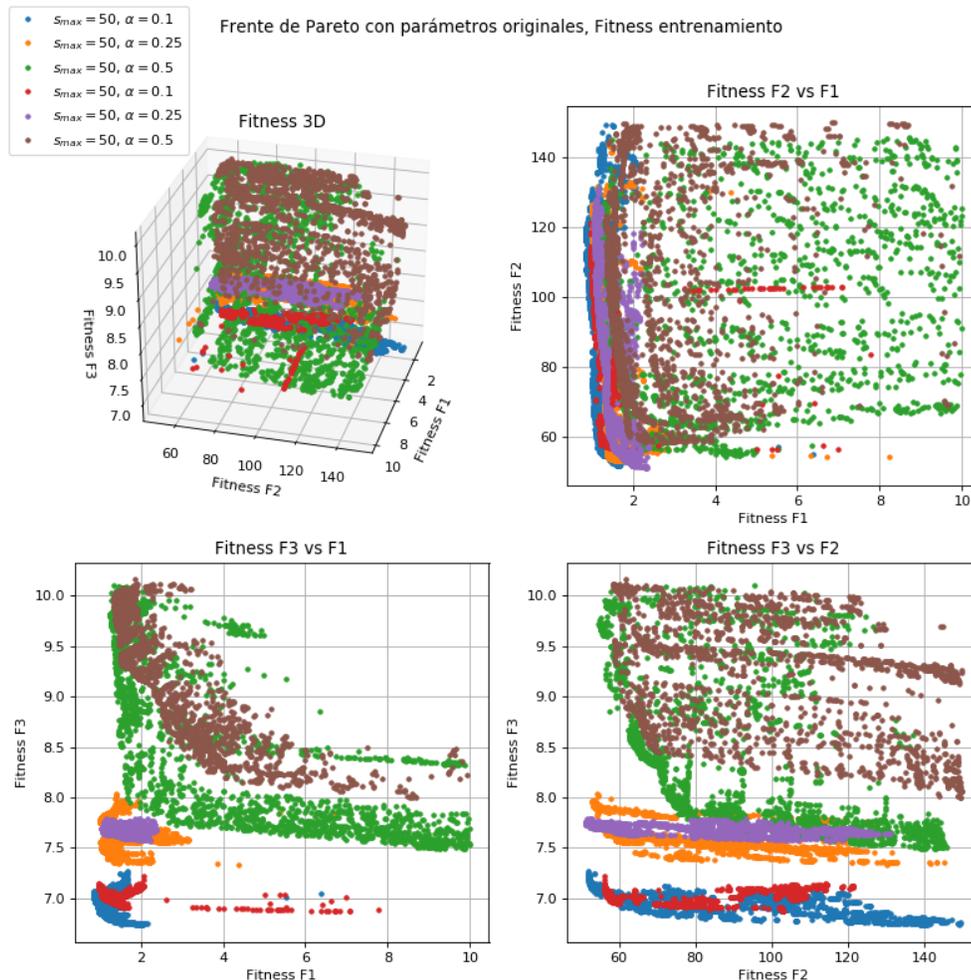


Figura 4.8: Frentes de Pareto en el conjunto de entrenamiento con diferentes α y tamaño de árboles máximo.

Al disminuir el número máximo de nodos y por lo tanto el espacio de búsqueda, para un mismo α se observa que las soluciones encontradas en entrenamiento con un mayor tamaño

máximo suelen dominar a aquellas obtenidos con un tamaño máximo de 20. Luego debido a la naturaleza no lineal de los coeficientes, puede que sea necesario un mayor número de términos en las expresiones para aproximar su comportamiento. Cabe notar que muchas de las funciones resultantes a partir de los árboles más grandes podrían ser simplificadas, por lo que es posible que dadas suficientes generaciones las corridas con tamaño máximo de 20 nodos lleguen a expresiones equivalentes a las corridas con tamaño máximo de 50 nodos.

podrían ser simplificadas a un menor tamaño, por lo que es posible que dadas suficientes generaciones, el frente de Pareto de las corridas con menor tamaño máximo sea más cercano al de aquellas con mayor tamaño.

Se calculó el $RMSE_{14}$ de cada individuo sobre el conjunto de validación y se obtuvo el frente de Pareto resultante para cada corrida, tal como se muestra en la figura 4.9. Se observa que los frentes de Pareto de las corridas con tamaño máximo de 50 nodos dominan a aquellas con máximo de 20 e igual α , con excepción de unos pocos puntos extremos con menor $RMSE_{14}(Pa)$ del grupo con tamaño máximo de 20 y $\alpha = 0,25$.

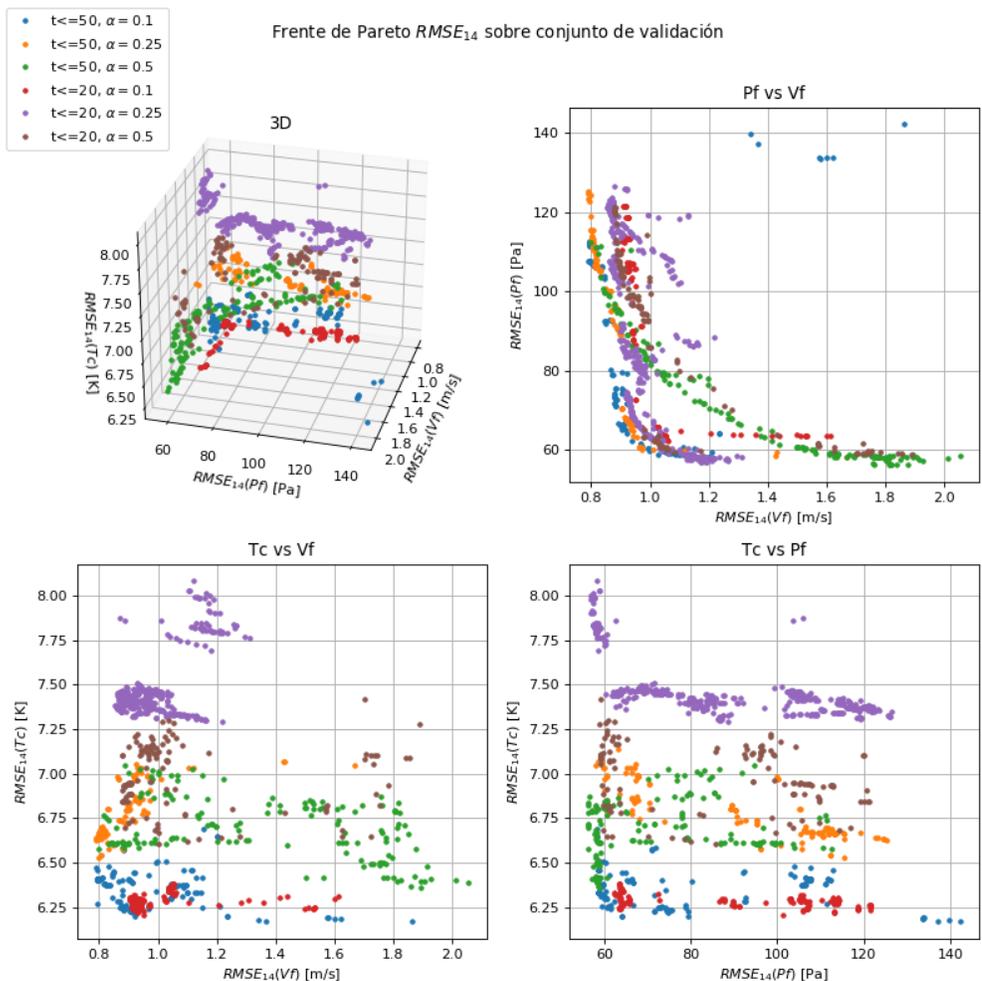


Figura 4.9: Frentes de Pareto en conjunto de validación con diferentes α y tamaño de árboles máximo.

4.2.3. Comparación de frentes obtenidos con expresiones anteriores

Obtenidos los individuos pertenecientes al frente de Pareto sobre el conjunto de validación, se procedió a calcular el $RMSE_{14}$ de los modelos de Reyes, Villa y García-Tapia, con el objetivo de poder compararlos. Gráficamente esto se puede ver en la figura 4.10. Es importante notar que para el modelo García-Tapia algunos datos no convergieron o las ecuaciones entregaban resultados inválidos, por lo que el $RMSE_{14}$ calculado solo incluye las salidas que resultaron válidas.

Puede observarse que los frentes de Pareto encontrados en este trabajo dominan completamente a las expresiones encontradas en trabajos anteriores, por lo que se puede concluir que fue posible encontrar expresiones que se ajustan de mejor manera a las salidas del modelo en ANSYS para el modelo de 102 celdas.

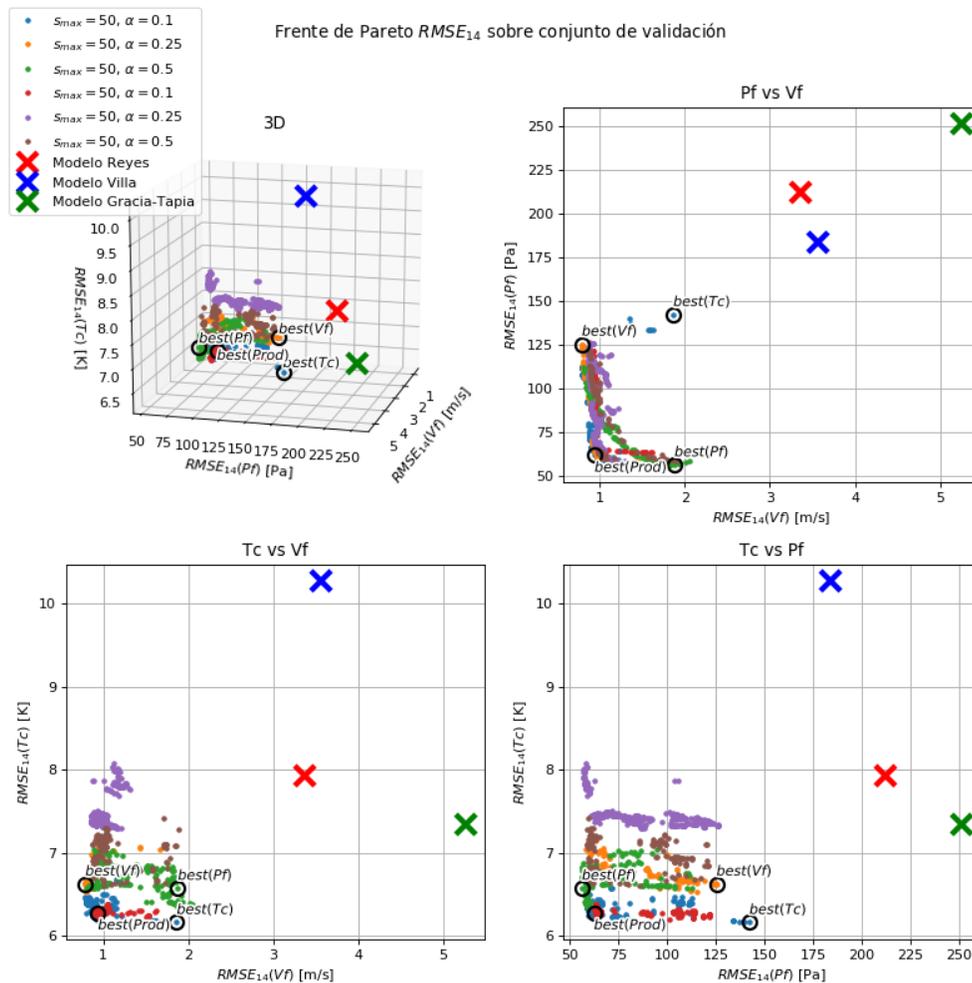


Figura 4.10: Frentes de Pareto obtenidos en este trabajo versus modelos anteriores en conjunto de validación.

Para el coeficiente de arrastre se encontró que ninguna de las expresiones evolucionadas en las distintas corridas tenían la forma esperada (dependiente del número de Reynolds [9][10]), tendiendo a ser números fijos o expresiones simples sin el número de Reynolds como término.

Esto podría deberse a que los parámetros ocupados resultaron en árboles pequeños debido a una penalización relativamente alta en comparación a las demás funciones objetivo. Entonces es posible que al variar los parámetros o la medida de Fitness evolucionen expresiones más cercanas a la esperada.

En cuanto al factor de fricción, las expresiones evolucionadas con el menor error tienen un número de nodos cercano a 50 y por lo tanto son de gran tamaño sin una simplificación eficiente. En cuanto a la forma de las expresiones, no se encontró una gran dependencia del número de Reynolds como era esperado (ecuación 2.17), con la mayoría de los árboles no poseyendo este término.

Las expresiones evolucionadas correspondientes al número de Nusselt si presentaron la forma esperada (ecuación 2.15), dependiendo del número de Reynolds y en la mayoría de los casos también de la columna en que se está evaluando.

Se escogieron 4 individuos del frente de Pareto obtenido en este trabajo según lo detallado en la sección 3.4.4, cuya posición gráfica con respecto al frente se puede observar en la figura 4.10. Los individuos resultantes fueron los siguientes:

- $best(Vf)$:

$$Coef\ arrastre = 0,9099 \quad (4.1)$$

Factor fricción =

$$(D_{fn}^2(S - (0,1940)^{(S^2)})^2(V_{mfn} - S + V_{mfn}^S + 0,8880/V_{mfn})^2 + \frac{((S - V_{mfn})^{D_{fn}})^{D_{fn}}}{(V_{mfn} * (S^2)^{(0,9598)})} - 1,7505) * (Rem^{(-0,6747)} - V_{mfn} + 0,6081) \quad (4.2)$$

$$Nusselt = Re^{(0,5092)} + \left(\frac{Re}{(Re - indCol)}\right)^{(24,5206)} \quad (4.3)$$

- $best(Pf)$:

$$Coef\ arrastre = Dfn \quad (4.4)$$

Factor fricción =

$$\frac{\left(\frac{0,672}{(D_{fn}^2 * V_{mfn})} + \frac{(6,2192 * S * (S + V_{mfn}))}{(D_{fn}^4 * Re^{((5,9696 * D_{fn}^2 * V_{mfn}))})}\right)}{(S + (Re^{(0,0968/S)})^{(0,3537/S)})} \quad (4.5)$$

$$Nusselt = Re^{(0,4708)} + 9,3523 \quad (4.6)$$

- $best(Tc)$:

$$Coef\ arrastre = 0,9601 \quad (4.7)$$

Factor fricción =

$$2Dfn - S - 2V_{mfn} - V_{mfn}^{\left(\frac{0,1523}{V_{mfn}^2}\right)} + Dfn^2 + (Dfn + V_{mfn} - V_{mfn}^{(S-1)} - (Dfn^2 - S^2)^4 + (Dfn - V_{mfn})^2)^2 + V_{mfn}^2 \quad (4.8)$$

$$Nusselt = Re^{((0,5702 * (indCol + (0,5702 * Re^{(0,9403^{indCol}}))^{(3,069 * indCol) / Re + 0,8519})) / Re + 0,4857)} \quad (4.9)$$

- $best(prod)$:

$$Coef\ arrastre = 1,4052 \quad (4.10)$$

Factor fricción =

$$\left(\frac{0,0762}{Vmfn^2}\right)^{(Vmfn^{(s^2)})} + (Dfn + \left(\frac{0,0199}{Vmfn^2} - (0,2609 * Vmfn^2)\right)^4 - ((S^2)^{(0,0422/Vmfn)} - S^2)^2) + \frac{0,0762}{Vmfn^2} - 0,0269 \quad (4.11)$$

$$Nusselt = Re^{((0,5702*(indCol+(0,5702*Re^{(0,9403^{indCol}})))/(3,069*indCol)/Re+0,8519))/Re+0,4857} \quad (4.12)$$

Las medidas de error de cada uno de estos 4 individuos en los distintos conjuntos de la base de datos se muestran en la tabla 4.1. Puede verse que los menores $RMSE_{14}$ encontrados no se alejan mucho de los valores para el punto $best(prod)$, debido a la forma que tiene el frente de Pareto encontrado, que presenta superficies casi paralelas a los ejes. Luego, el punto $best(prod)$ tiene un comportamiento cercano en cuanto al error a los mejores puntos de cada salida, tal como se puede ver en la figura 4.11.

Para el caso de la temperatura de las celdas centrales (T_c), el error en el conjunto de validación y prueba resulta ser casi siempre menor que aquel en el conjunto de entrenamiento en base a cual se evolucionaron las expresiones. Debido a lo pequeños que son los conjuntos es posible que la distribución de los datos en el espacio de entradas no sea la misma, existiendo puntos de operación que no se encuentren en los conjuntos de prueba y entrenamiento. Por lo tanto, el conjunto de entrenamiento podría poseer un mayor número de casos ‘difíciles’ para la temperatura.

Tabla 4.1: Error de los individuos elegidos para los tres conjuntos de datos.

Individuo	$RMSE_{14}$ entrenamiento			$RMSE_{14}$ validación			$RMSE_{14}$ prueba		
	Vf [$\frac{m}{s}$]	Pf [Pa]	Tc [K]	Vf [$\frac{m}{s}$]	Pf [Pa]	Tc [K]	Vf [$\frac{m}{s}$]	Pf [Pa]	Tc [K]
$best(Vf)$	0.7552	125.0300	6.8559	0.7881	125.3387	6.6291	0.8057	125.8524	6.7029
$best(Pf)$	1.5829	58.7101	8.0141	1.5295	61.0528	6.7722	1.5958	67.0525	6.846
$best(Tc)$	1.2475	137.4334	6.2870	1.3685	137.2241	6.1701	1.2646	140.0002	6.4728
$best(prod)$	0.9114	61.6566	6.5515	0.9372	62.4093	6.2823	0.9540	64.4000	6.5051

En las figuras 4.12 y 4.13 puede verse que la distribución de los errores en todos los casos se encuentran más cercanas a 0 con respecto a los modelos encontrados en trabajos anteriores, presentando una menor distancia entre el mínimo y máximo error. Se ve nuevamente que la presión tiende a subestimarse en todos los modelos, mientras que la velocidad y la temperatura muestran resultados más centrados.

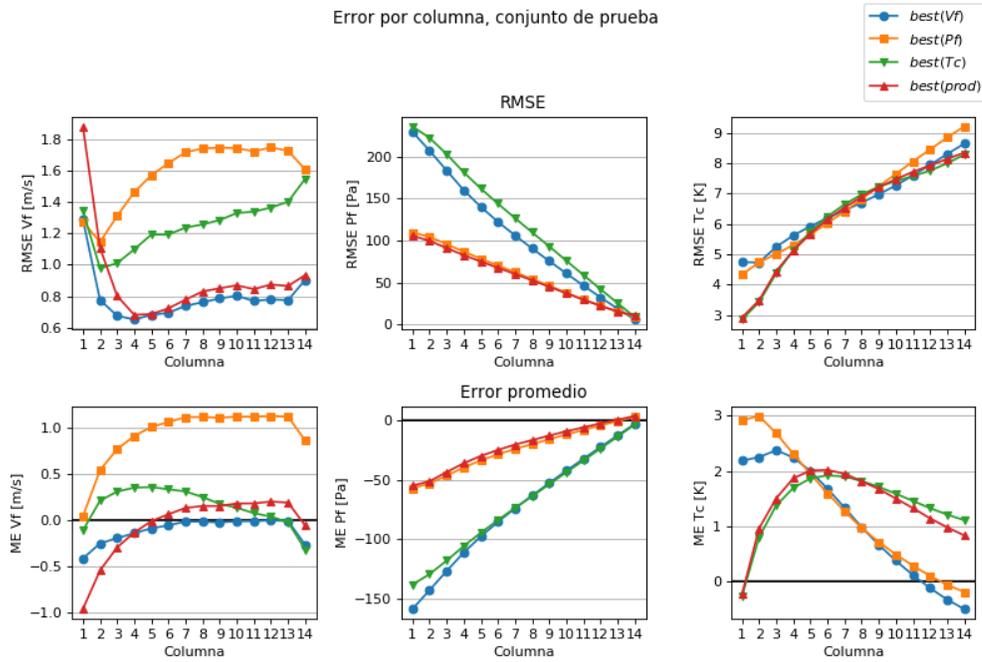


Figura 4.11: RMSE y ME por columna para los puntos elegidos sobre el conjunto de prueba.

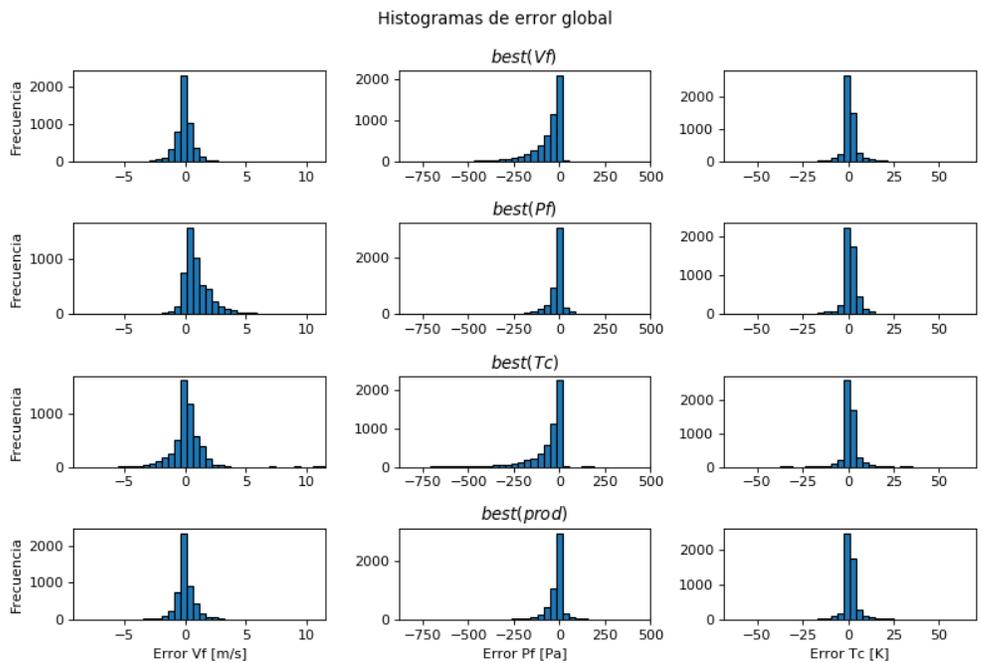


Figura 4.12: Histogramas de errores globales para los individuos elegidos sobre el conjunto de prueba.

Salida modelo vs salida ANSYS, bloques extremos

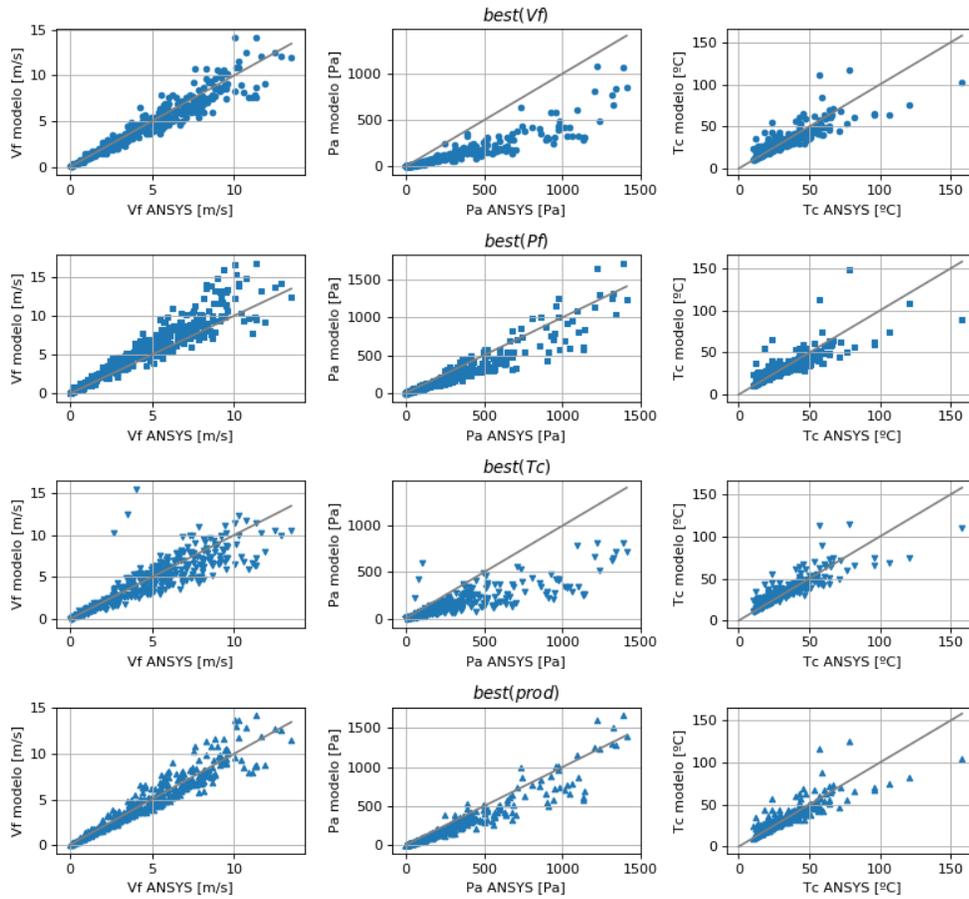


Figura 4.13: Salidas predichas vs salidas ANSYS en bloque extremos para los individuos elegidos, sobre conjunto de prueba.

4.2.4. Variación en forma del número de Nusselt

Se realizaron dos nuevas corridas de programación genética, manteniendo los mejores parámetros obtenidos anteriormente ($\alpha = 0,1$ y $s_{max} = 50$), pero variando la forma de obtener el número de Nusselt. En el primer caso se separaron los terminales ‘Re’ e ‘indCol’ en dos subárboles diferentes que son multiplicados entre sí para obtener el resultado final. Con esto se buscó simplificar la expresión final y así simplificar su comprensión. En el segundo caso se trabajó asumiendo que la dependencia del número de Nusselt con la columna tiene el mismo comportamiento que el el modelo original y multiplicando este factor por la expresión obtenida del árbol θ_3 (correspondiente al número de Nusselt).

Los resultados de las nuevas corridas se compararon con la corrida original de parámetros $\alpha = 0,1$ y $s_{max} = 50$, obteniendo los frentes de Pareto observados en la figura 4.14. Junto a lo anterior, para los individuos designados $best(prod)$ de cada corrida se calculó el error sobre el conjunto de prueba en todos los bloques, mostrado en la figura 4.15. Los árboles correspondientes a estos dos puntos pueden verse en el apéndice A.5 y A.6, y tienen las siguientes expresiones reducidas:

- $best(prod)_2$: mínimo producto entre $RMSE_{14}$ de Vf, Pf y Tc, separando $indCol$ de Re en diferentes subárboles θ_{3a} y θ_{3b} .

$$Coef\ arrastre = 1,3674 \quad (4.13)$$

$$\begin{aligned} Factor\ friccion = \\ 0,41706 * Vmfn^{-1,3830} * (Dfn^2 * Vmfn)^{-5,5319} * \frac{1}{(Vmfn + 0,1568)^2} \\ * (Dfn * (Dfn^2 * Vmfn)^{5,5319} * (Vmfn + 0,1568)^2 \\ + (0,1567 * Vmfn * (Vmfn - 0,19544) + 0,17568 * (Dfn^2 * Vmfn)^{2,7660} \\ * (Dfn - Vmfn) * (Vmfn + 0,1568))^2 \end{aligned} \quad (4.14)$$

$$Nusselt = (Re^{0,8086} + (16395,1769 * (0,9683)^{(2*Re)}))^{0,5842} * (0,8583^{colIndex} + 0,9822) \quad (4.15)$$

- $best(prod)_3$: mínimo producto entre $RMSE_{14}$ de Vf, Pf y Tc, multiplicando el árbol θ_3 por $Z(indCol)$ para obtener el número de Nusselt.

$$Coef\ arrastre = 1,4294 \quad (4.16)$$

$$\begin{aligned} Factor\ friccion = \\ \frac{(Dfn - S)}{((Dfn - Vmfn)^4)^{Vmfn} - (Dfn - \frac{0,5329}{Vmfn})^2} - \\ \frac{(Dfn - S - \frac{0,2359}{Vmfn} + \frac{Vmfn * (S * Vmfn - 0,6473)}{(Dfn - Vmfn)} + 0,2359)}{(S * Vmfn)} + 1 \end{aligned} \quad (4.17)$$

$$Nusselt = Z(indCol) * 0,8923 * \frac{(Rem^{0,76} + Rem) * ((Rem + Rem^{(\frac{104,0438}{Rem})})^{0,5058} + 0,1969)}{Rem} \quad (4.18)$$

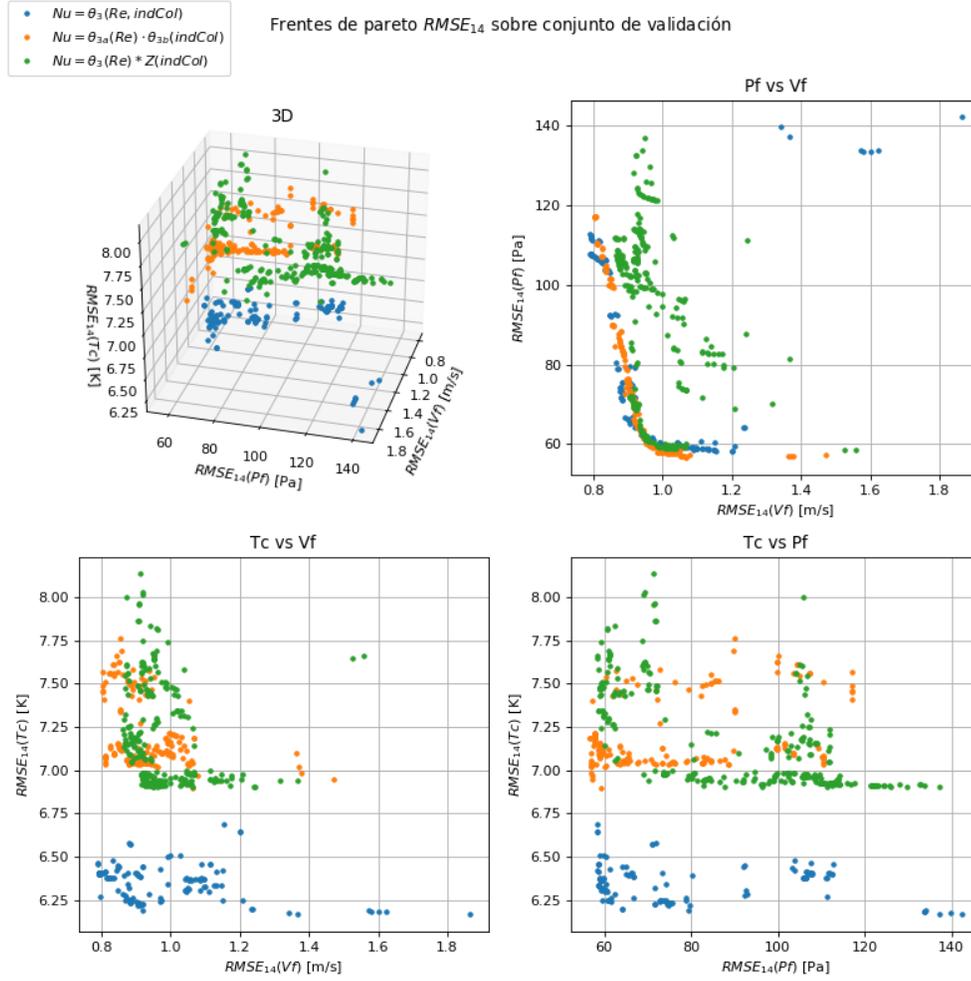


Figura 4.14: Frentes de Pareto variando la forma de calcular el número de Nusselt.

En la figura 4.14 se puede observar que la mayor diferencia entre los nuevos frentes de Pareto obtenidos (en naranja y verde) y el frente obtenido anteriormente (en azul) está en el error en la temperatura de las celdas centrales, teniendo aproximadamente $0,7 [K]$ de diferencia para $RMSE_{14}(Tc)$ en los puntos con similares errores en Pf y Tc . Esto probablemente se debe a que al separar el cálculo de Nusselt en partes dependientes del número de Reynolds y de la columna evaluada, se pierden interdependencias entre ambas variables que si se encuentran en los árboles evolucionados juntando ambos términos.

Por otra parte, las expresiones correspondientes al número de Nusselt de los nuevos individuos tienen una forma más simple en comparación con aquellas obtenidas juntando ambas variables, facilitando su interpretación. En cuanto a la dependencia del índice de columna, se observa que para el individuo $best(prod)_2$ el número de Nusselt tiene un comportamiento inverso al esperado, siendo decreciente en función de $indCol$ en vez de creciente como lo es al asumir $Z(indCol)$.

Es posible que las condiciones del banco de celdas utilizado sea diferente a los estudiados en [9], haciendo que no se cumplan las mismas correlaciones entre el número de Nusselt y el número de columnas de la configuración, por lo que sería necesario un estudio del caso

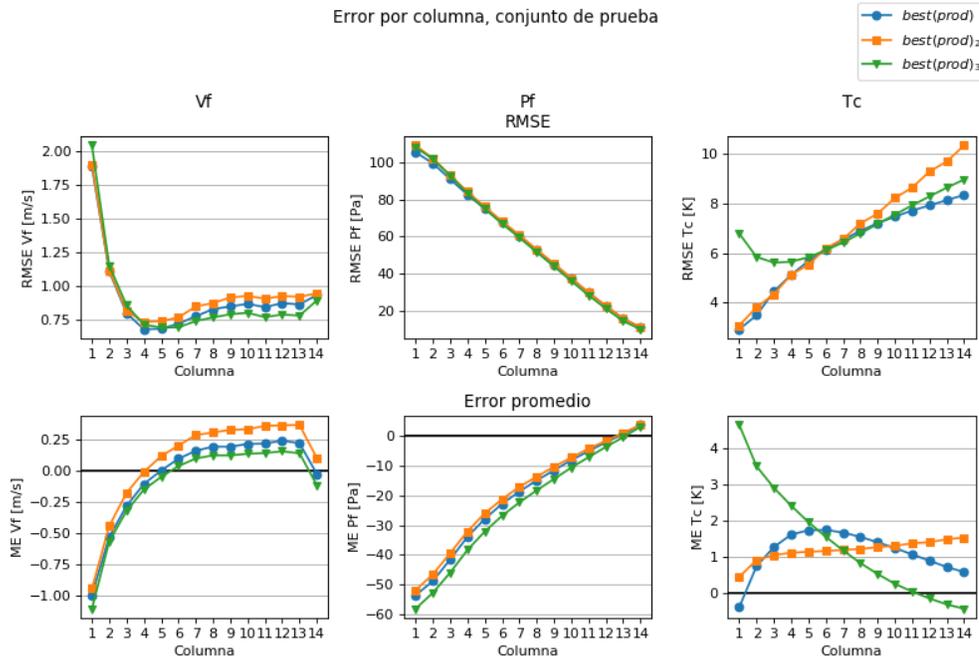


Figura 4.15: RMSE y ME por columna para los puntos $best(prod)$, $best(prod)_2$ y $best(prod)_3$ sobre el conjunto de prueba.

específico visto en este trabajo. Sin embargo, también es posible que las simplificaciones realizadas en el modelo fenomenológico hagan que los parámetros deban comportarse de manera diferente para lograr una salida más cercana a la referencia.

En la figura 4.15 se observa que los 3 individuos estudiados muestran errores similares tanto en Vf como en Pf, estando la mayor diferencia en Tc. Como es de esperarse al ver los frentes de Pareto (figura 4.14), el individuo $best(prod)$ tiene el mejor desempeño con un error menor que los otros 2 en casi todas las columnas y creciendo con menor inclinación en los bloques finales. Es de notar que $best(prod)_2$ tiene un mejor desempeño en las primeras columnas en relación a $best(prod)_3$, pero que el comportamiento se invierte a partir de la columna 6. Esto probablemente se debe a que al asumir la dependencia del número de Nusselt con respecto a la columna evaluada, el modelo debe sobreestimar la temperatura en las primeras columnas para disminuir el error total en la segunda mitad de la configuración.

4.2.5. Validez de modelos sobre configuraciones de 53 y 151 celdas

Se calculó el $RMSE_7$ y el $RMSE_{21}$ para Vf, Pf y Tc sobre las bases de datos de prueba de 53 y 151 celdas, respectivamente. Los individuos analizados fueron aquellos obtenidos de las diferentes corridas de programación genética entrenadas sobre la base de 102 celdas. Los frentes de Pareto obtenidos sobre la base de datos de 53 celdas se pueden ver en la figura 4.16 y aquellos sobre la base de 151 celdas en la figura 4.17.

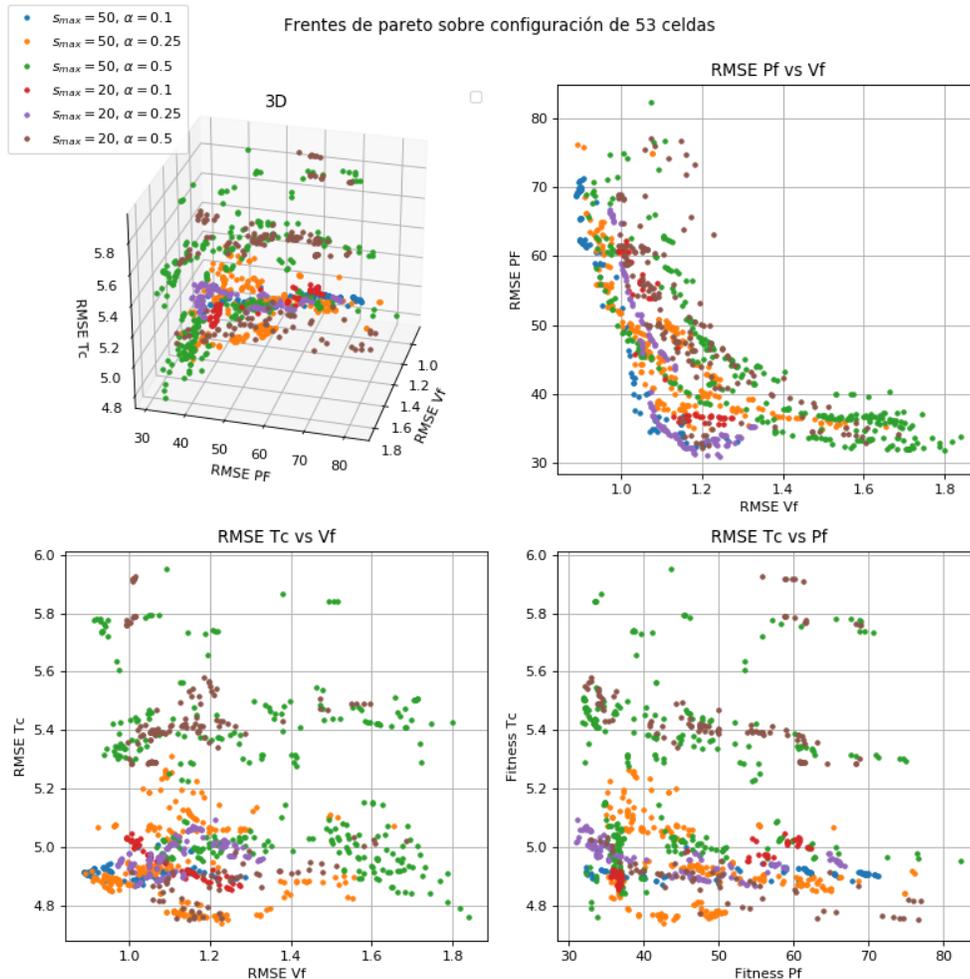


Figura 4.16: Frentes de Pareto sobre base de datos de 53 celdas.

Se observa que, a diferencia de lo visto sobre la base de datos de 102 celdas, las nuevas bases de datos en el caso $s_{max} = 50, \alpha = 0,1$ (en azul en ambas figuras) ya no dominan a todos los otros frentes con diferentes configuraciones de s_{max} y α . En particular en la figura 4.17, se ve que existen claras diferencias entre los frentes de Pareto para el error en Tc, con las corridas utilizando $s_{max} = 20$ y $\alpha = [0,1, 0,25]$ mostrando errores $RMSE_{21}$ de 0.5 a 1 [K] menores que el mejor individuo con $s_{max} = 50$.

Las diferencias entre los comportamientos observados sobre la base de datos de 102 celdas y las nuevas de 53 y 151 celdas probablemente se debe a que las corridas con árboles de tamaño máximo de 50 nodos tienen un mayor grado de ajuste sobre los datos de entrenamiento de 102

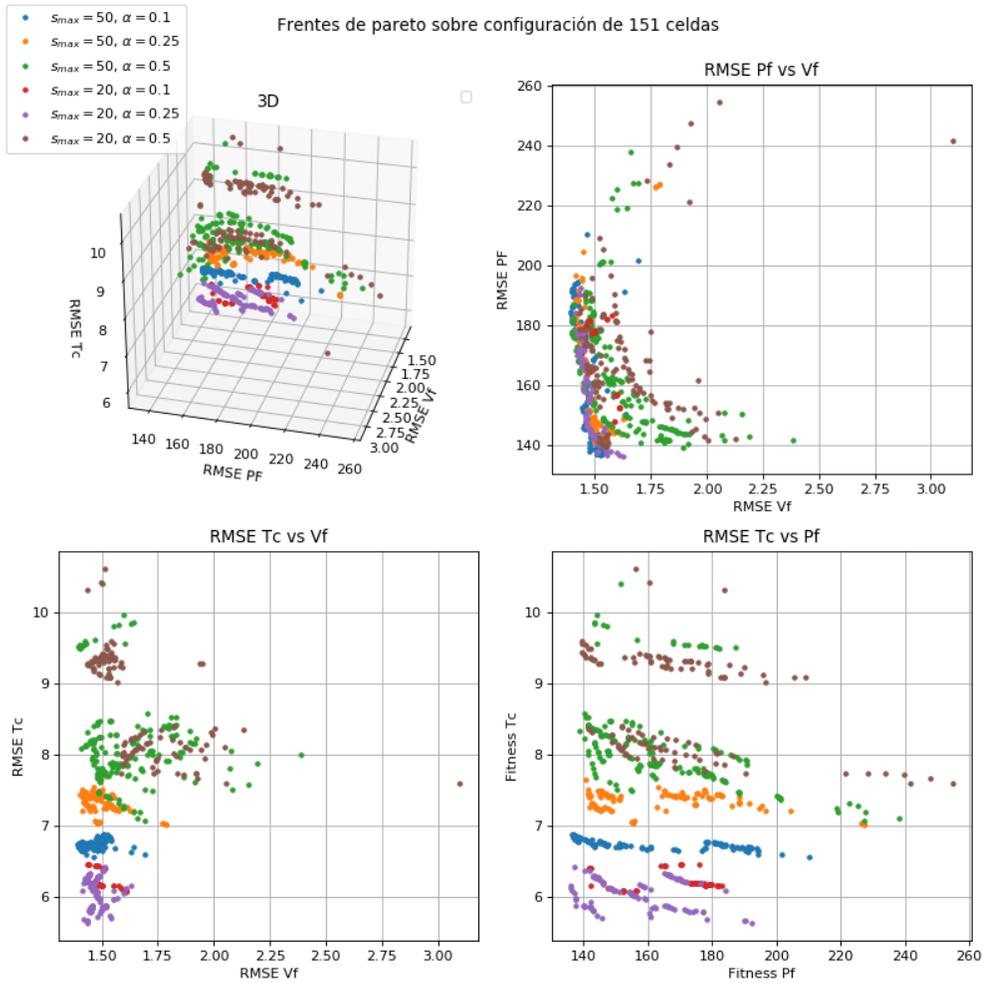


Figura 4.17: Frentes de Pareto sobre base de datos de 151 celdas.

celdas que aquellas con un tamaño máximo de 20 nodos, haciendo que los modelos obtenidos pierdan validez al extrapolar a una configuración con un mayor número de celdas.

En la figura 4.18 se puede observar el error del individuo *best(prod)* sobre las bases de datos de prueba de 53, 102 y 151 celdas. Se observa que para el caso de 53 celdas el RMSE por columna es muy similar al error en el caso de 102 celdas teniendo valores extremadamente cercanos, tanto para Vf y Tc (viendo de izquierda a derecha) como para Pf (visto de la última columna de cada configuración hacia atrás).

Se observa que al pasar al caso de 151 celdas la diferencia entre los errores se hace notoria, con los errores tanto en Vf como en Pf aumentando considerablemente en las primeras columnas, indicando que al aumentar el número de celdas de la configuración se dan condiciones que no se encuentran al entrenar el modelo sobre 102 celdas.

Es probable que los modelos obtenidos entrenando para un número N de celdas sean similarmente válidos al evaluar sobre configuración con $< N$ celdas, ya que las configuraciones más pequeñas estarían ‘contenidas’ en las configuración más grandes. Por otra parte, al pasar a configuraciones con $> N$ celdas, se encontrarían entradas a los modelos que no son posibles

con configuraciones más pequeñas, haciendo que los modelos entrenados no sean válidos en todo el rango.

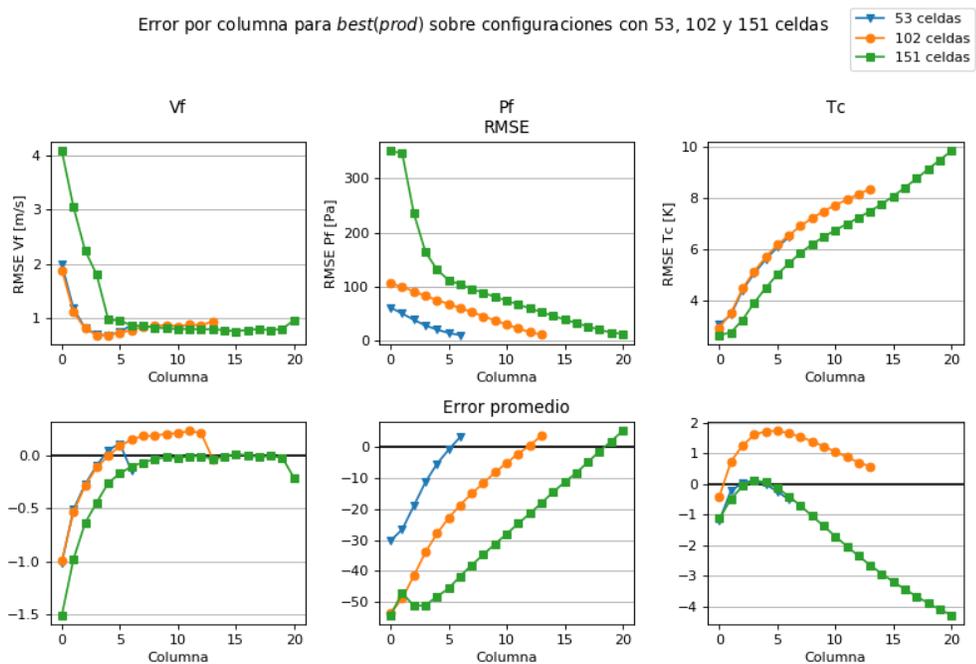


Figura 4.18: RMSE y ME por columna para el individuo *best(prod)* sobre bases de datos de 53, 102 y 151 celdas.

Conclusión

Este trabajo de memoria presenta un avance en el ajuste de un modelo fenomenológico de bancos de baterías, extendiendo el modelo a una configuración con un mayor número de celdas que las analizadas en trabajos anteriores. Para el ajuste del modelo fenomenológico se utilizó programación genética más optimización multiobjetivo para encontrar expresiones del coeficiente de arrastre, el factor de fricción y el número de Nusselt que se ajusten de mejor manera al problema analizado.

Para realizar este trabajo se construyó una base de datos con diferentes configuraciones de entradas simulando un modelo de 102 celdas en el software CFD ANSYS, para ser usado como estándar dorado de comparación con el modelo fenomenológico. Sin embargo, debido a la complejidad del modelo y restricciones de tiempo, no fue posible obtener la cantidad de puntos de datos pensados en un principio, terminando con conjuntos de menor tamaño al deseado.

Se evaluaron directamente las expresiones encontradas en trabajos anteriores para un modelo de 5 celdas sobre una configuración de 102 celdas, mostrando que los errores son considerablemente mayores en esta última. Con esto se muestra que al expandir el número de celdas del banco de baterías, el comportamiento de los parámetros analizados puede cambiar, necesitando encontrar nuevas expresiones que se adapten de mejor manera a estos requerimientos.

Utilizando programación genética más optimización multiobjetivo se evolucionaron simultáneamente las expresiones de coeficiente de arrastre, el factor de fricción y el número de Nusselt, logrando encontrar un frente de Pareto del problema. En una primera etapa se consideró la inclusión de un nuevo terminal a la expresión del número de Nusselt, correspondiente a la columna en que se evalúa, el cual había sido obviado en trabajos anteriores debido a la utilización de un modelo de un solo bloque. Esta adición mostró ser una mejora, pudiendo obtenerse expresiones con un menor error en el modelo de 102 celdas.

Se variaron parámetros del algoritmo de programación genética con el objetivo de controlar el crecimiento de los árboles y ver el efecto en los frentes de Pareto resultantes. Se mostró que un menor factor de regularización α entrega en árboles que se ajustan mejor al comportamiento de los parámetros analizados. Una de las falencias de las métricas de fitness escogidas, es que el rango de los errores de V_f , P_f y T_c son muy diferentes, por lo que aplicar el mismo factor de regularización a los 3 errores no produce los mismos resultados. Esto podría mejorarse llevando los errores a la misma escala (error porcentual, por ejemplo), o utilizando diferentes valores α para cada árbol.

Las expresiones encontradas para el coeficiente de arrastre no mostraron gran similitud con las expresiones teóricas esperadas, entregando generalmente números fijos o expresiones de pequeño tamaño. Para el factor de fricción se evolucionaron expresiones de gran tamaño con difícil simplificación, sin presentar en la mayoría de los casos el número de Reynolds como parámetro tal como era esperado. En cuanto al número de Nusselt, las expresiones encontrados son de forma parecida a la del modelo original.

Se analizaron detalladamente algunas de las expresiones encontradas, sobre la base de datos de prueba con 102 celdas, verificando que se ajustan de mejor manera al modelo que las expresiones desarrolladas en trabajos previos, pudiendo disminuir el RMSE sobre todas las salidas de 3,9361[m/s], 215,1223[Pa] y 7,4035[K] a 0,954[m/s], 64,4[Pa] y 6,5051[K], respectivamente.

Se estudiaron posibles variantes en la forma de calcular el número de Nusselt, subdividiendo sus variables en 2 partes o asumiendo que el comportamiento en cuanto a la columna obedece los mismos supuestos realizados en el modelo fenomenológico original de Jorge Reyes. En ambos casos, se mostraron diferentes comportamientos en función de la columna y mayores errores que en las pruebas originales, siendo posible conjeturar que el número de Reynolds y la columna evaluada no se pueden trabajar como elementos independientes en la expresión de Nusselt.

Al estudiar la validez de los modelos entrenados sobre bases de datos de 53 y 151 celdas, se observó que el comportamiento observado en la base de datos de 102 celdas se mantiene al pasar a una configuración con menor número de celdas, manteniendo la validez del modelo. Sin embargo, al pasar a una configuración con un mayor número de celdas, se observaron errores muy grandes en las columnas adicionales tanto en la velocidad de flujo como en la presión de fluido, pudiendo concluir que configuraciones más grandes presentan condiciones que no se encuentran en configuraciones con un menor número de celdas. Luego, sería necesario entrenar nuevos modelos si se desea disminuir el error para bancos con un mayor número de celdas o encontrar una relación que permita extrapolar a partir de los modelos más pequeños.

Como trabajo futuro se propone:

- Expandir la base de datos con un mayor número de configuraciones, con el objetivo de describir de mejor manera el espacio de entradas y obtener resultados más consistentes entre los conjuntos de entrenamiento, validación y prueba.
- Modificar el cálculo de los fitness para que se encuentren en la misma escala relativa, obteniendo resultados más consistentes para cada árbol al cambiar los parámetros del algoritmo.
- Estudiar otras posibles dependencias de las expresiones analizadas, ya que es posible que el comportamiento de los parámetros sea mejor descrito mediante términos no considerados actualmente, como separación entre celdas e índice de columna en el cálculo del coeficiente de arrastre.

Bibliografía

- [1] Francisco Villa Suárez. *Ajuste de modelo fenomenológico de celdas de baterías usando algoritmos evolutivos*. Santiago, Chile: Universidad de Chile - Facultad de Ciencias Físicas y Matemáticas, 2015. Disponible en <http://www.repositorio.uchile.cl/handle/2250/132913>.
- [2] John R. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, Cambridge, MA, USA, 1992.
- [3] Martin Klammer, J. Nikolaj Dybowski, Daniel Hoffmann, and Christoph Schaab. Pareto optimization identifies diverse set of phosphorylation signatures predicting response to treatment with dasatinib. *PLOS ONE*, 10(6):e0128542, jun 2015.
- [4] Germán Garcia and Nicolás Tapia. Evolución del modelo fenomenológico de bancos de baterías. Reporte interno DIE, 2018.
- [5] A. Thielmann, A. Sauer, R. Isenmann, M. Wietschel, and P. Plotz. Product roadmap lithium-ion batteries 2030. Technical report, Farunhofer Institute for Systems and Innovation Research ISI, 2012.
- [6] Ahmad A. Pesaran, G. H. Kim, M. Keyser, and National Renewable Energy Laboratory (U.S.). *Integration issues of cells into battery packs for plug-in and hybrid electric vehicles*. National Renewable Energy Laboratory Golden, CO, 2009.
- [7] John F. Wendt. *Computational Fluid Dynamics - An Introduction*. Springer, Berlin, 3 edition, 2009.
- [8] J. E. Reyes Marambio. Reporte de modelo paramétrico. Reporte interno, 2014.
- [9] A. Žkauskas. Heat transfer from tubes in crossflow. volume 18 of *Advances in Heat Transfer*, pages 87 – 159. Elsevier, 1987.
- [10] W.H. Bell. Turbulence vs drag—some further considerations. *Ocean Engineering*, 10(1):47 – 63, 1983.
- [11] Riccardo Poli, William B. Langdon, and Nicholas Freitag McPhee. *A field guide to genetic programming*. Published via <http://lulu.com> and freely available at <http://www.gp-field-guide.org.uk>, 2008. (With contributions by J. R. Koza).

- [12] E. Alfaro-Cid, J. J. Merelo, F. Fernández de Vega, A. I. Esparcia-Alcázar, and K. Sharman. Bloat control operators and diversity in genetic programming: A comparative study. *Evol. Comput.*, 18(2):305–332, June 2010.
- [13] Carlos A. Coello Coello, Gary B. Lamont, and David A. Van Veldhuizen. *Evolutionary Algorithms for Solving Multi-Objective Problems Second Edition*. Springer, 2007.
- [14] Sean Luke. ECJ evolutionary computation library, 1998. Available for free at <http://cs.gmu.edu/~eclab/projects/ecj/>.
- [15] Sean Luke and Liviu Panait. Lexicographic parsimony pressure. In *Proceedings of the 4th Annual Conference on Genetic and Evolutionary Computation*, GECCO'02, pages 829–836, San Francisco, CA, USA, 2002. Morgan Kaufmann Publishers Inc.
- [16] D. P. Incropera, F. P. & Dewitt. *Fundamentals of heat and mass transfer*. Wiley, 5th edition, 2002.
- [17] M.A. Mehrabian. Heat transfer and pressure drop characteristics of cross flow of air over a circular tube in isolation and/or in a tube bank. *Arabian Journal for Science and Engineering*, 32:365–376, 10 2007.

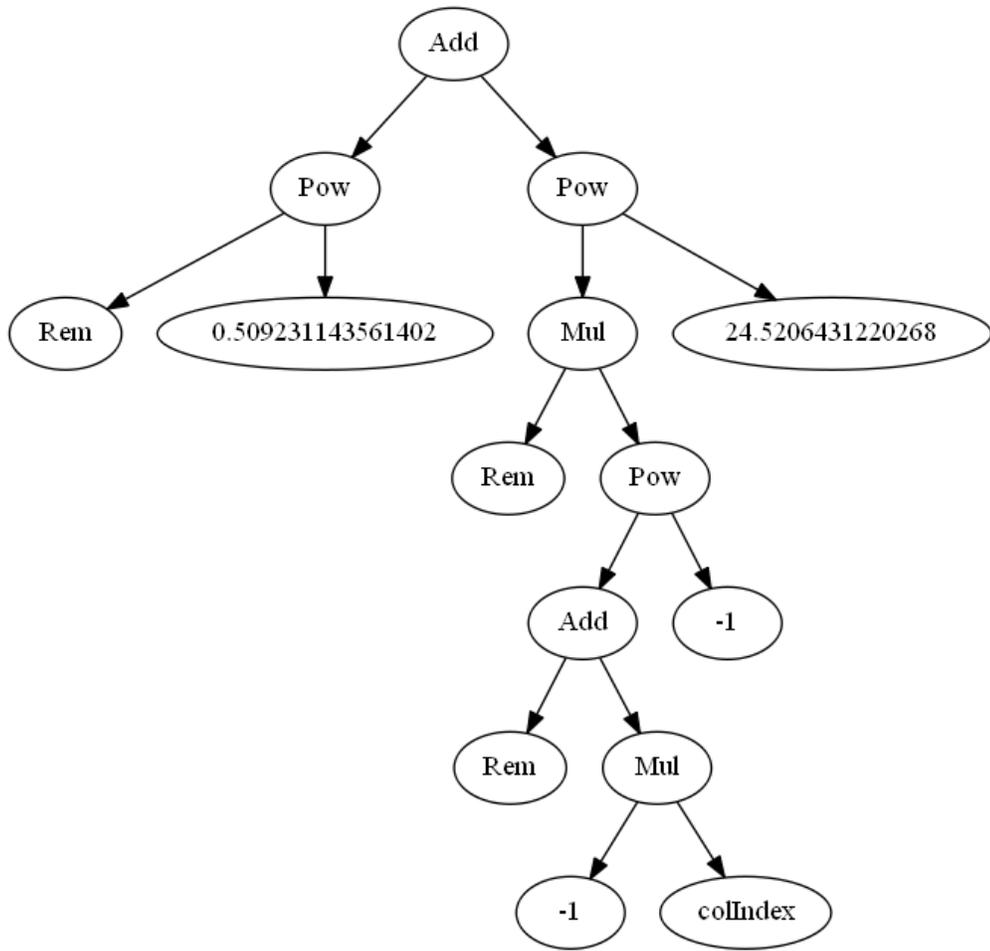


Figura A.3: Árbol de Nusselt, individuo $best(Vf)$

A.2. $best(Pf)$



Figura A.4: Árbol de coeficiente de arrastre, individuo $best(Pf)$

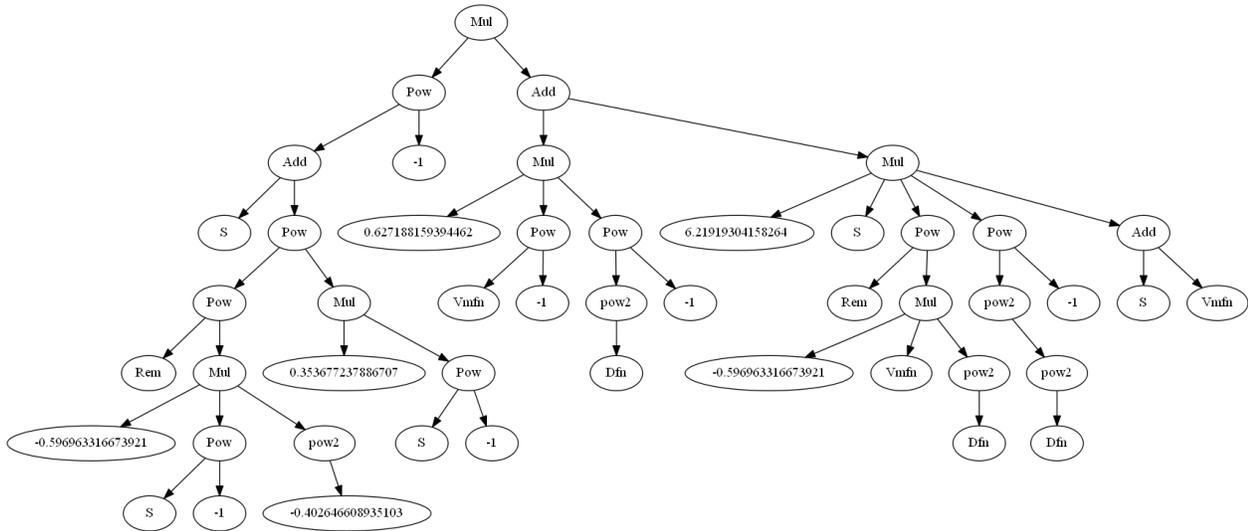


Figura A.5: Árbol de factor de fricción, individuo $best(Pf)$

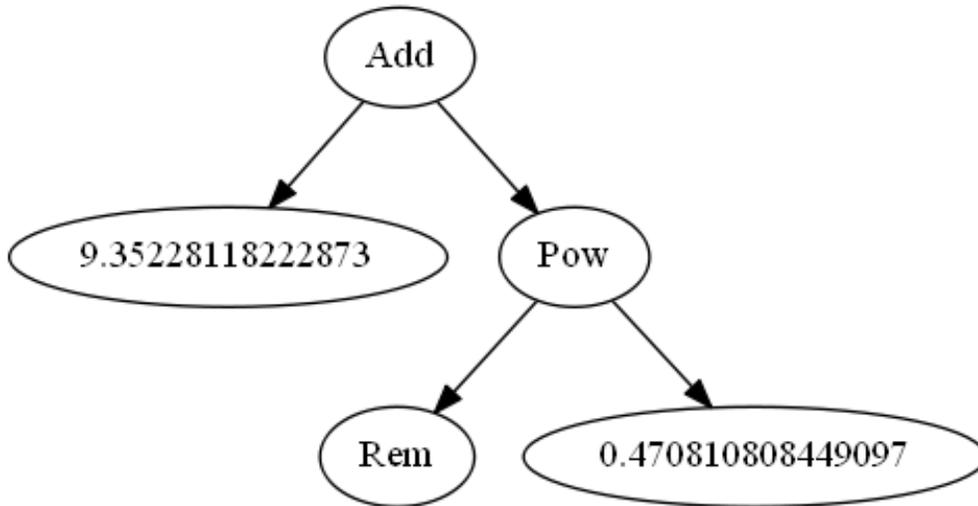


Figura A.6: Árbol de Nusselt, individuo $best(Pf)$

A.3. $best(Tc)$

0.960119305672208

Figura A.7: Árbol de coeficiente de arrastre, individuo $best(Tc)$

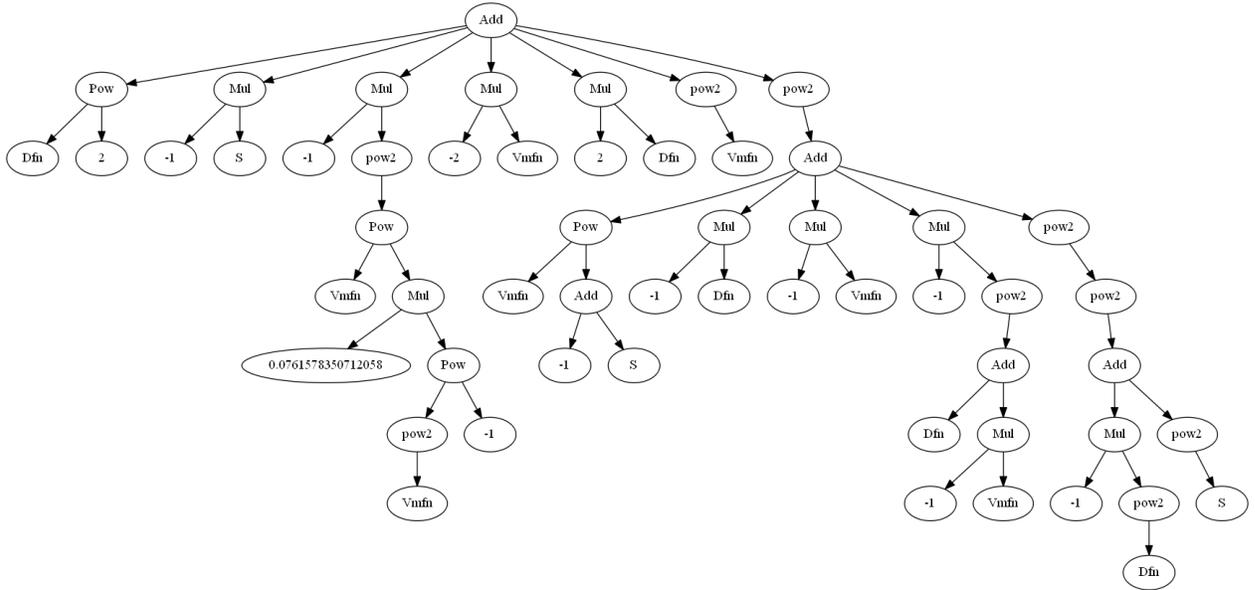


Figura A.8: Árbol de factor de fricción, individuo $best(Tc)$

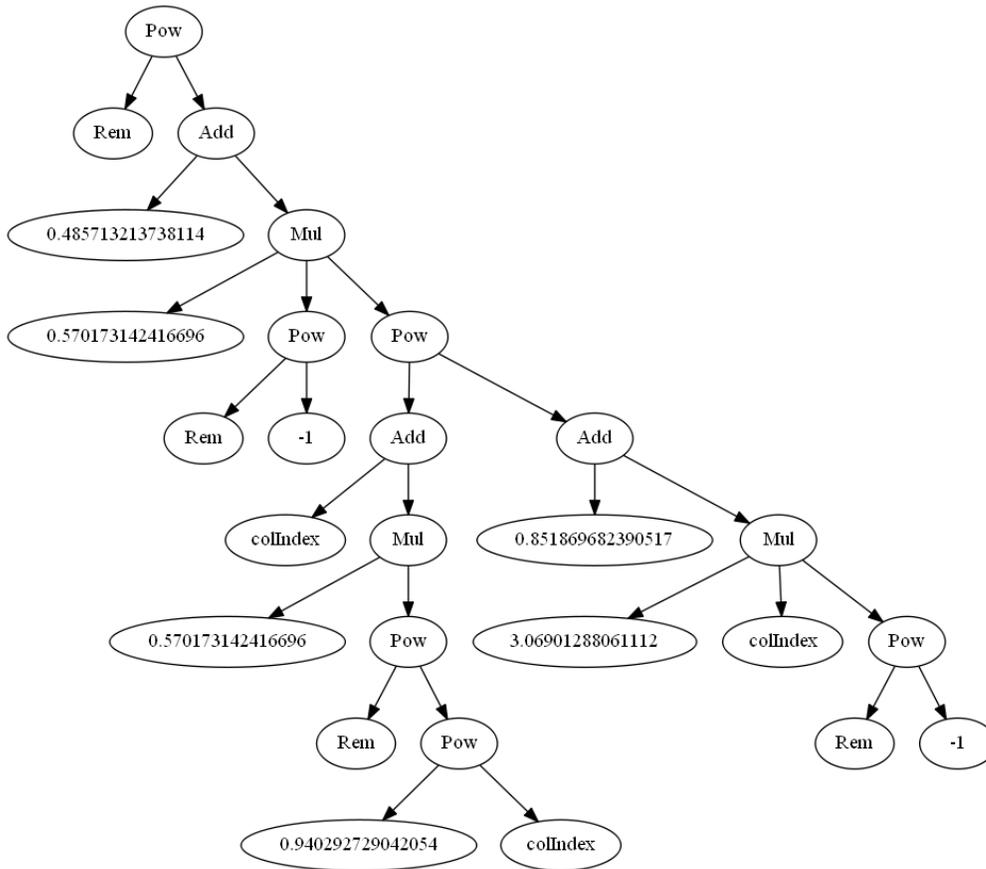


Figura A.9: Árbol de Nusselt, individuo $best(Tc)$

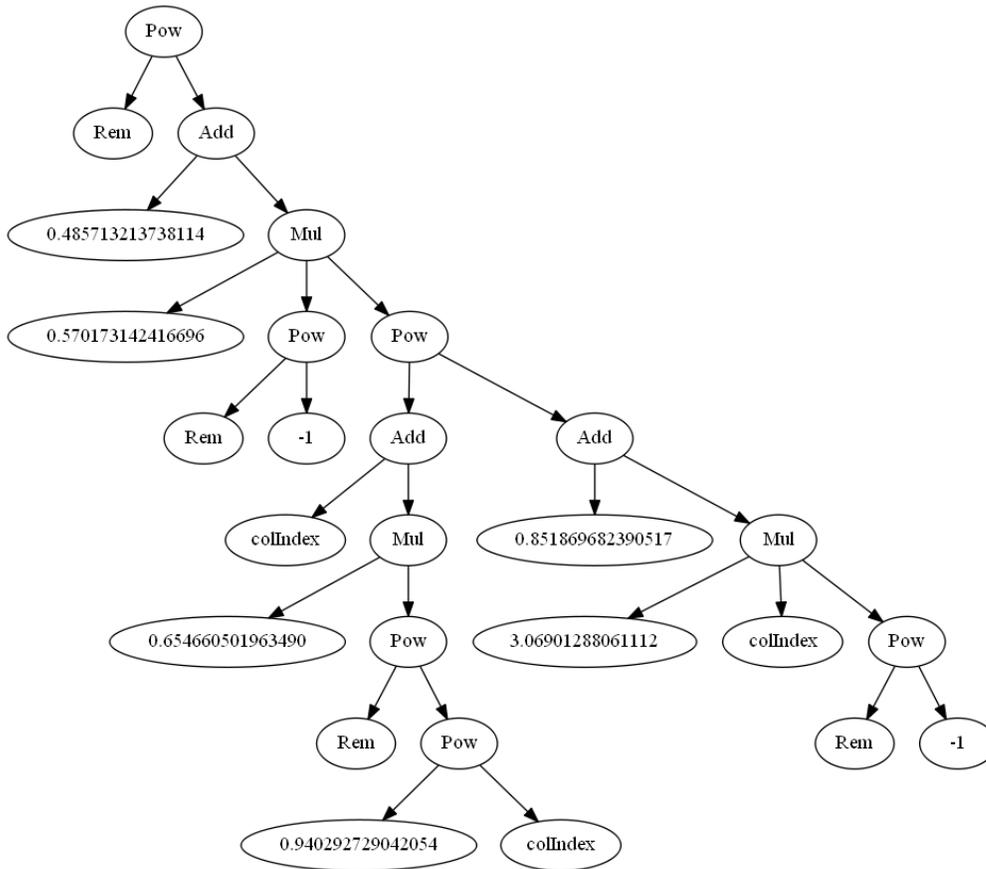


Figura A.12: Árbol de Nusselt, individuo $best(prod)$

A.5. $best(prod)_2$

1.36736654883127

Figura A.13: Árbol de coeficiente de arrastre, individuo $best(prod)_2$

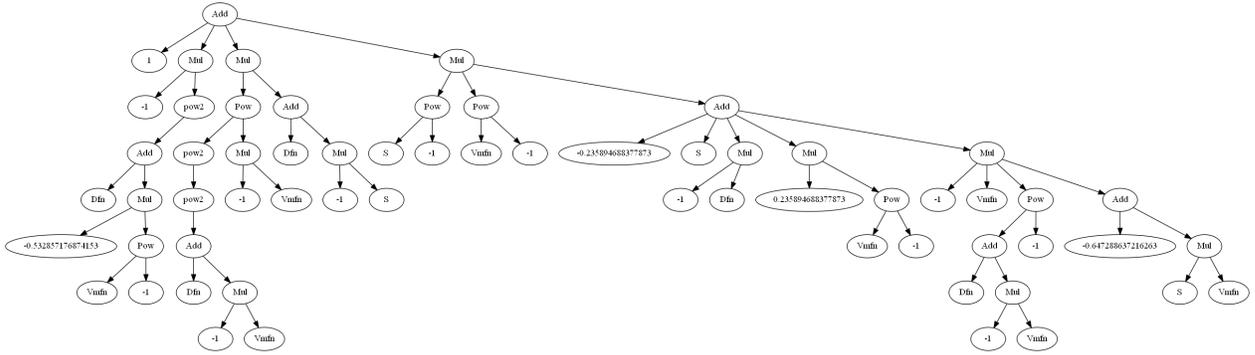


Figura A.14: Árbol de factor de fricción, individuo $best(prod)_2$

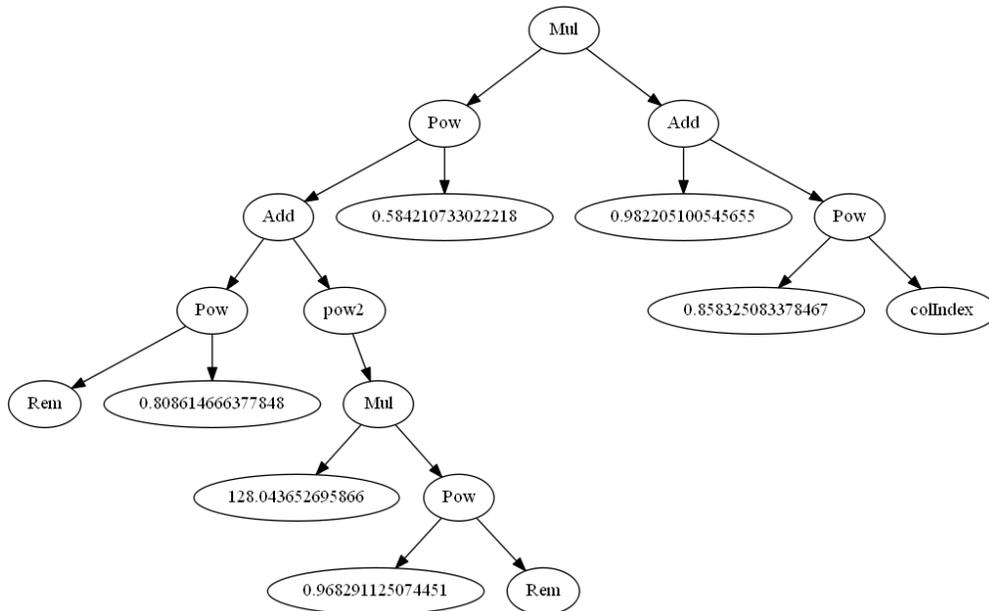


Figura A.15: Árbol de Nusselt, individuo $best(prod)_2$

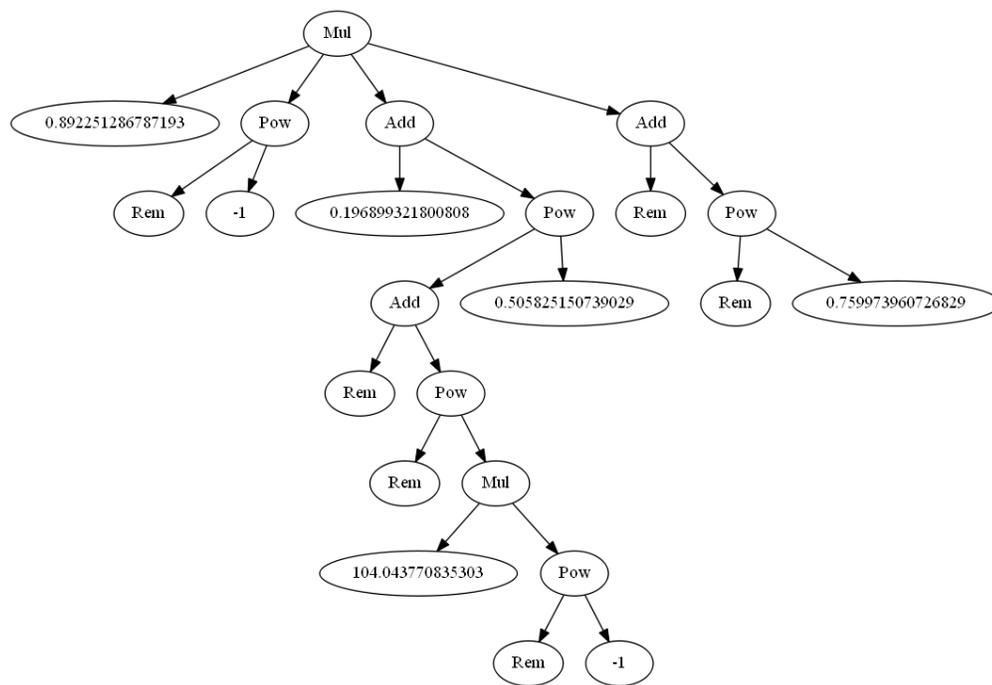


Figura A.18: Árbol de Nusselt, individuo $best(prod)_3$