

Optimización matemática en procesos industriales. Aplicación al estudio de aparatos de tratamiento de alimentos por campos eléctricos

Alfonso Fernández Bes

Trabajo Fin de Máster

Máster en Tratamiento Estadístico Computacional de la Información

Tutores: Benjamin Ivorra
Ángel M Ramos



Agradecimientos

En primer lugar quiero agradecer la ayuda y apoyo prestados a los tutores de este trabajo, Benjamin Ivorra y Ángel M. Ramos, que junto a Javier Yañez me han introducido en el mundo de la Optimización y de la Matemática Aplicada.

Agradezco la ayuda prestada, comentarios y consejos de los profesores que me han impartido clase en este máster y me han otorgado conocimientos tanto académicos como profesionales para mi completo desarrollo en la universidad.

Quiero agradecer el trabajo conjunto con todos mis compañeros en los trabajos realizados, cabe destacar a mis compañeras Noelia y Victoria por su constante apoyo e interés durante todo el curso académico.

Por último, en el apartado personal, mi gratitud a mi padres y hermanos por apoyarme en todo momento para seguir adelante, sobre todo durante este año tan duro. También agradecer a mi pareja Irene que ha sabido entender mi falta de dedicación durante estos últimos meses y ha estado ahí dándome ánimo en todo momento.

Resumen

El proyecto que se plantea en este Trabajo de Fin de Máster se centra en el estudio de diferentes algoritmos metaheurísticos de optimización, implementándose en MATLABTM (The Mathworks Inc., Natick, MA, USA). Los modelos planteados en este trabajo son una herramienta muy útil para el diseño de equipos industriales y optimización de procesos.

Se estudiará en particular un proceso industrial real en el ámbito del tratamiento de alimentos mediante técnica de campos eléctricos. Se quiere estudiar la forma geométrica óptima del aislante eléctrico y evaluar el modelo con respecto al campo eléctrico uniforme. Para la modelización física del problema se utilizará el programa COMSOL MultiphysicsTM (COMSOL AB, Stockholm, Sweden).

El estudio se enmarca en el ámbito de la optimización matemática de problemas continuos, pretendiendo así desarrollar los modelos y determinar el alcance que estos pueden tener para resolver problemas industriales.

Abstract

The aim of the project that is going to be treated in this Master's Thesis is the study of different metaheuristics in the optimization community, they will be programmed in MATLABTM (The Mathworks Inc., Natick, MA, USA). The models developed provide a useful tool to design suitable industrial equipments and optimize the processes.

Real industrial processes are studied in the area of pulsed electric field for food processing. The aim is to study the optimal geometry of the electric insulator and evaluates the models with respect to electric field uniformity. The physics model is programmed using COMSOL MultiphysicsTM (COMSOL AB, Stockholm, Sweden).

The framework of the study is mathematical optimisation of continuous problems, and the scope is developing models and determining their limits to solve the industrial problems.

Indice de contenidos

1.- Introducción

1.1.- Motivación y objetivos

2.- Modelo físico

3.- Metodología: Optimización matemática

3.1.- Búsqueda Tabú

3.2.- Temple Simulado

4.- Soluciones del modelo y resultados

5.- Conclusiones y futuros trabajos

6.- Bibliografía

1. Introducción

En esta primera sección se presentará el problema físico que se desea optimizar. Además se plantearán las motivaciones y los objetivos de este trabajo de fin de máster.

En la siguiente sección se explicará con detalle el modelo físico completo que será implementado en el software COMSOL Multiphysics TM.

En la tercera sección se describirá la metodología usada, que es la optimización matemática. En particular los algoritmos de la Búsqueda Tabú [2] y del Temple Simulado [9], que pertenecen al grupo de los algoritmos metaheurísticos.

En la sección cuarta se explicarán la implementación y aplicación de estos dos algoritmos estudiados a nuestro problema físico. También se mostrarán y valorarán los resultados obtenidos y a partir de estos se concluirá con las conclusiones y las líneas futuras obtenidas durante el desarrollo total de este trabajo.

El problema físico que se estudiará en este trabajo es un caso en el contexto del tratamiento de alimentos, que pertenece a la rama de la ingeniería alimentaria. En particular se estudiará el tratamiento mediante técnicas de campos eléctricos.

La ingeniería alimentaria o ingeniería de alimentos es la rama de la ingeniería que tiene como función el estudio de la transformación de materias primas de consumo humano en productos con una vida útil más prolongada fundamentada en la comprensión de fenómenos de la química de los alimentos, la biología y la física.

Algunas de las tecnologías de preservación de alimentos se basan en la inactivación o prevención del crecimiento de microbios y/o enzimas. En muchos casos, los alimentos son preservados utilizando factores como la temperatura, la actividad del agua, adicción de conservantes o pH. Y en algunos de estos casos, no se consigue inactivar los microorganismos deseados.

La técnica de campos eléctricos (Pulsed Electric Field, PEF processing), es un método innovador no termal de preservación de alimentos que utiliza cortos impulsos eléctricos para la inactivación de microorganismos. La tecnología PEF implica la aplicación de impulsos eléctricos a alimentos líquidos o semilíquidos localizados entre dos electrodos. Esta técnica puede ser utilizada para la pasteurización no termal de alimentos a temperaturas bajas o moderadas. El mecanismo de inactivación microbiana se basa en la descarga de impulsos eléctricos de alto voltaje, más altos de $70 \text{ kV}\cdot\text{cm}^{-1}$ en unos microsegundos dentro del líquido o dentro de la zona de tratamiento.

A diferencia de la pasteurización termal, donde el tiempo en el que el calor se aplica al alimento es un factor determinante, el tratamiento PEF es instantáneo. Por eso han sido investigadas nuevas tecnologías para sustituir las actuales que utilizan los convencionales métodos basados en calor.

1.1 Motivación y objetivos

La principal motivación de este trabajo, es la utilización de herramientas matemáticas en otras disciplinas como en la ingeniería alimentaria. A pesar de esto no se descuidará la parte física, que será la necesaria para poder modelizar de forma adecuada el problema que se quiere estudiar.

En particular, se quiere utilizar optimización matemática para mejorar las decisiones que afectan a procesos industriales reales como puede ser el tratamiento de alimentos. Pero no se desea sólo centrarse en esta rama de la ingeniería.

Objetivos

Sobre la parte matemática, se realizará un estudio amplio sobre dos algoritmos Búsqueda Tabú y Temple Simulado, en espacios continuos. Se quiere aplicar estos algoritmos de optimización a procesos industriales reales, en particular en la rama de la ingeniería alimentaria. Se implementarán una versión de cada uno de ellos en MATLABTM y se comprobará que funcionan adecuadamente buscando mínimos globales de algunas funciones test como Dixon & Price, Easom, Rosenbrock. . . .

El principal objetivo de este trabajo es el aprendizaje de estos algoritmos y el estudio de la posible aplicación a casos reales, consiguiendo una aplicación muy práctica de herramientas matemáticas.

En cuanto al modelo físico se implementará en COMSOL MultiphysicsTM que es un software para simular y resolver problemas de elementos finitos. Se conectarán ambos programas para que la salida deseada sea un modelo de COMSOL. Para ello se requerirá el estudio y uso de este programa informático para simular y resolver no sólo el caso práctico abordado, sino ser capaz de manejar este programa de forma hábil.

2. Modelo físico

La eficiencia del tratamiento PEF depende de la adecuada elección de ciertas variables del proceso, algunas de estas variables pueden ser la temperatura, el tiempo, la fuerza del campo eléctrico o la energía de entrada. Pero también las variables propias de los materiales a procesar, como la conductividad eléctrica del producto.

La fuerza de los campos eléctrico E es comunmente estudiada como la variable principal del proceso industrial y puede ser estimada mediante la siguiente ecuación:

$$(1) \quad E = \frac{V}{h}.$$

Donde V (kV) es el voltaje aplicado y h (cm) es la distancia entre los electrodos.

La energía disipada en un líquido Q_{espec} (J) puede ser estimada por la fuerza del campo eléctrico y por su índice de flujo:

$$(2) \quad Q_{spec} = \dot{m}^{-1} \cdot f \cdot \int_0^{\infty} \sigma(T) \cdot (E(t))^2 dt,$$

donde $\sigma(T)$ (J/s·K·m) es la conductividad media dependiente de la temperatura T (K), f es la frecuencia del pulso y \dot{m} es el índice de flujo másico (kg/s).

La fuerza eléctrica requerida para la pasteurización de un líquido está en un rango de $20 \text{ kV}\cdot\text{cm}^{-1}$ a $50 \text{ kV}\cdot\text{cm}^{-1}$. Para estimar el incremento en la temperatura producido por el pulso eléctrico se utiliza la siguiente ecuación:

$$(3) \quad \Delta T = \frac{Q_{spec}}{\rho(T) \cdot C_p(T)},$$

donde $C_p(T)$ (J/kg·K) representa el calor específico dependiente de la temperatura y $\rho(T)$ (kg/m³) es la densidad del producto tratado.

Como muchas de las variables en los procesos PEF son prácticamente imposible de determinar experimentalmente, se utilizarán técnicas de modelización multifísica para simular y predecir la distribución del campo eléctrico y la distribución de la temperatura entre otras variables.

La eficiencia del tratamiento de pulsos eléctricos en la inactivación de los microbios depende altamente de la uniformidad del campo eléctrico en la zona de tratamiento de la cámara de PEF. La no uniformidad de los impulsos eléctricos es el principal problema de este tratamiento que puede generar 2 problemas significativos.

El primero porque la aparición de picos en la distribución del campo eléctrico puede causar grandes incrementos en la temperatura y que en ciertas áreas haya un sobreprocesamiento, y una insuficiente inactivación en ciertas regiones donde el umbral no ha sido alcanzado.

El segundo es que la no uniformidad puede provocar graves problemas de erosión en el aislante del interior de la cámara y en los electrodos debido a descargas parciales. Existen dos formas de reducir el riesgo de estos picos, uno es reducir el voltaje aplicado en la zona de tratamiento, y la otra forma es cambiar el diseño y geometría de la cámara completa.

Algunos estudios han demostrado que para reducir los riesgos se puede modificar el diseño del aislante, lo que permite cambios en la distribución del campo eléctrico y de las características del flujo. Este será el estudio en el que nos basaremos. Se buscará optimizar el diseño del aislante para poder evitar al máximo los dos principales problemas comentados, ya que mejorará la uniformidad del tratamiento mediante pulsos eléctricos y reducirá la erosión en el material de la cámara y de los materiales dieléctricos.

El modelo multifísico PEF que utilizaremos se basa en un Sistema de Impulsos Eléctricos 25kW (Diversified Technologies, Inc., Bedford, MA, USA). La geometría de la cámara de tratamiento se basa en el modelo de Buckow et al. (2010) [10]. Consiste en 2 electrodos anclados en los extremos de la cámara, uno arriba y otro abajo. Otro electrodo de alto voltaje se encuentra en medio de estos dos electrodos. Las dimensiones y geometría de los electrodos y del aislante de la cámara de tratamiento PEF (Buckow et al. 2010) se muestra en la figura 1.

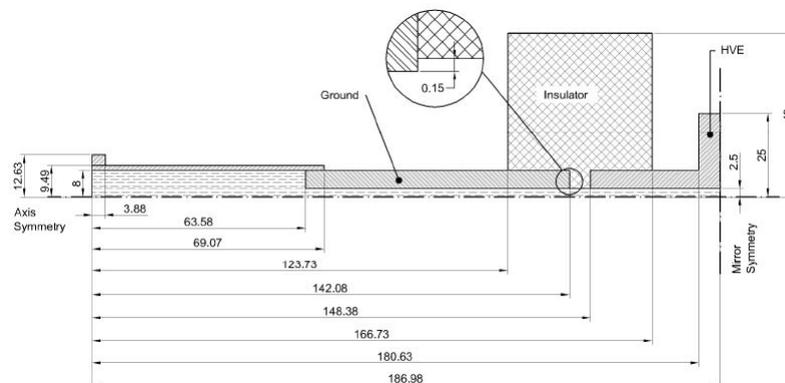


FIGURE 1. Dimensiones y geometrías de los electrodos y del aislante de la cámara de tratamiento PEF

El modelo de fluidos dinámicos fue diseñado en 3D, pero se comprobó que resolver ese modelo con un sistema con variables como la masa, momento, energía, descarga eléctrica y las condiciones de contorno, era muy costoso de resolver computacionalmente. Como la cámara es simétrica frente a uno de los ejes, se hizo una aproximación en 2D en la que los tiempos computacionales decrecen considerablemente.

Como se ha comentado previamente, la forma y las dimensiones geométricas del aislante de la cámara de tratamiento tiene un importante impacto sobre la distribución del campo eléctrico dentro de la cámara. Aparte de su uniformidad, también afecta a la magnitud de la fuerza del campo eléctrico por lo que el aumento de la temperatura también estará afectado. Se decide que la forma y sus dimensiones sean variables en el algoritmo de optimización. En el modelo de optimización de Kai Knoerzer et al. [1] se consideran 4 formas diferentes y se compara la distribución del campo eléctrico en ellas, estos resultados en la figura 2,

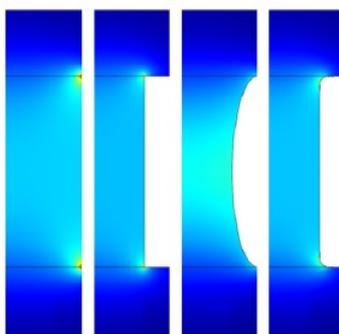


FIGURE 2. Comparación de la distribución del campo eléctrico en las 4 geometrías diferentes de aislante

Las variables que se utilizarán en los algoritmos diseñados en este trabajo serán todas las variables geométricas de medida que definen este tipo de geometrías, por lo que la solución que obtengamos mediante los algoritmos podrá tener una forma diferente de las 4 anteriores (figura 2), haciendo que la solución pueda ser mucho más variada.

El escenario multifísico que queremos modelizar está gobernado por transferencia de calor, dinámica de fluidos y campos eléctricos, el modelo que define el sistema PEF requiere resolver un sistema de ecuaciones en derivadas parciales que describen la conservación de la masa, momento, energía y carga eléctrica. A continuación describiremos algunas de las ecuaciones necesarias para el modelo.

Ecuación del potencial eléctrico:

$$(4) \quad \nabla \cdot (\sigma(T) \cdot \nabla \cdot V - J) = 0,$$

donde J (A/m²) es la densidad de la corriente eléctrica.

La relación del campo eléctrico y el potencial es la siguiente

$$(5) \quad E = -\nabla V.$$

La ecuación del campo de flujo está gobernada por la ecuación de continuidad (equilibrio de masa):

$$(6) \quad \frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{v}) = 0,$$

donde \vec{v} es el vector de velocidad.

Debido a las altas velocidades en la cámara, el fluido se dice turbulento (desordenado y no estacionario) si $Re > 2300$, siendo Re el número de Reynolds

$$(7) \quad Re = \frac{\rho \cdot v \cdot d}{\eta},$$

donde ρ es la densidad del fluido, v (m/s) velocidad característica del fluido, d (m) es la longitud del sistema y η (Pa·s o Kg/m·s) es la viscosidad dinámica

La viscosidad turbulenta η_T está dada por,

$$(8) \quad \eta_T = \rho C_\mu \frac{k^2}{\varepsilon},$$

donde $C_\mu = 0.09$, k es la energía cinética turbulenta y ε es el índice de turbulencia disipada.

La ecuación del momento es la siguiente:

$$(9) \quad \rho \left[\frac{\partial \vec{v}}{\partial t} + (\vec{v} \cdot \nabla) \vec{v} \right] = -\nabla P + \nabla \cdot ((\eta + \eta_T) \cdot \nabla \vec{v}) + \rho g,$$

donde \vec{v} es la velocidad media, P la presión (Pa), η representa la viscosidad dinámica y g es la constante de gravedad ($g=6.67 \cdot 10^{-11} \text{N} \cdot (\text{m}/\text{kg})^2$).

Para cada fenómeno físico, es necesario definir una serie de condiciones de contorno. La primera condición que hay que asumir es que el aislante es efectivamente un aislante eléctrico:

$$(10) \quad n \cdot J = 0.$$

Las condiciones en el electrodo de voltaje, que como hemos explicado antes se encuentra en medio de la cámara, donde

$$(11) \quad V = V_{max},$$

siendo V_{max} el máximo voltaje permitido en un impulso eléctrico, todo esto en los electrodos anclados donde el potencial en las fronteras debe de ser 0.

Para los modelos simétricos en 2D, es necesario asumir las condiciones de simetría en los ejes, que se pueden simplificar en las siguientes expresiones:

$$n \cdot (-k \nabla T) = 0 \text{ y } n \cdot \vec{v} = 0,$$

donde n es la dirección normal.

Las ecuaciones en derivadas parciales descritas anteriormente que definen este modelo físico, es decir las ecuaciones de conservación de la masa, del momento, de energía y de carga eléctrica, se modelizan con el programa COMSOL MultiphysicsTM. Y se resuelven discretizando el problema mediante el método de los elementos finitos. Por la gran cantidad de parámetros dependientes e independientes que hay que tener en cuenta para la mejora de la técnica de PEF, se creará un parámetro que resume la máxima información necesaria, esto se desarrollará en la sección 4 *Soluciones del modelo y resultados*.

3. Metodología: optimización matemática

A continuación se describirá la metodología utilizada, centrandonos en dos algoritmos heurísticos que son la búsqueda tabú y el temple simulado.

La investigación operativa es una rama de las matemáticas que hace uso de modelos matemáticos y algoritmos con el objetivo de ser usados como apoyos a la toma de decisiones.

La optimización o programación matemática se puede definir como la ciencia que determina la mejor solución a nuestro problema. Para definir nuestro problema utilizaremos modelos, un modelo es una herramienta que nos sirve para lograr una visión bien estructurada de la realidad. Por ello la generalización de la teoría de optimización comprende un área muy importante en las matemáticas aplicadas.

Los problemas de optimización son comunes en muchas disciplinas y diferentes dominios, se debe encontrar una solución óptima o cerca de la óptima.

Un modelo matemático comprende principalmente 3 conjuntos básicos de elementos:

- **Variables de decisión:** son las incógnitas o decisiones que deben determinarse resolviendo el modelo.
- **Restricciones:** representan el conjunto de relaciones bien definidas (expresadas mediante ecuaciones e inecuaciones) que ciertas variables están obligadas a satisfacer, para tener en cuenta las limitaciones tecnológicas, económicas y otras del sistema, el modelo debe de cumplirlas.
- **Función objetivo:** mide la medida de efectividad del sistema como una función matemática de las variables de decisión. La solución óptima será aquella que produzca el mejor valor de la función objetivo.

Existen diferentes tipos de técnicas de optimización para diferentes modelos, principalmente se separarían en dos tipo distintos de métodos:

- Métodos exactos: que recorren todas las soluciones (enumeración explícita) o parte de ellas (enumeración implícita), buscan y garantizan un óptimo local:
 - **Programación lineal**, consiste en optimizar una función lineal (función objetivo), de tal forma que las variables de dicha función están sujetas a una serie de restricciones expresado en un sistema de inecuaciones lineales.

$$\begin{aligned} \min_x c^T x \\ Ax = b \\ x \geq 0 \\ x \in \mathbb{R}^n, c \in \mathbb{R}^n, A \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m \end{aligned}$$

- **Programación entera mixta**, la función objetivo es una función lineales, pero a diferencia de la programación lineal, esta puede tener decisiones de tipo discreto que están representadas por variables enteras o binarias.

$$\begin{aligned} \min_x c^T x + d^T y \\ Ax + By = b \\ x, y \geq 0 \\ x \in \mathbb{Z}^n, y \in \mathbb{R}^l, c \in \mathbb{R}^n, d \in \mathbb{R}^l \\ A \in \mathbb{R}^{m \times n}, B \in \mathbb{R}^{m \times l}, b \in \mathbb{R}^m \end{aligned}$$

- **Programación no lineal** en este caso tanto la función objetivo como las restricciones son no lineales, el campo de aplicación de la programación no lineal es muy amplio.

$$\begin{aligned} \min_x f(x) \\ g(x) = 0 \\ h(x) \geq 0 \\ l \leq x \leq u \\ f : \mathbb{R}^n \rightarrow \mathbb{R} \\ g, h : \mathbb{R}^n \rightarrow \mathbb{R}^m \end{aligned}$$

- Métodos heurísticos: son algoritmos aproximados que buscan una buena solución aunque no sea la óptima, en un tiempo reducido. Estos métodos no pueden asegurar la optimalidad al no recorrer explícitamente o implícitamente todas las soluciones.

En concreto en este estudio, nos centraremos en el modelado mediante algoritmos heurísticos.

Las heurísticas son métodos de optimización que tratan de explotar el conocimiento específico del problema. Son normalmente diseñadas para un problema particular, estos métodos son especialmente populares en investigación operativa y en inteligencia artificial.

Algunas heurísticas más conocidas son:

- **Algoritmos iterativos**: tratan a partir de una solución factible, construir otra mejor e iterar el proceso de mejora hasta verificar cierto criterio de parada. Sea $x^0 \in X$ la solución factible inicial. A partir de $x^k \in X$, se construye otra $x^{k+1} = h(x^k) \in X$ verificando $f(x^{k+1}) \leq f(x^k)$ donde f es la función objetivo del problema de minimización. La elección de la función h y el valor inicial x^0 caracterizan el método iterativo.

- **Algoritmos voraces:** los métodos voraces (greedy) van construyendo paso a paso la solución de forma que hasta el final no se tiene ninguna solución factible, no siempre se puede garantizar la optimalidad de la solución obtenida siguiendo este método.
- **Algoritmos metaheurísticos:** son procedimientos iterativos que guían una heurística subordinada combinando de forma inteligente distintos conceptos para explorar y explotar adecuadamente el espacio de búsqueda. Cuando se diseña una metaheurística hay un conjunto de parámetros que, al variar su valores, determinan las características de las heurísticas que están utilizando. Se puede entender como un procedimiento parametrizado. Algunos de estos algoritmos son: Temple simulado, Algoritmos genéticos, Búsqueda Tabú, Redes neuronales y Relajación Lagrangiana. El término meta-heurística fue acuñado en 1986 por Glover [4 y 5], al introducir la búsqueda tabú.

En este trabajo nos centraremos en dos de los algoritmos heurísticos más conocidos la Búsqueda Tabú (Tabu Search) y el Temple Simulado (Simulated Annealing). Ambos algoritmos surgieron para la búsqueda de soluciones globales de problemas de optimización combinatoria.

La optimización combinatoria estudia el modelado y solución algorítmica de problemas donde se busca minimizar (o maximizar) una función de múltiples variables definidas sobre un conjunto discreto. Como ejemplos podemos mencionar, entre otros, el ruteo y carga de vehículos en redes de distribución, el diseño de redes de telecomunicación, la planificación de la producción, la selección de carteras financieras, la planificación de la generación de electricidad y la distribución de ambulancias en una región para asegurar un cierto nivel de servicio a su población.

En la optimización combinatoria confluyen la matemática discreta, la teoría de conjuntos y la programación matemática. La solución de algunos de estos problemas de optimización combinatoria necesita una enumeración implícita de todas las soluciones, lo que provoca que los algoritmos no puedan resolver los problemas de tamaño medio o grande, de ahí proviene la utilización de algoritmos metaheurísticos.

Como la mayoría de los algoritmos metaheurísticos han sido diseñado para resolver problemas combinatorios discretos hay que hacer una adaptación de estos algoritmos al espacio de soluciones continuo. Como veremos más particularmente en cada algoritmo, el cambio principal se encuentra en la definición de vecino o de siguiente movimiento.

3.1 Búsqueda Tabú

La Búsqueda Tabú (TS) es una metaheurística originalmente diseñada para resolver problemas de optimización combinatoria como hemos señalado previamente.

Básicamente trata de diversificar la búsqueda y de escapar de óptimos locales, este algoritmo utiliza una lista con soluciones previas. Una estrategia Búsqueda Tabú simple combina memoria de corta (lista tabú) con mecanismos de búsqueda local. Nuevas soluciones que son creadas por la búsqueda local son añadidas a la lista y las soluciones más antiguas son eliminadas en pasos sucesivos. La búsqueda local sólo puede crear nuevas soluciones que no existen en la lista tabú T . Para impedir problemas de memoria, normalmente la longitud de esta lista es finita. Se suele usar un algoritmo voraz que evalúe las vecindades de una solución y que continúe con una solución x' que no esté contenida en T .

El propósito de la lista tabú es permitir escapar de una solución local a una solución óptima, prohibiendo soluciones ya visitadas y explorando nuevas áreas no registradas.

Se implementará una adaptación de la Búsqueda Tabú básica a problemas continuos llamada Búsqueda Tabú Directa [2]. En este método, se utilizará búsquedas basadas en estrategias directas para dirigir la búsqueda tabú. Estas estrategias están basadas en el método Nelder-Mead [11], también se definirá un nuevo concepto de lista tabú con reglas antibucles con la utilización de dos nuevos conceptos llamados regiones tabú y semiregiones tabú. Se usará diversificación e intensificación de búsqueda para mejorar las soluciones encontradas.

A partir de la primera presentación de la Búsqueda Tabú por Glover [4 y 5], numerosos estudios han aparecido sobre su adaptación al continuo de este algoritmo, junto al presentado en este trabajo destacamos también la Búsqueda Tabú Continua Intensificada (Enhanced Continuous Tabu Search, ECTS) [3], lo primero que hace es una búsqueda diversificada para localizar las áreas más prometedoras, después en esas áreas intensifica la búsqueda.

El algoritmo directo a diferencia del anterior tiene tres fases: exploración, diversificación e intensificación. Estas fases se comentarán más adelante.

Este algoritmo puede clasificarse como un método multi-arranque. Los métodos multi-arranque establecen una contrucción de procedimientos de exploración global y búsquedas locales. Estos métodos obtienen buenos resultados en la optimización global no-lineal y problemas combinatorios.

A continuación se explicará con detalle algunas nociones del algoritmo. Se empezará por los elementos de memoria, las estrategias de búsqueda y la definición de vecino. Después se explicará el método completo de la Búsqueda Tabú Directa.

Elementos de memoria

El concepto de memoria tiene una labor muy importante en la Búsqueda Tabú. Usando de forma eficiente este concepto en los esquemas de intensificación y diversificación se consigue una técnica de búsqueda inteligente. Utilizaremos 4 elementos de memoria diferentes que son los siguientes:

- **Lista Tabú (TL):** En esta lista se almacenan algunas soluciones visitadas, $TL = \{t_i\}_{i=1}^L$, los elementos están en orden descendente en función del rango que depende de la función objetivo, por lo que es una lista de las mejores soluciones visitadas. Esta lista se usa principalmente para la última fase de intensificación.
- **Región Tabú (TR):** Son definidas como esferas de radio r_{TR} y sus centros son los puntos de la TL, donde $r_{TR} > 0$.
- **Semi Región Tabú (Semi-TR):** Para cada TR, se define Semi_TR como la región envolvente de esta TR, con un radio r_{STR} desde su centro, donde $r_{STR} > r_{TR}$. Si un movimiento cae en alguna Semi_TR, entonces se aplicará un procedimiento que evite volver a la vecindad de una solución ya visitada.
- **Lista de Región Visitada (VRL):** Para guardar alguna información histórica necesitamos una lista de las regiones visitadas e información sobre esta región. Esta lista por cada elemento almacena, el centro ξ_i de la región visitada, el radio ρ_i que define una esfera de la región, y la frecuencia φ_i a la que se visita esta región, por lo que $VRL = \{(\xi_i, \rho_i, \varphi_i)\}_{i=1}^M$, donde M es el número total de regiones visitadas que se encuentran en la lista. Para la diversificación se utiliza la información de esta lista, se trata de generar soluciones que no se encuentren cerca de las regiones visitadas con más frecuencia. Con este fin se define la función $\Phi(\varphi)$ para poder distinguir regiones más frecuentes de las menos frecuentes, se define de la siguiente manera

$$\Phi(\varphi) = \gamma(1 - e^{-\gamma(\varphi-1)}),$$

donde $\gamma \in (0, 1]$ es una valor constante. Notese que esta función es estrictamente creciente y acotada por γ .

Estrategias de búsqueda y procedimientos

Procedimiento 3.1

El procedimiento 3.1 se aplica para crear movimientos que eviten caer en vecindades de soluciones ya visitadas.

- (1) Sea el punto x que cae en las Semi-TRs v con centros $t_1 \dots t_v$. Se calcula el centroide \bar{t} de las Semi-TRs y la distancia máxima d_{max} entre x y todos los centros,

$$\bar{t} = \frac{1}{v} \sum_{i=1}^v t_i$$

$$d_{max} = \max_{i=1, \dots, v} \{ \|x - t_i\| \}.$$

- (2) Contruir la dirección de búsqueda, esta será paralela a los ejes de coordenadas pero en la dirección de $x - \bar{t}$, la dirección de búsqueda son determinadas por $sign((x)_i - (\bar{t})_i)e_i$, $i = 1, \dots, n$, donde $e_i \in \mathbb{R}^n$ es un vector unitario. Los puntos vecinos son generados a lo largo de estas direcciones con un tamaño del paso de $\beta > 0$, donde $\beta > d_{max} + r_{TR}$ para asegurar que no cae dentro de la región tabú.

La figura 3 ilustra este procedimiento, donde la solución x cae en las semi-regiones tabú en dos dimensiones. En este caso las direcciones de la búsqueda d_1 y d_2 son construidos siguiendo el vector $(x - \bar{t})$, donde como hemos explicado \bar{t} es el centroide y tomando β de la siguiente manera

$$\beta > r_{TR} + \max\{\|x - t_1\|, \|x - t_2\|\}.$$

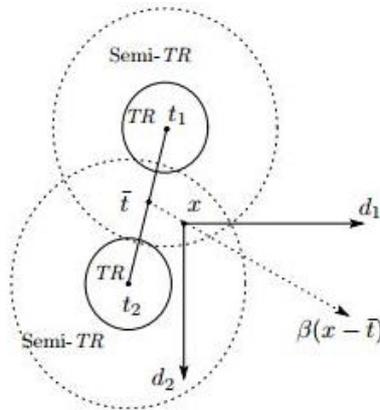


FIGURE 3. Búsqueda de vecino fuera de las semi-regiones tabú

Procedimiento 3.2

El procedimiento 3.2 se utilizará cuando la solución no caiga en las regiones tabú sino en las regiones visitadas, el método es el siguiente

- (1) Se genera de el punto x de forma aleatoria dentro del dominio de la función objetivo a minimizar (o maximizar) f .
- (2) Se calcula los siguientes valores

$$d_i = \frac{\|x - \xi_i\|}{(1 + \Phi(\varphi_i))}, \quad i = 1, \dots, M, \quad \Phi(\varphi) = \gamma(1 - e^{-\gamma(\varphi-1)}).$$

Si $\min_{1 \leq i \leq M} (d_i / \rho_i) \geq 1$, entonces se acepta x , sino se vuelve al paso 1.

El punto x es aceptado por el procedimiento 3.2 si satisface la siguiente ecuación $\frac{\|x - \xi_i\|}{\rho_i} \geq 1 + \Phi(\varphi_i) \forall i = 1, \dots, M$. Esto significa que ningún punto puede ser aceptado si pertenece a una región visitada previamente, es más, un punto que esté cerca de una región visitada frecuentemente será difícil que sea aceptado, esto se mide en función de γ ya que cuanto más cerca esté de 1, menos posibilidad habrá de aceptar valores cercanos a regiones visitadas.

Estrategia de búsqueda de vecinos locales

Para explorar la región alrededor de una solución y generar el siguiente movimiento, se definen vecindades y estrategias de búsqueda local con métodos directos. Para el algoritmo de Búsqueda Tabú Directo, existen dos estrategias de búsqueda que dan buenos resultados: estrategia de búsqueda Nelder-Mead (NMS) y la estrategia de búsqueda del Patron Adaptativo (APS) (ver [11] y [6]).

En este trabajo se utilizará la búsqueda basada en el método Nelder-Mead. Este método es un método de búsqueda directa para la minimización sin restricciones de funciones multidimensionales. Este algoritmo se ha convertido en uno de los métodos más ampliamente usados para la optimización no lineal.

El método Nelder-Mead trata de minimizar una función escalar no lineal de n variables usando sólo valores de la función, sin obtener ninguna información de la derivada (ni implícita ni explícitamente), esto es una gran ventaja cuando desconocemos analíticamente la función objetivo o es muy complejo el cálculo de su derivada para poder utilizar un algoritmo en el descenso en la dirección del gradiente. Otra ventaja a destacar es que tiende a necesitar menos evaluaciones de la función que otras alternativas, esto puede mejorar el tiempo de ejecución ya que la evaluación puede ser cara y consuma mucho tiempo.

Se generan $p(=n)$ puntos vecinos $\{y_i\}_{i=1}^n$, y $q(=1 \text{ o } 0)$ punto local. Los puntos vecinos son generados a lo largo de las direcciones paralelas a los ejes empezando por el punto x en el que nos encontramos, si este valor x cae en las semi-regiones tabú se aplica el procedimiento 3.1 para poder salir de estas áreas sin caer en otra región tabú. Sino se construyen las direcciones de forma aleatoria. Para generar el punto local, se construye un simplex S que consiste en la solución actual x y el conjunto de n puntos vecinos $\{y_i\}_{i=1}^n$, es decir $S = \{x, y_1, \dots, y_n\}$.

Algunas iteraciones del método de Nelder-Mead se comienzan en este conjunto S . Si se consigue un punto que mejora la solución desde este método, entonces llamaremos a ese punto mejora y_{n+1} haciendo $q = 1$, en caso contrario $q = 0$.

Para explicar de una forma más correcta las estrategias del Nelder-Mead, mostraremos a continuación en la figura 5 las estrategias en un espacio de 2 dimensiones.

Dada una solución x y dos vecindades y_1 e y_2 generadas como vecinos de x . Para encontrar un punto local construimos el conjunto $S = \{x, y_1, y_2\}$ como en la figura 4.

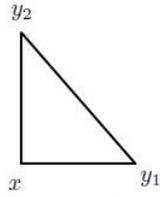


FIGURE 4. Conjunto $S = \{x, y_1, y_2\}$

Asumiendo que el peor punto del conjunto S es x , se aplica las operaciones del método Nelder-Mead descritas en la figura 5 para encontrar un mejor movimiento, si existe este valor será guardado como un punto local.

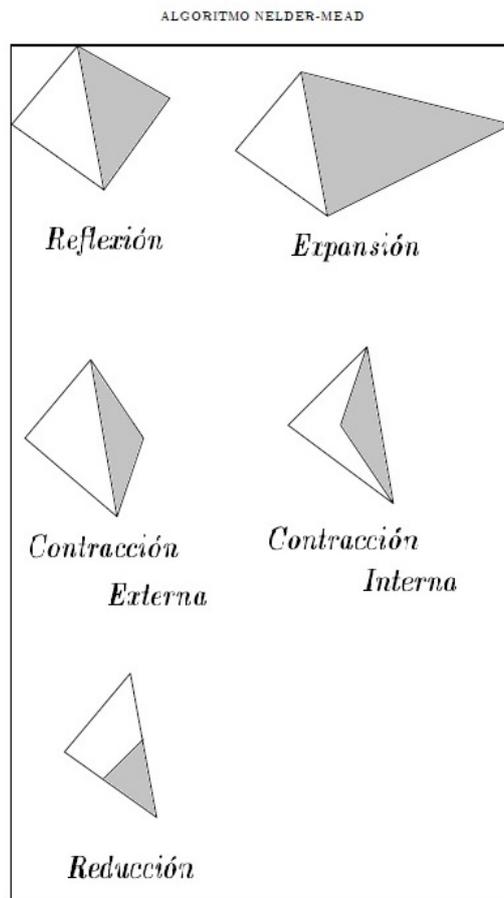


FIGURE 5. Estrategias del NM en 2D

En el caso de 3 dimensiones que aún es asumible una visualización, con un conjunto inicial $S = \{x, y_1, y_2, y_3\}$, la reflexión, contracción y expansión están representadas en la figura 6

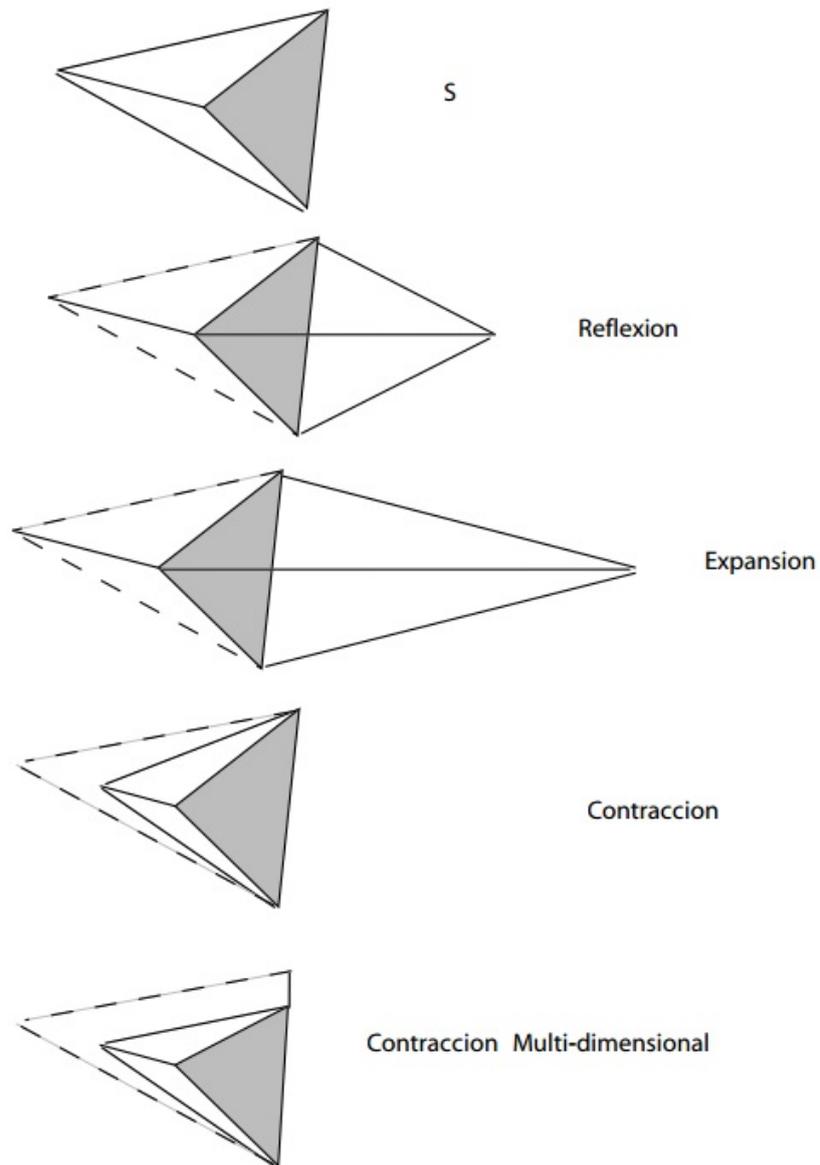


FIGURE 6. Estrategias del NM en 3D

Existen otras estrategias de búsqueda de vecinos con buenos resultados en su implementación en la búsqueda tabú en las que no entraremos en profundidad, por ejemplo la estrategia de Búsqueda de Patrones Adaptativa (APS), es un método basado en la aproximación en la dirección descendente (ADD), en este método utiliza diferentes puntos cercanos a $x \in \mathbb{R}^n$ para generar una aproximación en la dirección descendente de la función f en x .

Búsqueda Tabú Directa (DTS)

En este apartado, describiremos con detalle como es el método de la búsqueda tabú modificado con los elementos de memoria y con el método Nelder-Mead.

El algoritmo DTS consta de 3 fases: exploración, diversificación e intensificación. La estructura del algoritmo de Búsqueda Tabú Directa se muestra en el diagrama de la figura 7, y a continuación se describe cada una de sus fases con más detalle.

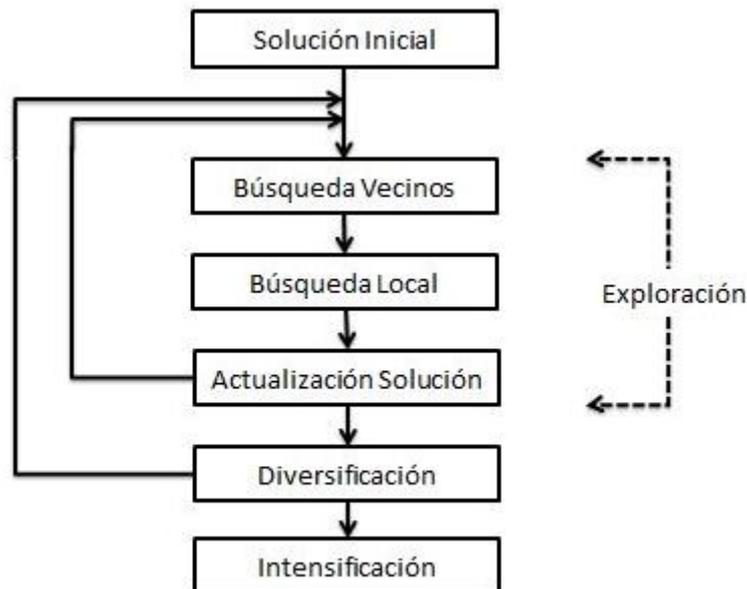


FIGURE 7. Estructura principal del método DTS

Exploración

Empezando con la solución inicial $x_0^{(0)}$, en un bucle principal con cada iteración se hace una estrategia de búsqueda local (Nelder-Mead) para generar n puntos vecinos $\{y_i\}_{i=1}^n$ en cada valor de la iteración en la que se encuentre $x_k^{(j)}$. Si se encuentra un movimiento que mejore la solución a lo largo de los puntos generados, actualizamos la solución y se incorpora al siguiente bucle. En el caso en el que no se mejore la solución $x_k^{(j)}$, es decir, es mejor solución que todas las vecindades generadas, entonces se generan q puntos vecinos $\{y_{n+i}\}_{i=1}^q$ donde $q = 0$ o 1 según el método Nelder-Mead.

Después, la solución $x_k^{(j)}$ se actualiza como el mejor valor obtenido en la vecindad y de los puntos locales, $x_{k+1}^{(j)} := \operatorname{argmin}_{i=1, \dots, n+q} \{f(y_i)\}$.

La lista tabú (TL) también se actualiza con el valor $x_k^{(j)}$ reemplazando al peor valor de la lista si este fuera peor que $x_k^{(j)}$. Si una nueva región es alcanzada, entonces se actualiza la lista de región visitada (VRL) añadiendo la información de esta región. Los bucles de esta fase estarán limitados por un valor máximo.

Diversificación

La fase de diversificación se aplica para buscar un nuevo valor inicial $x_0^{(j+1)}$ con el que comenzar otra vez la fase de exploración. Esta fase se lleva a cabo cuando se ha terminado el límite del bucle de la exploración o cuando no se consigue un movimiento de mejora.

Con la lista de región visitada (VRL), se aplica el procedimiento 3.2 para generar el punto inicial $x_0^{(j+1)}$ en alguna nueva región. Entonces es cuando se inicializa la fase de exploración empezando en este nuevo valor.

Intensificación

Esta fase comienza al terminar completamente las fases de exploración y diversificación. Una vez obtenida la lista tabú completa, que tiene las soluciones ordenadas por preferencia, se toman las mejores soluciones y se aplica la modificación de Kelley [7] del método Nelder-Mead a esos valores, haciendo una búsqueda local en las vecindades de esos valores.

A continuación se detallará una descripción específica y formal del algoritmo completo (DTS_{NMS}).

Algoritmo $DTS_{NMS}(f, x_0^{(0)})$

Etapa 1. Inicialización

Se eligen l_{main} , l'_{main} , l_{inner} y $l'_{inner} \in \mathbb{Z}^+$. Se toma $x_0^{(0)}$ como valor inicial, y se crean los conjuntos lista tabú (TL) y la lista de regiones visitadas (VRL) vacías.

Etapa 2. Búsqueda de Exploración-Diversificación (Bucle principal).

Sea $j := 0$ y se repite el bucle hasta l'_{main} haciendo iteraciones hasta que falle la mejora o que j exceda l_{main} .

Etapa 2.1. Búsqueda de Exploración (NMS) (Bucle interior).

Sea $k := 0$ y se repite el bucle hasta l'_{inner} haciendo iteraciones hasta que falle la mejora o que k exceda l_{inner} .

Etapa 2.1.1. Búsqueda de direcciones. Si la solución obtenida cae en las semi-TR, se usa el procedimiento 3.1 para construir las direcciones $d_i := (-1)^{\tau_i} e_i$, $i = 1, \dots, n$, donde $e_i \in \mathbb{R}^n$ es un vector unitario y τ_i es un número aleatorio entero, y se elige el tamaño del paso $\{\Delta_i\}_{i=1}^n$.

Etapa 2.1.2. Búsqueda de vecindad. Se generan n puntos vecinos $y_i := x_k^{(j)} + \Delta_i d_i$, $i = 1, \dots, n$. Si se encuentra un movimiento mejor durante el proceso, se para la generación de puntos, se toma el valor $x_{k+1}^{(j)}$ igual al valor encontrado, y se salta hasta el paso 2.1.4.

Etapa 2.1.3. Búsqueda local. Aplicamos n iteraciones del método Nelder-Mead empezando con el conjunto $S = \{x_k^{(j)}, y_1, \dots, y_n\}$. Si se mejora el punto inicial del método desde estas iteraciones, se crea el punto y_{n+1} igual a este punto encontrado, y se toma $q := 1$. Si no se encuentra, se toma $q := 0$ y $x_{k+1}^{(j)} := \operatorname{argmin}_{i=1, \dots, n+q} \{f(y_i)\}$.

Etapa 2.1.4. Actualización de parámetros. Sea $x_k^{(j)}$ reemplaza al peor elemento de la lista tabú y se actualiza esta misma. También se actualiza la lista de regiones visitadas y se toma $k := k + 1$.

Etapa 2.2. Búsqueda de Diversificación.

Se genera un nuevo punto inicial $x_0^{(j+1)}$ utilizando el procedimiento 3.2. Se actualiza la lista tabú y la lista de regiones visitadas, y se toma $j := j + 1$.

Etapa 3. Intensificación

Se aplica la modificación de Kelley [7] del método Nelder-Mead empezando en las mejores soluciones de la lista tabú.

3.2 Temple Simulado

Temple Simulado (o Recocido Simulado SA) es un algoritmo de búsqueda local (metaheurística) capaz de escapar de un óptimo local. Es fácil de implementar, tiene propiedades de convergencia y es usado para generar movimientos que escalan óptimos locales. Es un método probabilístico propuesto por Kirkpatrick, Gelatt y Vecchi (1983) [9]. Se diseñó para encontrar mínimos globales de funciones de coste que poseen multitud de mínimos locales.

Los algoritmos de temple simulado están basados en una analogía de un fenómeno físico.

El método de recocido se utiliza en la industria para obtener materiales más resistentes, o más cristalinos, en general, para mejorar las cualidades de los materiales que se manejan. El proceso consiste en derretir el material (calentando a altas temperaturas). Luego se hace descender la temperatura muy lentamente por etapas, dejando que en cada una de esas etapas los átomos estén en equilibrio (es decir, que los átomos alcancen una configuración óptima para esa temperatura). En esa situación los átomos tienen una distribución aleatoria dentro de la estructura del material y la energía del sistema es máxima. Al final del proceso, los átomos forman una estructura cristalina altamente regular, el material alcanza así una máxima resistencia y la energía del sistema es mínima.

Experimentalmente se comprueba que si la temperatura se hace descender bruscamente o no se espera suficiente tiempo en cada etapa, al final la estructura del material no es la deseada.

La física estadística o mecánica estadística se encarga de desarrollar una serie de métodos para estudiar el comportamiento de grandes cantidades de átomos de un sistema. Debido a que en promedio, en un sistema hay 10^{23} átomos por cm^3 , solamente puede estudiarse el comportamiento más probable del sistema en equilibrio a una temperatura dada. La experimentación mostró que los átomos de un sistema en un proceso de recocido se comportan según el factor de probabilidad de Boltzmann.

En 1953 Metrópolis [12] modeló el proceso de recocido: en cada paso del algoritmo se le da al átomo un desplazamiento aleatorio y se mide el cambio de energía ΔE . Si $\Delta E \leq 0$ se acepta el desplazamiento. Si $\Delta E \geq 0$, se acepta el desplazamiento con probabilidad

$$e^{-\frac{\Delta E}{T \cdot K}},$$

donde T es la temperatura a la que se encuentra el sistema y K es la constante de Boltzmann, $K \approx 1,3806504 \times 10^{-23} \frac{J}{K}$.

Kirkpatrick, Gelatt y Vecchi pensaron que era posible establecer una analogía entre los parámetros que intervienen en la simulación termodinámica de Metrópolis y los que aparecen en los métodos de optimización local, tal y como muestra la siguiente tabla adjunta.

Termodinámica	Optimización
Configuración	Solución factible
Configuración Fundamental	Solución óptima
Energía de la Configuración	Coste de la Solución

TABLE 1. Analogía entre términos termodinámicos y de optimización

Se estableció una relación entre el proceso de los átomos del material que están buscando un equilibrio, y las soluciones visitadas por los procedimientos de búsqueda local. El algoritmo del temple simulado es un método donde todo movimiento de mejora es aceptado y también se permiten movimientos de no mejora de acuerdo con una probabilidad basada en la analogía con el proceso físico de enfriamiento y siendo función de la temperatura del sistema.

La estrategia de este algoritmo es comenzar con una temperatura inicial alta, esto implica una probabilidad alta de aceptar un movimiento de no mejora. En cada iteración se va reduciendo la temperatura y por lo tanto la probabilidad va decreciendo conforme avanza el algoritmo. De este modo, inicialmente se realiza una diversificación de la búsqueda sin controlar demasiado el coste de las soluciones visitadas. A la vez que el tiempo pasa y la temperatura se reduce es más difícil aceptar malos movimientos.

Por esta caracterización, este algoritmo es capaz de salir de óptimos locales al aceptar movimientos de no mejora. Al final del proceso es tan poco probable aceptar que no se producen movimientos que empeoren la solución. La figura 8 muestra la aceptación de soluciones y movimientos del algoritmo del temple simulado.

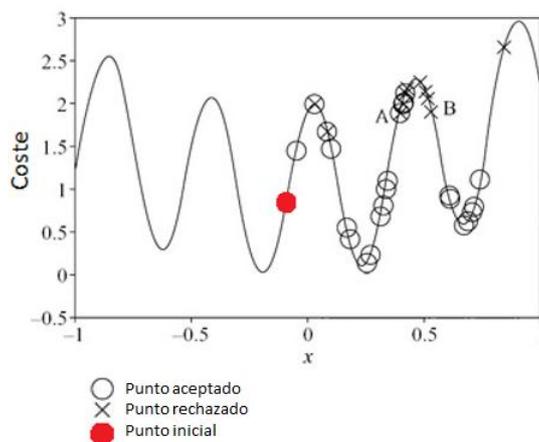


FIGURE 8. Ejemplo de movimientos utilizando un algoritmo del Temple simulado

Igual que el algoritmo de la búsqueda tabú, el algoritmo del temple simulado fue diseñado inicialmente para resolver problemas de optimización combinatoria, para este tipo de problemas este método funciona muy bien proporcionando soluciones rápidas.

Como nuestro problema físico no es combinatorio, hace falta adaptar el algoritmo a los problemas de optimización global continua, es decir los problemas de la siguiente forma

$$f^* = \min_{x \in X} f(x),$$

donde $X \subseteq \mathbb{R}^n$ es un dominio continuo, y se garantiza la existencia de que f tenga un mínimo. El cambio que hay que hacer para adaptarlo al continuo es la forma de generación de movimientos o la definición de vecino. Se decide utilizar el método del descenso en la dirección del gradiente.

Método del descenso en la dirección del gradiente

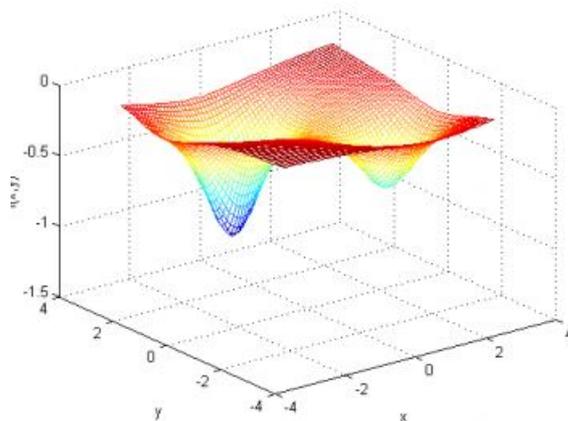
Este método es una de las técnicas más antiguas para minimizar una función definida en un espacio multidimensional. Se trata de un algoritmo iterativo que puede ser visto como un sistema dinámico. Básicamente se basa en que la dirección contraria a la del vector gradiente en un punto es la dirección de más rápido decrecimiento de la función en ese punto. La formulación del algoritmo es la siguiente

$$x_{k+1} = x_k - \eta \nabla f(x_k),$$

donde $\eta \geq 0$, y $\nabla f(x_k)$ es el gradiente en el punto inicial, en nuestro caso debido a la complejidad de la función se estimará el gradiente mediante una aproximación de Taylor centrada.

El algoritmo del descenso en la dirección del gradiente es un método de optimización local, en el momento en el que encuentre una solución local parará. Por ejemplo al minimizar la siguiente función en dos variables:

$$g(x, y) = -\frac{1}{1+2(x-1,5)^2+(y+1)^2} - \frac{6}{1+(x+1)^2+(y-1,5)^2}$$



Dependiendo de donde se inicialice el algoritmo se alcanzará el mínimo global o el mínimo local, vease un ejemplo en la figura 9

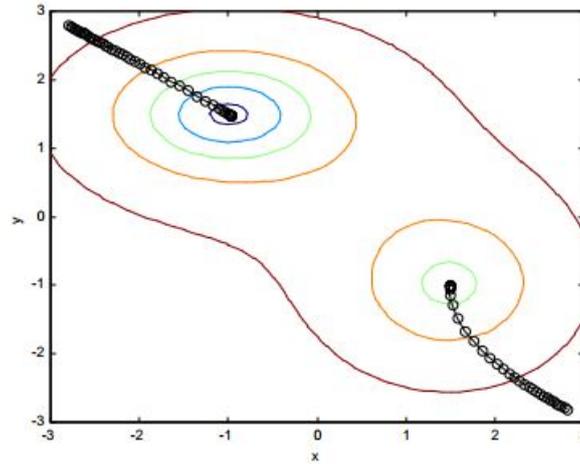


FIGURE 9. Alcance de mínimos locales de la función g

Por lo que nuestro temple simulado necesitará otra definición de vecino aparte de la proporcionada por el algoritmo en el descenso de la dirección del gradiente.

Se generará una dirección aleatoria $d_i \in \mathbb{R}^n$, y se estimará un tamaño de paso Δ_i lo suficientemente pequeño para que no salga del dominio de continuidad de la función f .

Para la correcta implementación del algoritmo hace falta determinar algunos parámetros del sistema.

Función de Aceptación

En la adaptación de los algoritmos del temple simulado a la optimización global, muy poca variedad de funciones de aceptación han sido utilizadas. En la mayoría de los casos se ha utilizado la función A también llamada función Metrópolis

$$A(x, y, t) = \min\left\{1, \exp\left\{-\frac{f(y) - f(x)}{t}\right\}\right\}.$$

La función de aceptación de Metrópolis siempre acepta un punto en descenso, es decir, dado un punto y que mejora o no empeora la función de coste con respecto a x , siempre será aceptada, es decir $f(y) \leq f(x)$. Pero para no estancarse en un mínimo local, es necesario aceptar pasos de ascenso (permitir que el algoritmo escale colinas para salir del mínimo local). Este movimiento de escalada se muestra en la figura 10

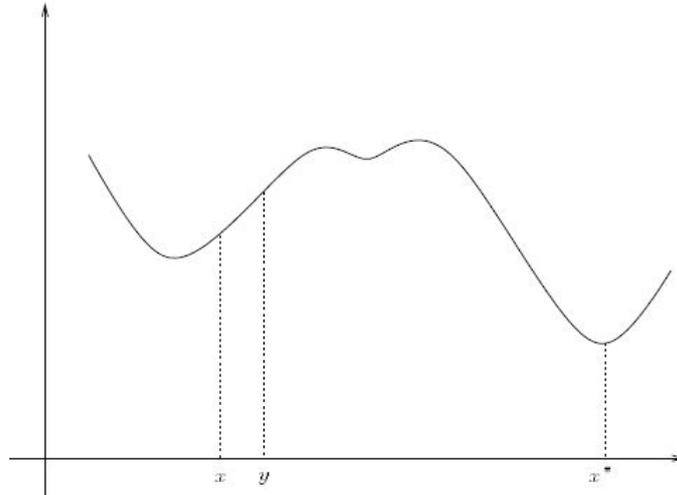


FIGURE 10. Movimiento de escalada

Como podemos ver es necesario un movimiento de $x \rightarrow y$ para poder acceder a la región donde se encuentra el óptimo global x^* . La probabilidad de aceptar está controlada por el parámetro de temperatura t . Como la temperatura decrece a 0, también la probabilidad de aceptar movimientos de ascenso decrecerá a 0.

Otra posibilidad es la función de aceptación llamada criterio de Barker

$$A(x, y, t) = \frac{1}{1 + \exp\left\{-\frac{f(y) - f(x)}{t}\right\}}.$$

Cuando la temperatura decrece, los movimientos de descenso son más probables de ser aceptados, a no ser de que sea muy pequeña la diferencia, mientras que los de ascenso suelen ser rechazados. Esta función tiene un comportamiento muy similar a la función Metropolis.

En nuestro caso se utilizará el siguiente método,

- (1) Se genera un número aleatorio u siguiendo la distribución $U(0, 1)$.
- (2) Se calcula $\delta = f(y) - f(x)$ y $\exp(-\frac{\delta}{t})$.
- (3) Si $\delta < 0$ o $(\delta \geq 0 \wedge u < \exp(-\frac{\delta}{t}))$ entonces $x = y$.

La probabilidad de aceptar incrementos de coste es mayor para incrementos menores que mayores, y a igualdad de incrementos la probabilidad de aceptarlo es mayor cuando mayor sea t .

En el caso en el que no se haya aceptado el valor y se provenga de un vecino por el método del descenso en la dirección del gradiente, se generará el siguiente vecino en una dirección aleatoria. Ya que sino se hiciese este método se generaría el mismo vecino y entrando en bucle, lo único que cambiaría sería el valor simulado u .

Para diseñar un algoritmo del temple simulado hay que determinar algunos parámetros del sistema.

- **La Temperatura Inicial** se suele determinar realizando una serie de pruebas para alcanzar una determinada fracción de movimientos aceptados.
- **Solución Inicial**, en todas las versiones, como el sistema debe ser “derretido” antes de implementar el algoritmo. Esto quiere decir que la solución inicial sea una solución factible del problema, como esto puede ser un problema, se suele implementar un algoritmo voraz (Greedy) de tipo local para buscar la solución inicial para el temple simulado.
- **La Velocidad de Enfriamiento**, que junto a la temperatura definirán la capacidad de movimiento de las soluciones. La velocidad dependerá del factor de enfriamiento geométrico

$$T_{n+1} = \alpha T_n, \text{ donde } \alpha < 1, \text{ muy cercano a } 1$$

$$T_{n+1} = \frac{1}{1 + \beta T_n}, \text{ donde } \beta \in \mathbb{R} \text{ cercano a } 0.$$

- **Número de Iteraciones**, habrá que fijar un número máximo de iteraciones de búsqueda de solución, ya que en ciertas ocasiones se podrá aumentar la temperatura para realizar más movimientos, por lo que sería posible entrar en un ciclo que se repite de forma indefinida.
- **Criterio de Parada**, utilizar el óptimo global como criterio de parada es útil para cuando se conozca dicho valor como en las funciones test, pero en nuestro problema se utilizará el máximo de iteraciones o que la temperatura alcance un valor casi nulo, por analogía física que se “congele el sistema”.

La estructura del algoritmo del Temple Simulado o Recocido Simulado se muestra en el diagrama representado en la figura 11, y a continuación se detallará una descripción específica y formal del algoritmo completo (SA).

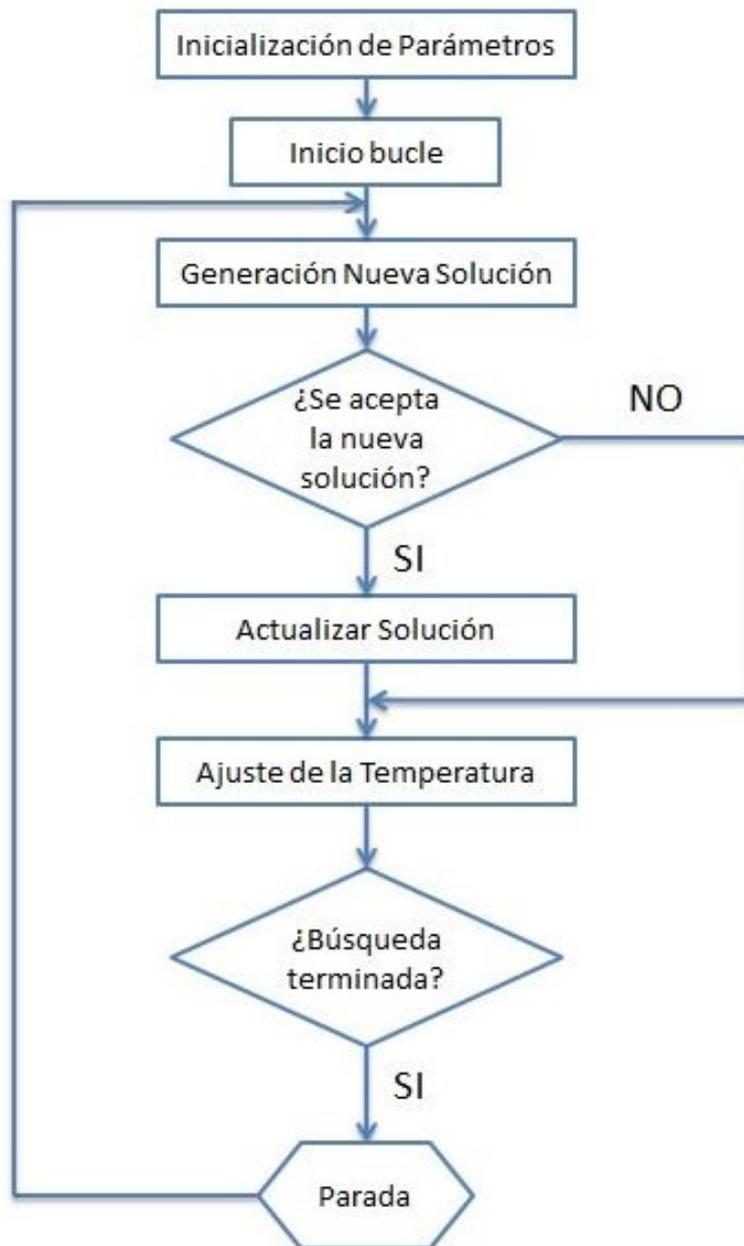


FIGURE 11. Estructura principal del algoritmo del Temple Simulado

Algoritmo Temple Simulado $SA(f, x_0^{(0)})$

Etapa 1. Inicialización

Se eligen T_0 la temperatura inicial, α factor de enfriamiento $\alpha < 1$ muy cercano a 1, M número máximo de iteraciones permitidas, tol una variable que marca la tolerancia aceptada para seguir ejecutando una búsqueda por el descenso en la dirección del gradiente o bien utilizar una búsqueda aleatoria en el dominio continuo de la función de coste f . Se toma x_0 como valor inicial, y se guarda el valor de su función de coste $c = f(x_0)$.

Etapa 2. Búsqueda de Posibles soluciones (Bucle principal).

Sea $I := 0$ y se repite el bucle hasta $T < 0.1$ o que las iteraciones alcancen al máximo M .

Etapa 2.1. Búsqueda con el descenso en la dirección del gradiente

Si $contador > tol$, se salta al paso 2.2.. Si no se genera el siguiente punto

$$x_{k+1} = x_k - \eta \nabla f(x_k),$$

$$contador = contador + 1.$$

Se puede modificar el parámetro η para que la solución pertenezca al dominio de continuidad de la función f y se salta al paso 3.

Etapa 2.2. Búsqueda en una dirección aleatoria.

Si $contador > tol$ se genera una dirección aleatoria $d_i \in \mathbb{R}^n$, y un paso Δ_i lo suficientemente pequeño para que no salga del dominio de continuidad de la función f , al ampliar la diversificación de la solución aumentamos un poco la temperatura para aceptar las soluciones, $T = T + \beta$ con $\beta > 0$. Y se pone el contador en nulo $Contador = 0$.

Etapa 3. Aceptación de solución

Se calcula $\delta = f(x_{k+1}) - f(x_k)$, y se genera de forma pseudo-aleatoria un valor $u \stackrel{d}{=} U(0, 1)$ entonces si $\delta < 0$ o $(\delta \geq 0 \wedge u < \exp(-\frac{\delta}{T}))$ entonces se acepta la solución x_{k+1} .

Etapa 3.1. Actualización de los parámetros

Si se acepta x_{k+1} y $f(x_{k+1})$ es la solución óptima actualiza $x = x_{k+1}$. $I = I + 1$ y $T = T * \alpha$.

4. Soluciones del modelo y resultados

En esta sección se explicarán las soluciones de los algoritmos para las funciones test donde se probó la efectividad de los algoritmos, luego se explicará su aplicación al problema industrial real como es la función objetivo que se quiere minimizar, y por último las soluciones del problema industrial real.

La programación de los algoritmos se ha desarrollado con MATLABTM y el modelo físico con COMSOL MultiphysicsTM, se conectaron ambos programas para generar las salidas y conseguir la solución necesaria. Para la resolución se ha utilizado un ordenador con un procesador Triple-Core 2.1 GHz y 4GB de RAM, con procesador de 64 bits.

Experimentos numéricos

Para chequear la eficiencia del algoritmo implementado de la búsqueda tabú y del temple simulado descritos en la sección 3, les aplicaremos algunas problemas de test para problemas de optimización: Sphere, Branin, Easom, Goldstein-Price, Rosenbrock y Dixon-Price.

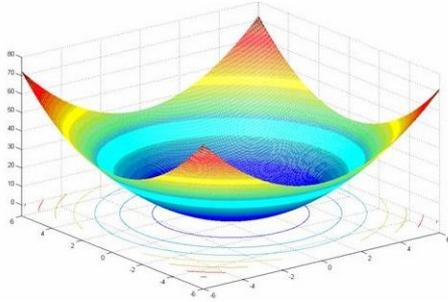
Estas funciones son usadas frecuentemente para validar algoritmos, representan una amplia variedad de dificultades en problemas de optimización por lo que son ideales para verificar diversas propiedades de nuestros nuevos algoritmos, algunas de las propiedades que se desean verificar son las siguientes:

- Velocidad de convergencia
- Precisión
- Robusteza

También se han utilizado para estimar algunos parámetros de los algoritmos, en el caso del temple simulado para comprobar que temperatura inicial es óptima o la velocidad de enfriamiento, la búsqueda tabú directa tiene muchos más parámetros como el radio de las bolas de la lista de las regiones visitadas o el límite de elementos en la lista tabú.

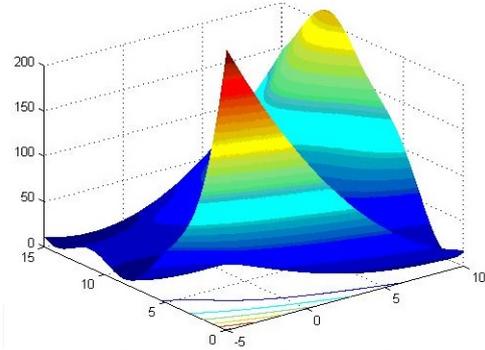
A continuación se mostrará una completa descripción de las funciones.

Función Sphere



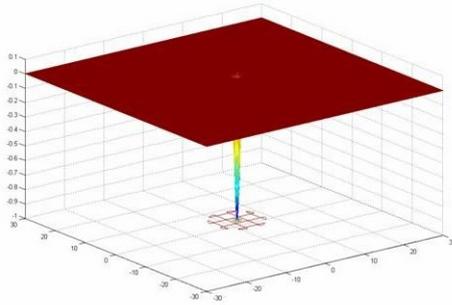
$$f(x) = \sum_{i=1}^n x_i^2$$

Función Branin



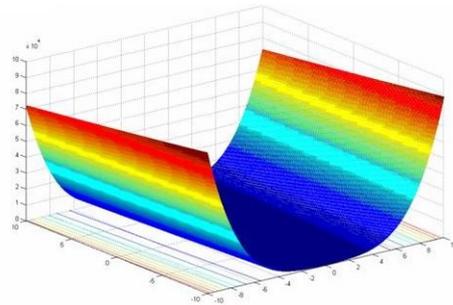
$$f(x) = (x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6)^2 + 10(1 - \frac{1}{8\pi})\cos(x) + 10$$

Función Easom



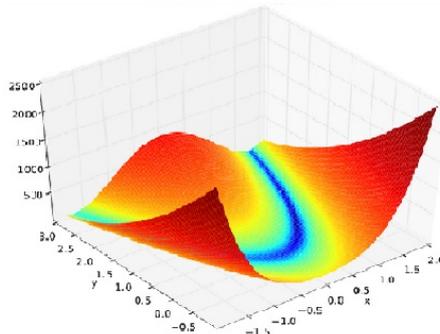
$$f(x) = -\cos(x_1)\cos(x_2)\exp(-((x_1 - \pi)^2 + (x_2 - \pi)^2))$$

Función Dixon-Price



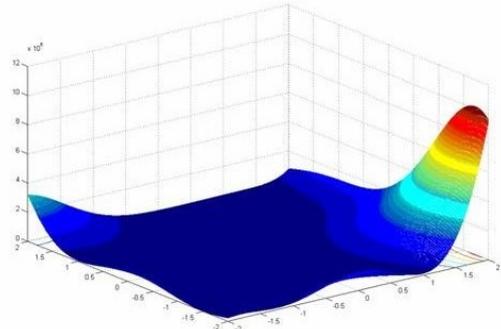
$$f(x) = (x_1 - 1)^2 + 2(2x_2^2 - x_1)^2$$

Función Rosenbrock



$$f(x) = 100(x_2 - x_1^2)^2 + (x_1 - 1)^2$$

Función Goldstein-Price



$$f(x) = (1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 13x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)) \\ (30 + (2x_1 - 3x_2)^2(18 - 32x_1 + 12x_1^2 - 48x_2 - 36x_1x_2 + 27x_2^2))$$

Como de las diferentes funciones test que utilizamos conocemos el mínimo global de cada una de ellas h_0^* , por lo que tomaremos el siguiente criterio de parada:

$$|h_0^* - \tilde{h}_0| \leq \epsilon_1 |h_0^*| + \epsilon_2.$$

Donde \tilde{h}_0 es la solución obtenida por el algoritmo, $\epsilon_1 = 10^{-4}$ y $\epsilon_2 = 10^{-6}$. Los valores ϵ_i miden la precisión que se quiere obtener de las soluciones encontradas.

Se añade este criterio de parada a los nuevos algoritmos implementados y se registra el número de llamadas medio de cada uno de ellos para las diferentes funciones test.

A continuación se muestran algunos resultados obtenidos mediante el algoritmo de temple simulado en la tabla 2, para algunas funciones test no funcionaba correctamente, así que se ha tenido que ir ajustando el algoritmo para estas funciones, el motivo principal por el cual no funcionaban correctamente ha sido que en algunas funciones el descenso en la dirección del gradiente no funcionaba adecuadamente, porque la función del gradiente era bastante compleja.

Los parámetros del algoritmo del temple utilizados son la temperatura inicial $T = 1500$, el factor de enfriamiento $\alpha = 0.9$, máximo de iteraciones *maximo* = 10^6 , el parámetro η que acompaña al vector opuesto al gradiente se inicia en $\eta = 0.007$, pero en el caso que la nueva solución se salga fuera de los límites del rango de las funciones se utilizando $\eta = 0.007/k$ con $k = 2, 3, 4, \dots$, tolerancia máxima *tol* = 15.

Se han hecho 100 simulaciones de soluciones iniciales aleatorias dentro del rango de cada una de las funciones test, y se ha calculado en la primera columna la media de llamadas a la función en los casos en los que se satisfacía el criterio de parada definido anteriormente y en la segunda columna es el porcentaje de soluciones que han cumplido el criterio de parada.

Función	Media llamadas	% criterio de parada
Sphere (n=3)	20	100
Dixon & Price (n=2)	584	51
Branin (n=2)	49	100
Easom (n=2)	12388	43
Goldstein-Price (n=2)	126906	100
Rosenbrock (n=2)	1000000	87

TABLE 2. Resultados del algoritmo del Temple simulado, número medio de llamadas y % que cumple el criterio de parada

Como se puede ver en la tabla 2 las funciones con las que más problemas se ha tenido han sido Goldstein-Price y Rosenbrock, ya que la dirección de búsqueda del mínimo era aleatoria, en el caso de la función Rosenbrock se llega al límite de iteraciones del algoritmo, si se cambia la precisión del criterio de parada a $\epsilon_1 = 10^{-3}$ y $\epsilon_2 = 10^{-4}$. Y la función Easom que puede ser la más compleja no se llega a alcanzar en muchas de las simulaciones.

Ahora nos centraremos en el algoritmo de la búsqueda tabú directa, hay muchos parámetros que son libres para su cambio según convenga. Los valores de entrada serán la función a minimizar, un valor inicial, y el rango de cada una de las funciones en las que se quiere encontrar el mínimo global. Los radios de las esferas de las regiones tabú, semi-regiones tabú, regiones visitadas y el paso de dirección tienen que tener el siguiente comportamiento para que el algoritmo funcione correctamente,

$$r_{TR} < r_{STR} < \Delta_i < \rho,$$

$$\text{con } \rho = 0.25\delta,$$

$$\Delta_i = (0.1 + 0.025w)\delta,$$

$$r_{TR} = 0.005\delta \text{ y } r_{STR} = 2r_{TR},$$

$$\text{siendo } \delta := \max_{l \leq i \leq n} (u_i - l_i),$$

donde los valores u y l son los valores de los límites de cada una de las variables. En el estudio de las regiones visitadas se comprobó que el mejor valor para γ es $\gamma = 0.25$, ya que valores superiores a 0.4 falla la exploración. En la fase de intensificación se decide tomar las 3 mejores soluciones halladas, y centrar en ellas una búsqueda más intensa.

A continuación en la tabla 3 se muestra algunos resultados obtenidos mediante el algoritmo de la búsqueda tabú directa, a diferencia del temple simulado en este caso no se ha podido obtener los valores de llamadas media de cada función test, ya que en el algoritmo se utiliza una caja negra de búsqueda lineal no pudiéndose calcular la media. Se realizan 100 simulaciones de soluciones iniciales aleatorias dentro del rango de cada una de las funciones test.

Función	% criterio de parada
Sphere (n=3)	84
Dixon & Price (n=2)	100
Branin (n=2)	99
Easom (n=2)	0
Goldstein-Price (n=2)	62
Rosenbrock (n=2)	39

TABLE 3. Resultados del algoritmo de la búsqueda tabú, el % que cumple el criterio de parada

Como podemos ver tenemos problemas principalmente con la función Easom, esta función es muy uniforme en 0 pero tiene un pico hasta su mínimo global que es $f(\pi, \pi) = -1$, pero el área alrededor de este pico es prácticamente 0, el problema de este algoritmo al aplicarse para esta función es que si se fijan los valores de los radios de las regiones y semi-regiones tabú y regiones visitada que no sea lo suficientemente pequeño en esta zona en la que se encuentra el mínimo global no se buscara de forma intensa, si se cambiasen estos valores a extremadamente pequeños, el algoritmo para esta función debería funcionar eficientemente, pero con las demás funciones haría muchísimas llamadas a la función y necesitaría muchas más repeticiones sobre todo en la fase de exploración y diversificación, por eso se decide mantener el algoritmo, porque funciona de forma correcta con el resto de funciones.

Optimización industrial

Las formas de los modelos de aislante han sido diseñados mediante COMSOL Multiphysics 3.5aTM, y una interfaz entre los modelos de COMSOL y Matlab 2011bTM, lo primero que se hace es generar los modelos que se resuelven y se almacenan automáticamente para su análisis posterior, el siguiente paso es convertir los datos de COMSOL a matrices de Matlab con una resolución espacial de 0.01mm, después se aplica la evaluación del modelo.

Como se ha visto en la sección 2, para el tratamiento mediante el PEF en las cámaras, existe un importante número de parámetros dependientes e independientes a tener en consideración. El volumen de la zona tratada, la uniformidad de la distribución del campo eléctrico, incluyendo la prevención de áreas con picos de altos niveles de campos eléctricos y también la presión causada por las diferencias de diámetro entre el aislante y los electrodos.

Se seleccionan cinco parámetros capaces de ser medidos y se crea una ecuación derivada de estos parámetros que expresa el rendimiento de cada escenario en un sólo parámetro, esta función generará la función objetivo que incluiremos en nuestros algoritmos, el parámetro sin dimensión de rendimiento DPP (Dimensionless Performance Parameter), los cinco parámetros que se tienen en cuenta están descritos en las ecuaciones 13-17 y el parámetro a generado por ellos está descrito en la ecuación 12:

$$(12) \quad DPP = TVV^{a_1} \cdot PDV^{a_2} \cdot MV^{a_3} \cdot UV^{a_4} \cdot EPV^{a_5}.$$

Con:

$$(13) \quad TVV = \frac{V_{zona}}{V_{max}},$$

$$(14) \quad PDV = \frac{(d - ins)^4}{d^4},$$

$$(15) \quad MV = \frac{E_{av}}{\frac{V_0}{h_{min}}},$$

$$(16) \quad UV = \frac{n_{av \pm porcentaje}}{n_{total}}$$

y

$$(17) \quad EPV = \frac{E_{av}}{E_{max}}.$$

Donde V_{zona} es el volumen de la zona tratada (región aislante) del respectivo escenario, V_{max} es el volumen más grande que se puede tratar. E_{av} es la media de la fuerza del campo eléctrico en la región aislante, V_0 es el potencial aplicado inicialmente, h_{min} es el mínimo de la distancia de los electrodos de todos los escenarios investigados se toma $h_{min} = 1mm$, $n_{av \pm porcentaje}$ es el número de elementos con una fuerza de campo eléctrico superior a un 10% de la media, n_{total} es el número total de elementos en la zona tratada y E_{max} es el máximo de la fuerza de campo eléctrico en cada escenario.

Todos los parámetros tienen valores entre 0 y 1, por lo tanto DPP tendrá también valores entre 0 y 1 con 1 siendo el valor óptimo.

El primer parámetro es la variable del volumen de tratamiento TVV (Treatment Volume Variable) que expresa el volumen de la zona de tratamiento relacionada con el volumen de la mayor zona tratada investigada, donde el diámetro del aislante es igual al diámetro del electrodo.

El segundo parámetro es la variable de pérdida de presión PDV (Pressure Drop Variable) que expresa la pérdida de presión provocada por la reducción del diámetro del aislante que se deriva de la ecuación simplificada de Bernuolli 18:

$$(18) \quad \frac{\rho \cdot v_1^2}{2} + p_1 = \frac{\rho \cdot v_2^2}{2} + p_2.$$

El tercer parámetro es la variable de magnitud MV (Magnitud Variable) que expresa la magnitud de la fuerza del campo eléctrico del modelo analizado, está relacionada a la media del campo eléctrico de la región aislante.

El cuarto parámetro es la variable de uniformidad UV (Uniformity Variable) que indica la uniformidad del campo eléctrico en la región hueco. Este parámetro cuenta con todos los elementos que tienen un campo eléctrico con un 10% superior a la media y se relativiza con todos los elementos del área.

El último parámetro es la variable de los picos del campo eléctrico EPV (Electric field strength Peak Variable) que indica la magnitud de los picos de campo eléctrico de la zona tratada que relaciona el valor medio con el valor máximo de la respectiva configuración.

Como se puede ver la mayoría de los parámetros tienen incluidos variables que son características geométricas del aislante, que serán las variables de entrada de la función objetivo y como valor inicial del algoritmo, estas variables son dos alturas, dos longitudes, una altura de la elipse, una longitud de la elipse, y el séptimo parámetro es la curvatura de la elipse.

Los exponentes a_1, a_2, a_3, a_4 y a_5 pueden ser ajustados según la importancia y relevancia en el modelo de cada uno de los parámetros. Incrementando los exponentes se le dá más importancia a los respectivos parámetros. Debido al estudio de Kai Knoerzer et al. [1], hay que poner más énfasis en la variable de uniformidad de la distribución del campo eléctrico y prevención de picos. Por lo que los exponentes de MV, UV y EPV deben ser superiores a TVV y PDV.

Se asume que TVV y PDV tienen la misma importancia sus valores deben ser similares, aunque se hace que el exponente de PDV sea ligeramente superior, ya que se necesita que la pérdida de presión se mantenga baja por eso se fijan $a_1 = 0.9$ y $a_2 = 1$. MV y EPV aseguran la eficiencia y seguridad del tratamiento controlando la concentración del campo eléctrico en la zona tratada y previniendo de picos pronunciados que provocarían como hemos explicado en la sección 2 sobreprocesamiento del alimento y erosión en los electrodos y en el aislante, por lo que se decide darles la misma importancia y fijando los valores en $a_3 = a_5 = 1.5$. Como se ha comentado el principal objetivo de este problema de optimización es la uniformidad del campo eléctrico por lo que el exponente más alto (por importancia) es el aplicado a la variable de uniformidad UV siendo $a_4 = 2$.

Incluyendo todos los exponentes, los parámetros dan salida valores entre 0 y 1, y de la misma manera será el parámetro principal a optimizar, el valor óptimo se encuentra en el máximo 1, este máximo DPP determina la geometría con el mejor rendimiento para este estudio.

Como se diseñaron los algoritmos de forma que buscasen un mínimo global hay que cambiar este parámetro para buscar el mínimo global en vez del máximo por lo que se toma como función objetivo la función 19:

$$(19) \quad J := \frac{1}{TVV^{a_1} \cdot PDV^{a_2} \cdot MV^{a_3} \cdot UV^{a_4} \cdot EPV^{a_5}}.$$

El máximo de *DPP* corresponde al mínimo de esta nueva función donde los valores de entrada como hemos explicado son las características geométricas del aislante.

La figura 12 muestra el algoritmo de optimización en un diagrama de flujo, marcando los diferentes pasos desde la generación del modelo, la solución de el, la extracción de datos y evaluación del rendimiento de los respectivos escenarios, encontrando la mejor configuración geométrica de nuestro problema.

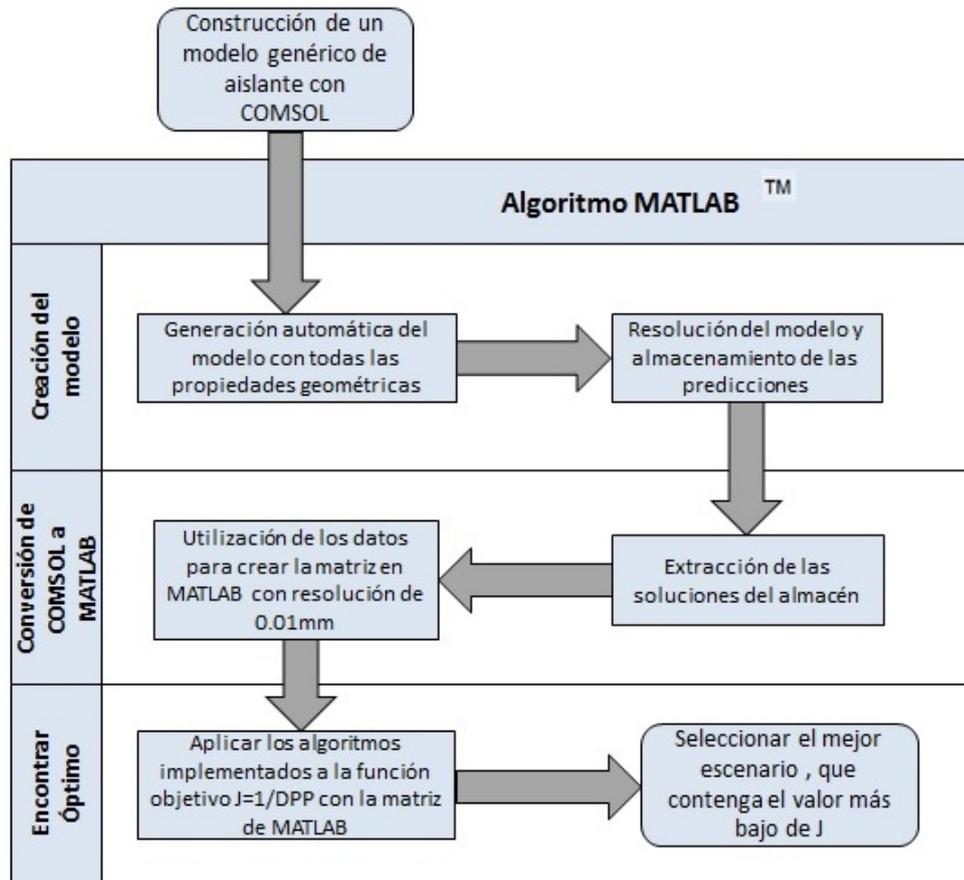


FIGURE 12. Diagrama de flujo del algoritmo de optimización (paso inicial: generación del modelo), incluye la conversión de los datos desde COMSOL Multiphysics TM a MATLAB TM y la determinación del óptimo

Aplicación al problema industrial

Se utilizará como valor inicial a los dos algoritmos implementados en Matlab una de las soluciones dada por Kai Knoerzer et al. [1], en su estudio desarrollado generaron más de 100000 modelos 2D simétricos con respecto a los ejes de las 4 formas que se mostraron en la sección 2, de estos modelos con el que mejor resultado obtienen es con el modelo rectangular con las esquinas redondeadas, a continuación aparecen algunas simulaciones sobre la distribución del campo eléctrico en diferentes formas, se puede comprobar la uniformidad en los modelos que hemos comentado en la figura 13.

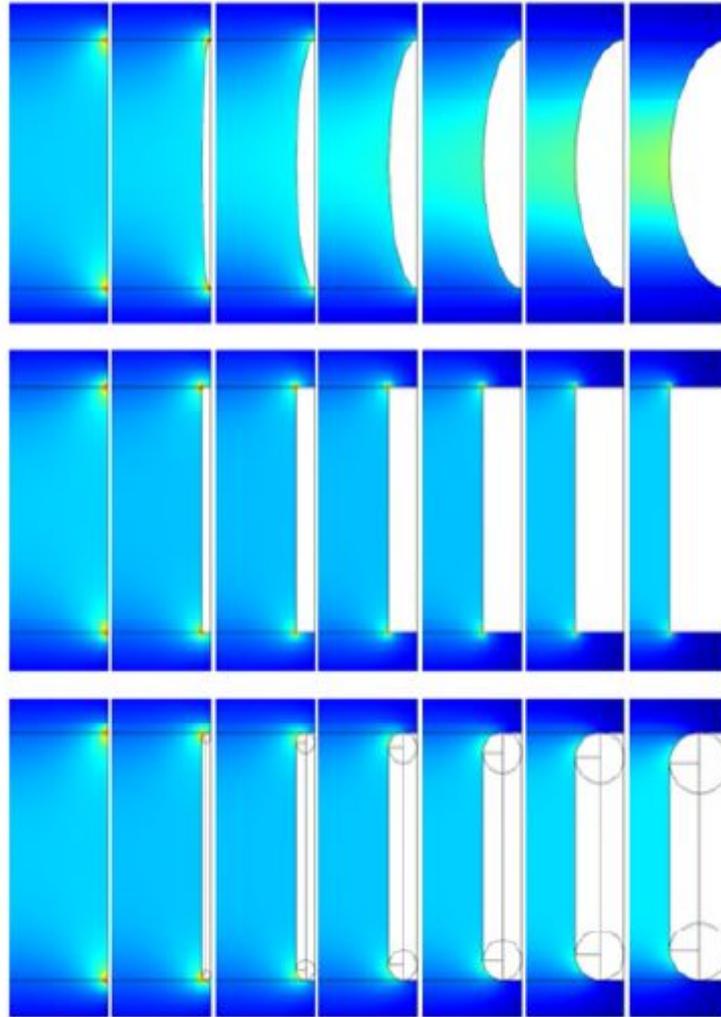


FIGURE 13. Ejemplos de las simulaciones de la distribución de campo eléctrico en diferentes geometrías

La configuración inicial de la que se va a partir, es la que cuenta con los siguientes parámetros

$$\text{Altura1} = 0.25$$

$$\text{Altura2} = 0.3\hat{3}$$

$$\text{AlturaElipse} = 0.5$$

$$\text{Longitud1} = 0.2$$

$$\text{Longitud2} = 0.2$$

$$\text{LongitudElipse} = 0.3$$

$$\text{CurvaturaElipse} = 1$$

La estructura de este aislante con un mallado triangular es la figura 14

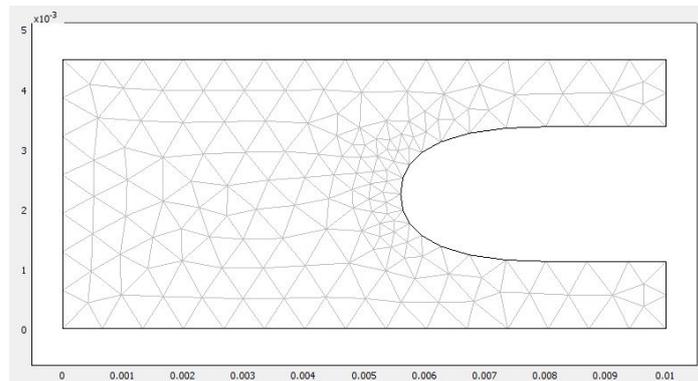


FIGURE 14. Estructura de malla triangular de la solución inicial x_0

El valor de la función de coste en la solución inicial x_0 es

$$J(x_0) = 5.7527 \cdot 10^5.$$

Esta estructura es la 2D-simétrica con respecto a un eje, la estructura en 3d se muestra en la figura 15 y en la figura 16

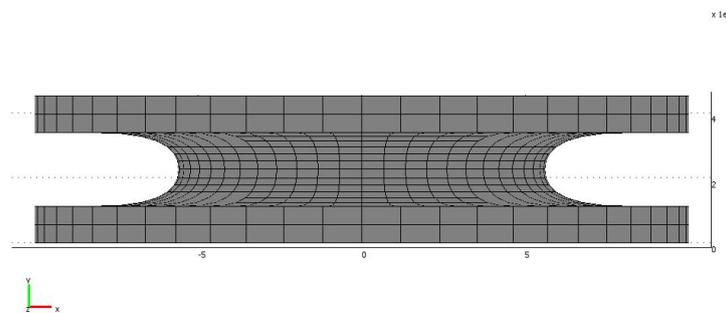


FIGURE 15. Rotación sobre el eje Y de la solución inicial x_0

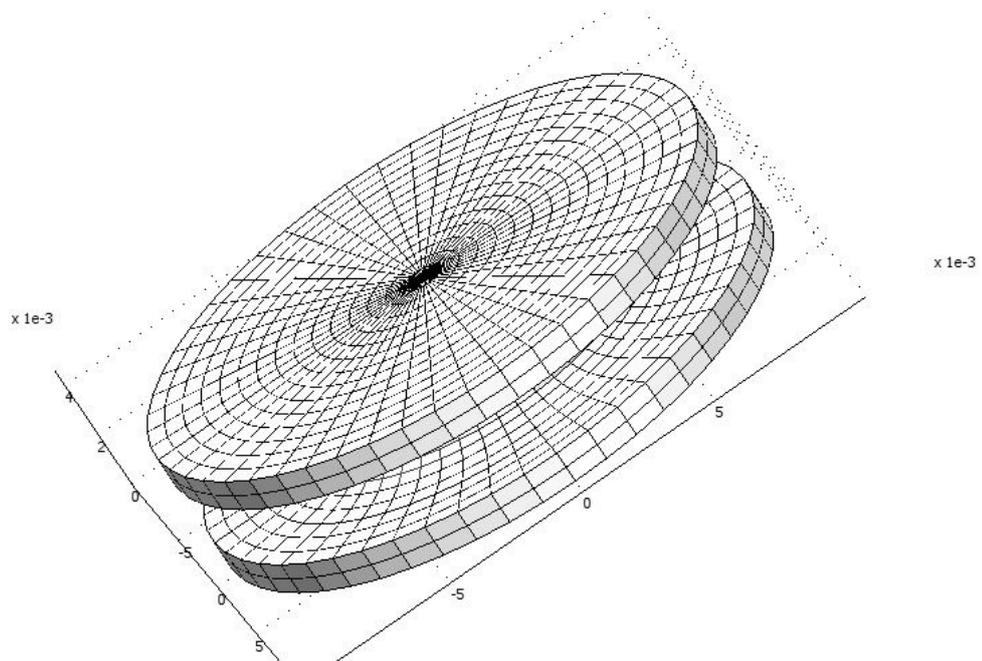


FIGURE 16. Rotación sobre el eje Y de la solución inicial x_0

La solución del problema físico en la estructura inicial x_0 se muestra en la figura 17

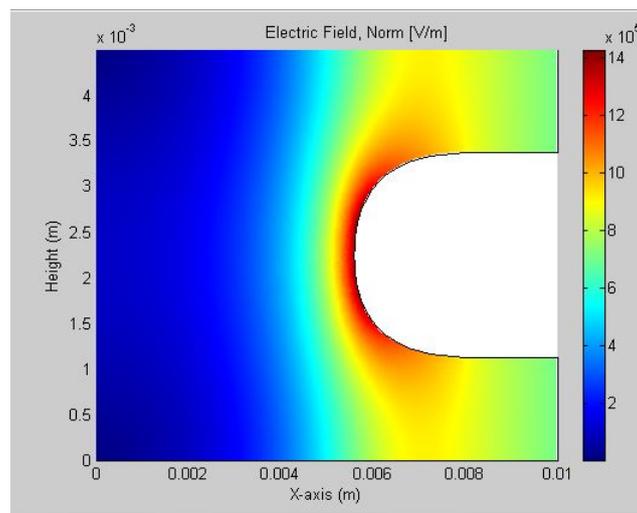


FIGURE 17. Distribución del campo eléctrico en la geometría de x_0

Los parámetros que definen el parámetro principal DPP en x_0 son los siguientes:

$$TVV = 0.6, PDV = 0.2401, MV = 0.0868, UV = 0.0565 \text{ y } EPV = 0.2702.$$

Lo que se quiere al optimizar esta función de coste con los correspondientes coeficientes, es maximizar lo máximo posible los parámetros EPV , UV y MV en ese orden y se quiere que sean superiores a TVV y PDV , para que la distribución del campo eléctrico sea lo más uniforme posible.

El primer algoritmo que aplicaremos será el algoritmo de la Búsqueda Tabú Directa, se inició con la solución anterior llamamosla x_0 , y se obtuvo $x_1 = DTS_{NMS}(x_0, J)$, y se repitió el proceso $x_2 = DTS_{NMS}(x_1, J)$, esto llevó bastante tiempo de computación como muestra la figura 18:

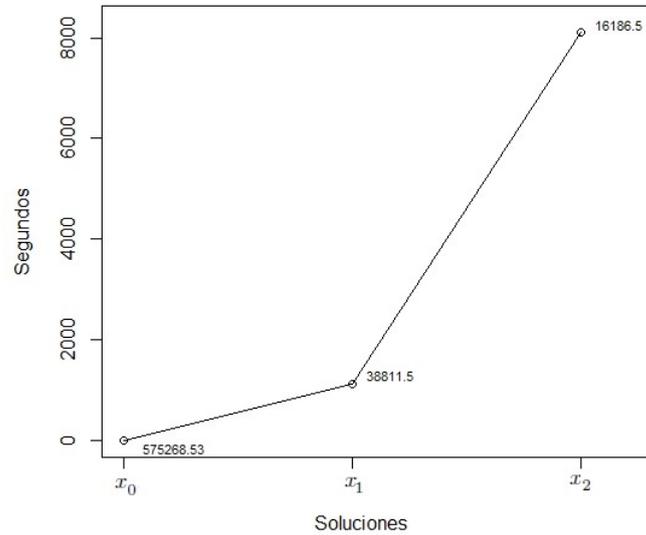


FIGURE 18. Soluciones en el eje X, tiempo en el eje Y, coste en la función objetivo en cada punto respectivamente, x_0 solución inicial, $x_1 = DTS_{NMS}(x_0, J)$ y $x_2 = DTS_{NMS}(x_1, J)$

La solución x_2 , tiene las siguientes valores que definen sus características geométricas:

$$Altura1 = 0.0897$$

$$Altura2 = 0.2293$$

$$AlturaElipse = 0.4918$$

$$Longitud1 = 0.0938$$

$$Longitud2 = 0.5459$$

$$LongitudElipse = 0.5463$$

La figura 19 refleja claramente que esta solución x_2 tiene una configuración muy diferente a la anterior, y a cualquiera simulada en Kai Knoerzer et al. [1],

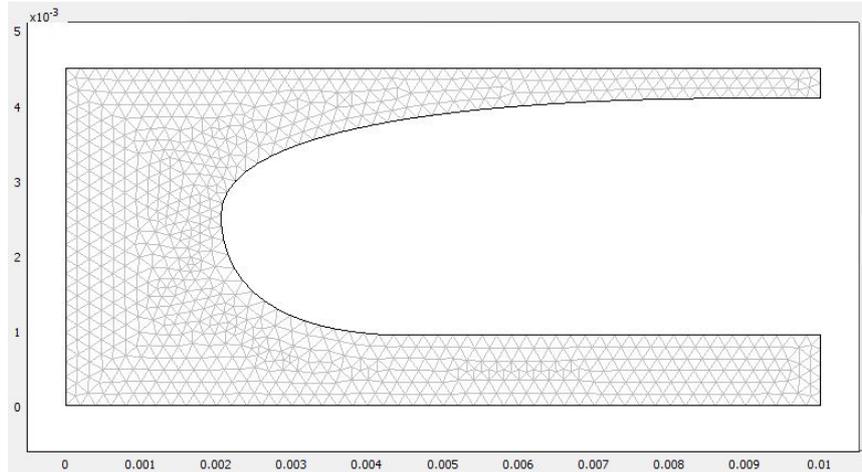


FIGURE 19. Estructura de malla triangular de la solución encontrada x_2 mediante el algoritmo DTS

La solución al problema físico con la configuración x_2 se muestra en la figura 20 y su función de coste de x_2 es

$$J(x_2) = 1.6187 \cdot 10^4.$$

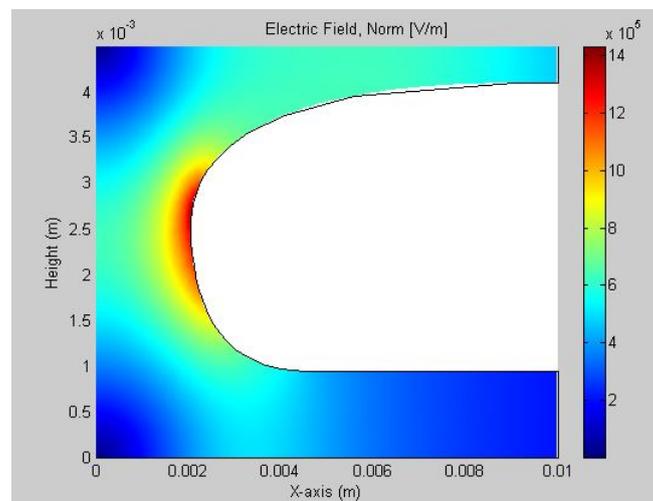


FIGURE 20. Distribución del campo eléctrico en la geometría de x_2

Los parámetros que definen el parámetro principal DPP en x_2 son los siguientes:

$$TVV = 0.28821, PDV = 0.0423, MV = 0.1993, UV = 0.4158 \text{ y } EPV = 0.43853.$$

Estos valores mejoran en términos generales a la geometría inicial. En la figura 20 ya es visible que la distribución del campo eléctrico es más uniforme que con la configuración inicial x_0 . Esto se puede verificar estudiando los parámetros de la función objetivo, los nuevos valores de MV , UV , y EPV son mayores consiguiendo minimizar los problemas provocados por una distribución no uniforme del campo eléctrico. Uno de los valores más importante es $EPV = 0.4385$ indica que la media del campo eléctrico en la zona tratada es de aproximadamente un 44% de la fuerza máxima del campo eléctrico en la cámara de tratamiento. Y el parámetro más importante UV significa que aproximadamente el 42% de la zona tratada está expuesta a un campo eléctrico de $\pm 10\%$ de la media. Los parámetros TVV y PDV empeoran pero esto se entiende ya que se trata de optimización multiobjetivo y se da el caso de que al mejorar unos objetivos se empeoran otros. La tabla 4 ilustra la mejora de los parámetros con respecto a la solución inicial.

Parámetros	x_0	x_2
TVV	0.6	0.288
PDV	0.2401	0.042
MV	0.0868	0.199
UV	0.0565	0.416
EOV	0.2702	0.438

TABLE 4. Comparación de los parámetros de la función objetivo, siendo x_0 la solución inicial y $x_2 = DTS_{NMS}(x_1, J)$

Rotando la forma obtenida en la figura 19 conseguimos la geometría completa en 3D como ilustra las siguientes figuras 21 y 22:

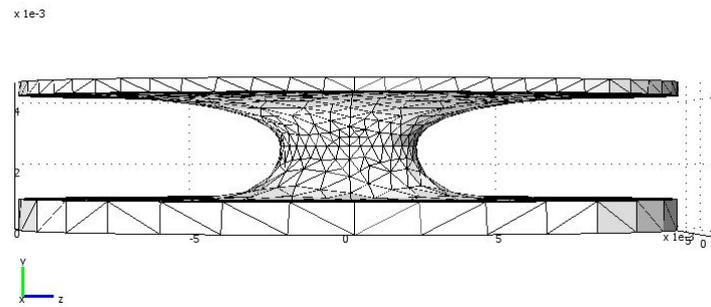


FIGURE 21. Rotación sobre el eje Y de la solución encontrada x_2

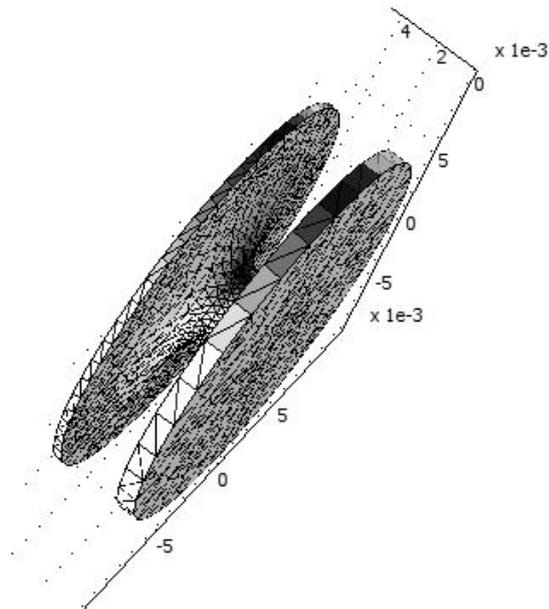


FIGURE 22. Rotación sobre el eje Y de la solución encontrada x_2

El segundo algoritmo que se aplica es algoritmo del Temple Simulado, se inició con la solución x_0 ya mostrada su configuración anteriormente en la figura 14.

Se obtuvo una solución x_1 que no mejora la función coste J (ecuación 19 de esta misma sección), se consiguieron unos valores altos para los parámetros TVV y PDV , pero para los parámetros que tienen más peso en la función objetivo por sus coeficientes MV , UV y EPV son extremadamente bajos haciendo que el parámetro general DPP sea muy pequeño en consecuencia la función objetivo J sea altísima.

Se probó a inicializar desde diferentes soluciones en el dominio de la función pero no se consiguieron resultados mejores que los que proporciona la solución x_0 . Que no funcione este algoritmo correctamente puede deberse a que el método de elementos finitos, que aplica el software COMSOL, tiene errores numéricos que se acumulan y si estos errores son más grandes que la variación de los parámetros del algoritmo puede provocar que no se encuentre una solución factible. Y la variación de los parámetros puede ser debida a la estimación del gradiente de la función objetivo de 7 dimensiones.

5. Conclusiones y futuros trabajos

En este trabajo se pretendía estudiar diferentes algoritmos de optimización global para funciones de dominio continuo para aplicarlos a casos industriales reales. Se desarrollaron los algoritmos de la Búsqueda Tabú y el Temple Simulado. También se estudio un caso práctico industrial de tratamiento de alimentos mediante campos eléctricos donde se aplicaron estos dos algoritmos, una vez que fuesen testeados con algunas funciones que son usualmente utilizadas para la comprobación de algoritmos de búsqueda de mínimos globales.

Los resultados numéricos de este testeo mostraron que el algoritmo del Temple Simulado proporcionaba buenas soluciones para las funciones consiguiendo un gran porcentaje de acierto al encontrar el mínimo global, en cambio el algoritmo de Búsqueda Tabú no conseguía tan buenos resultados como el Temple Simulado. Esto es interesante ya que al aplicar ambos algoritmos al caso industrial fue el algoritmo de la Búsqueda Tabú el que encontró una buena solución al problema.

El motivo del fallo del algoritmo del Temple Simulado es un tema interesante de estudio ya que puede haber diferentes razones, una de ellas podría ser el algoritmo de búsqueda directa que utiliza ya que la estimación del gradiente es bastante compleja y se están buscando soluciones en 7 dimensiones.

En futuros trabajos es posible el estudio más exhaustivo de la función de coste J propuesta por Kai Knoerzer et al. [1], por ejemplo variando los coeficientes para dar diferentes pesos a los parámetros dentro de esta función objetivo, pudiendose comprobar que coeficientes serán optimos en este caso industrial.

Otra opción es desarrollar otros algoritmos que encuentren mejores soluciones como algoritmos Genéticos.

También como se trata de optimización multiobjetivo se podría estudiar una metodología de teoría de juegos, pero en este trabajo por falta de tiempo no ha podido desarrollarse.

REFERENCES

- [1] K. Knoerzer, P. Baumann y R. Buckow. *Pulsed Electric Field (PEF) Treatment Chamber Optimisation*.
- [2] A.R. Hedar y M. Fukushima. *Tabu Search by direct search methods for nonlinear global optimization*. *European Journal of Operations Research*, 170 329–349, 2006.
- [3] R. Chelouah., P. Sarry. *Tabu Search applied to global optimization*. *European Journal of Operations Research*, 123 256–270, 2000.
- [4] F. Glover. *Tabu Search-Part I*. *ORSA journal on Computing* 1, 190–206, 1989.
- [5] F. Glover. *Tabu Search-Part II*. *ORSA journal on Computing* 2, 4–32, 1990.
- [6] A.R. Hedar y M. Fukushima. *Heuristic pattern search and its hybridization with simulated annealing for nonlinear global optimization*, *Optimization Methods and Software*, 19 291–308, 2004.
- [7] C.T. Kelley. *Detection and remediation of stagnation in the Nelder-Mean algorithm using a sufficient decrease condition*, *SIAM Journal on Optimization*, 10 43–55, 1999.
- [8] F. Rothlauf. *Design of Modern Heuristics Springer*, 2011.
- [9] S. Kirkpatrick, C.D. Gelatt y M.P. Vecchi. *Optimization by simulated annealing*. *Science Vol 220 No 4598*, 671-680 1983.
- [10] R. Buckow, R. Schroeder, P. Baumann, P. Berres y K. Knoerzer. *Simulation and Evaluation of Pilot-Scale Pulsed Electric Field(PEF) Processing*, *Journal of Food Engineering*, 2010.
- [11] J.A. Nelder y R. Mead. *A simplex method for function minimization* *The computer journal*,7 308-313 1965.
- [12] N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller y E. Teller. *Equation of State Calculations by Fast Computing Machines*, *J.Chem Phys.*,21(6) 1087-1092 1953.