University of Padova

Department of Physics and Astronomy ”Galileo Galilei”

Volvo Car Corporation

Propulsion and Energy ”Strategy and Execution”

Master Thesis in physics of data

Research on Optimizing Passive Filter Weight in Automotive Applications While Maintaining Performance

Supervisor
Prof. Marco Zanetti
University of Padova

Industrial-Supervisor
Dr. Raik Orbay
Industrial-Co-Supervisor
Dr. Bahram Ganjipour
Volvo Cars

Master Candidate
Ms.Roya Joulaei Vijouyeh

Student ID
2050954

Academic Year

2023-2024

The world is a mountain in which your words are echoed back to you
Rumi (Molana), Divan-e Shams

Abstract

Passive filters are widely used in high and low-voltage electricity to remove unwanted frequencies in power supplies, and communication devices, ensuring EMI-free operation which stands for an operation without Electromagnetic Interference. These filter designs protect sensitive equipment from being damaged by high levels of electrical noise. Despite their effectiveness, passive filters are typically heavy and expensive. We want to design a new configuration of passive filter that optimizes the weight and maintains performance levels. By having a quasi-3D simplified model capable of importing finite element mesh files, we categorize the mesh elements corresponding to ferromagnets and air with labels '1' and '0', respectively. This approach is known as topology optimization and draws inspiration from the Ising model in statistical mechanics, where the goal is to achieve equilibrium by minimizing the system's energy, governed by the Hamiltonian operator. By utilizing this framework, we aim to identify the minimum energy configuration, analogous to the behavior observed in ferromagnetic systems within the Ising model. The aim is to maximize the number of zeros, corresponding to mesh elements made of air, thereby reducing the amount of ferromagnets used to decrease weight and cost. This approach is a combinatorial optimization which is a method used to find the optimal solution from a finite set of discrete solutions. This problem belongs to the NP-hard class in complexity computation and does not have an effective polynomial-time solution.

In this work, the problem is approached using three distinct AI-based methods: a Random Agent, a Simulated Annealing Agent, and a Simulated Annealing Agent with tunneling. The Random Agent serves as a baseline model, making decisions randomly without considering energy minimization or the state of the system. While simplistic, it provides a reference for evaluating the effectiveness of more advanced strategies. The Simulated Annealing Agent, inspired by the physical annealing process, iteratively explores the solution space by allowing occasional increases in energy, thereby avoiding local minima and gradually converging to a low-energy configuration. This method aligns closely with the behavior of systems seeking equilibrium, as modeled in the Ising framework. Finally, we extend the Simulated Annealing approach by introducing quantum tunneling, a mechanism that enables the agent to overcome high energy barriers by "tunneling" through them, rather than escaping slowly via probabilistic jumps. This allows the system to better explore complex energy landscapes and increase the likelihood of reaching the global energy minimum. By comparing these methods, we aim to identify the most effective strategy for minimizing the system's energy and achieving equilibrium, in line with the Hamiltonian-inspired behavior of the Ising model.

As an application of the combinatorial optimization inspired by quantum processes, we include a combinatorial optimization example with the knapsack problem, which is intended to be solved on a real quantum computer. To approach this, we map the knapsack Hamiltonian to the Ising model and then convert it into a Quadratic Unconstrained Binary Optimization (QUBO) format, utilizing the `QuadraticProgramToQubo` converter from Qiskit. This approach will help to explore possible solutions that minimize the device's weight with minimal performance loss.

Contents

Listing of figures

Listing of acronyms

FEM	Finite Element Method
RA	Random Agent
SA	Simulated Annealing
tSA	Simulated Annealing with Tunneling
P	Algorithm that solves the task and runs in Polynomial time
NP	Algorithm that solves the task and runs in Nondeterministic Polynomial time
QAOA	Quantum Approximate Optimization Algorithm
AQO	Adiabatic Quantum Optimization
H	Hamiltonian
EBIT	Earnings Before Interest and Taxes
QUBO	Quadratic Unconstrained Binary Optimization
BEV	Battery Electric Vehicles
CFC	chlorofluorocarbons
BiCGStabL	Biconjugate Gradients Stabilized Method
CM Filter	Common Mode Filter
EMC	Electromagnetic Compatibility
EMI	Electromagnetic Interference

1
On Volvo Cars

The car brand that cares about people and the world we live in. The legacy of the Volvo Car Corporation started in 1927 with the belief that they can only make the safest cars for the Swedish roads. Now the brand has entered the last decade to complete its 100 years with a presence in over 100 countries. The company aims to build sustainable, safe, and high-quality cars. According to statistics of 2024, the number of cars that are produced is about 795,000 to 815,000 units per year. The corresponding core EBIT margin reached a record high of 8.1 percent, versus 6.3 percent in the same period last year. This improvement in the underlying profitability was a result of the company's focus on pricing discipline, internal cost control, and sustained growth in sales [7].

1.1 Volvo Cars heritage and a long tradition of sustainability

On the morning of April 14, 1927, the first Volvo car rolled out of the factory gates in Gothenburg, Sweden. Officially named the ÖV4, this inaugural model was an open tourer equipped with a four-cylinder engine. The world's first Volvo was now ready to embark on its journey seen in Fig1.1. Fast forward to September 1, 1944, when Volvo unveiled a car that would become pivotal to its international success. Known as "the little Volvo," the PV 444 symbolized post-war optimism. Within two weeks of its debut in Stockholm, 2,300 orders had already been placed for this promising new model which can be seen in Fig1.2. The Volvo Duett was launched as "two cars in one" – for both work and leisure. It was the first in a long line of estates that have made Volvo synonymous with this type of practical car. Since then, more than 6 million Volvo estates have been made pictured in Fig1.3.

Sustainability, at Volvo Cars, is not just a priority but a fundamental aspect of business operations. The ambition has always been to lead by example, setting high standards for environmental responsibility. This commitment to the environment dates back to the 1940s, and today, it is stronger than ever. For instance, Back in the day, ozone-depleting chlorofluorocarbons (CFC) were often used in car air-conditioning systems. In response to this environmental issue, Volvo cars company launched the world's first car free of CFCs and two years later we eliminated these harmful molecules from the entire product line. andother example, The Volvo V60 was the world's first diesel plug-in hybrid that could be driven on diesel power alone, as a diesel-electric hybrid, or as an electric car. This was something that no other manufacturer could offer at the time, making this model an attractive alternative with low fuel consumption and long range [8].

Figure 1.1: 1927 – First car rolls out from the factory

Figure 1.2: 1944 – The "little Volvo" comes along

Figure 1.3: 1953 – The estate story begins with Duett

1.2 Volvo Cars: Driving Towards Environmental Sustainability

Volvo Cars is deeply committed to sustainability, consistently prioritizing eco-friendly practices in its operations. It is dedicated to use sustainable materials in its manufacturing and aims to reduce its carbon footprint through innovative technologies and energy-efficient processes.

Volvo Cars has set forth ambitious sustainability targets for the year 2030 with the objective of significantly enhancing its environmental stewardship. The key goals include:

- A reduction in **CO_2 emissions per car by 75%** compared to 2018 levels [8].
- A decrease in **energy usage in operations per average car by 40%** compared to 2018 levels.
- Achieving **30% average recycled content** across its fleet, with new car models incorporating at least 35% recycled content [8].
- A **50% reduction in water use** in its operations per average car compared to 2018 levels [8].
- Ensuring that **at least 99% of all waste** from its operations is either reused or recycled [8].

Since launching its sustainability strategy in 2019, Volvo Cars has made significant strides towards these goals:

- Approximately **69% of company operations** are now powered by climate-neutral energy, up from 55% in 2019.

- **100% of the electricity** used across its manufacturing plants globally is climate neutral, an increase from 80% in 2019.
- There has been a **19% reduction in CO_2 emissions per car** since 2018.

This project aims to contribute to carbon dioxide reduction for the environment. By decreasing the amount of ferrite in each vehicle, even by as little as one gram, the cumulative impact becomes substantial over the vehicle's lifetime tracking into account 700K units sold. By decreasing the weight of the vehicle through reduced ferrite content, not only does it contribute to lower CO2 emissions but also enhances the vehicle's performance under the test standards. This ensures that the environmental benefits are both substantial and measurable, helping Volvo Cars meet international emissions standards more effectively.

2
Introduction

The main purpose of this thesis is to minimize the weight of a passive filter, a crucial component used in the charging process of Volvo BEVs. By addressing the issues of its weight and cost, this research aims to optimize the filter design to be both efficient and cost-effective, while maintaining its essential functionality. The potential benefits of reducing the filter's weight are significant. Lighter cars not only decrease the cost of vehicle operation but also contribute to environmental conservation. Fossil fuels should be phased out in order to support global efforts to combat climate change. BEV enables phasing out of fossil fuels. Making the BEV more efficient minimizes electricity generation-related emissions. This approach to minimizing emissions underscores the broader environmental benefits of this innovation. It is worth mentioning that Electromagnetic compatibility (EMC) is the ability of electrical equipment and systems to function acceptably in their electromagnetic environment, by limiting the unintentional generation, propagation and reception of electromagnetic energy which may cause unwanted effects such as electromagnetic interference (EMI) or even physical damage to operational equipment [9]. Electromagnetic compatibility (EMC) is the ability of electrical equipment and systems to function acceptably in their electromagnetic environment, by limiting the unintentional generation, propagation and reception of electromagnetic energy which may cause unwanted effects such as electromagnetic interference (EMI) or even physical damage to operational equipment.
As illustrated in Fig 2.1, as an example Common Mode (CM) passive filter, is chosen which is composed of a ferromagnet and one or two conductors. For the purposes of this project, we focus on a simplified 2D model considering only one conductor. This simplification aids in the analysis and understanding of the CM filter's behavior while maintaining the core functionality in the charging system. The charging process is really important because it can feed spurious signals to the cities electrical power if the high frequencies are not filtered properly.

Minimizing the weight of the component through mathematical optimization involves selecting the optimal solution from a set of available alternatives, guided by specific criteria. This process is categorized into two main subfields: discrete optimization and continuous optimization. Our project specifically utilizes discrete optimization, a method integral to applied mathematics and computer science. In this context, the component of interest

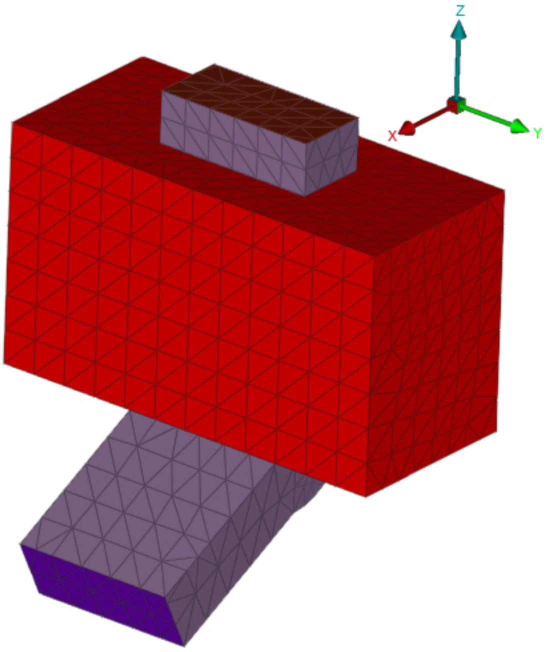


Figure 2.1: Computational mesh unit (red for ferrite and blue for conductor).

is represented by a mesh of 32 domains, where each domain can be either 0 or 1, forming a binary choice system. This binary framework facilitates an efficient and effective resolution to our problem, ensuring that the filter maintains its essential functionality while also becoming lighter and less costly. The discrete nature of our data and the use of linear equations enable the application of Finite Element Method, which will be elaborated upon in subsequent chapters of this report.

Combinatorial optimization is a subfield of mathematical optimization that consists of finding an optimal object from a finite set of objects [10]. These problems can be viewed as searching for the best element of some set of discrete items; therefore, any sort of search algorithm or meta-heuristic can be used to solve them. The goal of meta-heuristics is to generate a set of candidate solutions and select the best one based on the evaluation of the quality of the solutions [11]. Central to this field is the objective function, which serves as a quantitative measure evaluating the ”goodness” of a complex system by seeking to minimize or maximize multiple independent variables.

Many methods are published for the purpose of combinatorial optimization based on the concept of population that manipulates a set of solutions. Simulated Annealing (SA) is a probabilistic technique inspired by the metallurgical process of annealing. This process entails heating and then controlled cooling of a material to enhance the size of its crystals and minimize defects. Similarly, simulated annealing explores the solution space by initially allowing a broad search that includes suboptimal solutions (reflecting high temperatures) and progressively narrowing the search to do in on the most promising solutions as the ”temperature” decreases. This method is particularly effective for finding approximate solutions in complex optimization scenarios, where traditional methods might fail to escape local optima.

Understanding the classifications of P, NP (non-deterministic polynomial-time), NP-complete, and NP-Hard problems is crucial in computational theory, as they categorize problems based on their complexity and solvability. This framework profoundly impacts various fields, including cryptography, optimization, and algorithm design, shaping the strategies researchers and practitioners employ in tackling complex problems. SA stands out as an effective technique for managing particularly challenging optimization problems, especially those classified as NP-hard. In computational complexity theory, NP-hardness denotes a category of problems known to be at least as difficult as the most challenging problems in NP (nondeterministic polynomial time). These problems are notorious for their absence of efficient (polynomial time) solutions, rendering exact solutions impractical as problem sizes increase. This context highlights the value of simulated annealing, as it provides a probabilistic method to approximate solutions in scenarios where traditional methods falter.

Finllay, we utilize Qiskit, an open-source quantum computing framework, to address the knapsack problem. We broaden the classical Knapsack problem to a two-dimensional (2D) lattice arrangement. We map the knapsack Hamiltonian into the Ising model and subsequently transform it into a Quadratic Unconstrained Binary Optimization (QUBO) format, using the converter provided by Qiskit [12]. We implement this because the quantum effects are described by the Ising type binary interactions and quantum computers support these quadratic formulations in Ising Hamiltonian. Therefore, we need to recast the inequality constraints into an inequality quadratic form by introducing slack variables which allows us to reach to the QUBO form. The goal is to encode the Knapsack problem into the ground state of the Hamiltonian and the state that we are looking is the one that minimizes the energy. Energy is the expectation value of the Hamiltonian that we are looking.

3
Literature Study

In this section, we provide a literature review and background knowledge to describe the context and the main concepts of this master thesis. This information serves as a foundation for the subsequent chapters and helps the reader to follow the arguments, methods, and results.

3.1 Ising formulations of many NP problems

In recent years, several studies have focused on solving combinatorial optimization. In the following, one can find an overview of the most relevant studies in this area. First, we start by studying [13] which provides strategies for mapping a wide variety of NP problems like partitioning problems, binary integer linear programming, and covering and packing problems to the Ising Model. These terms will be explained in chapter 6.
Spin glasses are disordered magnetic systems known for their complex energy landscapes filled with local minima, which make optimization challenging. We are interested in spin glasses design because with it we can design our problem as a phase change transition problem like as nature. In statistical mechanics, the phase change of a system can be considered with temperature. In other words, if there is a sudden change in magnetism at a certain temperature, we say there is a phase transition at that temperature. Considering our problem, changing our ferromagnets to air which is the main purpose of our project can be considered exactly as phase change from paramagnetism to ferrimagnetism in the Ising model. According to the [13] article, we can see how spin glass problems can be mapped to Ising models, where the spins in the model represent binary decisions in the optimization problem. The interactions between directions (representing constraints or relationships in the original problem) are formulated such that the energy minimization of the Ising model corresponds to finding an optimal solution to the original problem. Translating complex computational problems into quantum-friendly formats, is to enable the groundwork for leveraging quantum computation to tackle problems that are currently intractable for classical computers. For each problem, the author constructs an Ising Hamiltonian whose ground state corresponds to a

solution to the problem. The number of spins required and the interaction between them varies depending on the complexity of the original problem. The formulation of classical NP problems as Ising models is significant for quantum computing because it suggests that adiabatic quantum optimizers if realized practically, could potentially solve a wide range of computational problems more efficiently than classical algorithms. The paper begins by discussing the potential of adiabatic quantum optimization (AQO) to solve NP-complete and NP-hard problems. The process involves initializing a quantum system in the ground state of an easy-to-prepare Hamiltonian (H_0) and then adiabatically evolving it to the Hamiltonian encoding the problem of interest (H_P) which can be seen in equation 3.1.

$$H(t) = \left(1 - \frac{t}{T}\right) H_0 + \frac{t}{T} H_P \tag{3.1}$$

The ground state of H_P encodes the solution to the problem. The formulation of classical NP problems as Ising models is significant for quantum computing because it suggests that adiabatic quantum optimizers, if realized practically, could potentially solve a wide range of computational problems more efficiently than classical algorithms [13].

The limitations and challenges in embedding these Hamiltonians onto actual quantum devices are also discussed in this paper. The number of qubits, their connectivity, and the precision required in specifying interaction strengths are critical factors that determine the feasibility of solving specific problems on quantum hardware. Another important part of this article is packing problems which we can map from NP problems to the Ising model. These problems can often be thought of as asking: how one can pick elements out of a set (such as vertices out of a graph's vertex set) so that they "cover" the graph in some simple way (e.g., removing them makes the edge set empty) [13]. One of the most important examples of packing problems is the Knapsack problem. Knapsack is an optimization algorithm that involves selecting a subset of items from a given set, each with a specific weight and value, to maximize the total value while staying within a weight limit. The goal is to decide which items to include in a knapsack so that the total weight does not exceed the capacity of the knapsack, and the total value is maximized. The article also speculates on the future potential of AQO, discussing its limits and the possibility that while it might not always offer a faster solution than classical algorithms for all NP problems, it could still be beneficial for certain cases. Moreover, it is mentioned that other NP problems might be easier to solve than previously thought due to certain properties that could be exploited by AQO or other quantum computing methods. Finally, the paper concludes with acknowledgments and references, detailing contributions from other scholars and previous studies that have explored related concepts in quantum computing and optimization problems. This article serves as a critical resource for understanding how quantum approaches can be tailored to tackle some of the most challenging problems in computer science and physics.

3.2 Optimization by Simulated Annealing (SA)

A pivotal piece in the literature on Simulated Annealing, the article [14] by Scott Kirkpatrick titled 'Optimization by simulated annealing', published in 1983 in a science journal, provides foundational insights that are integral to this thesis. According to this paper, there is a deep connection between statistical mechanics and combinatorial optimization particularly in the concept of simulated annealing. The SA process consists of first "melting" the system being optimized at a high effective temperature, then lowering the temperature by slow stages until the sys-

tem ”freezes” and no further changes occur [14]. At each temperature, the simulation must proceed long enough for the system to reach a steady state. A fundamental question in simulated annealing concerns what happens to the system in the limit of low temperature. In the context of Boltzmann distribution, the ground state refers to the lowest energy state available to a system among all possible states. The Boltzmann distribution is directly related to temperature through the Boltzmann factor which is given by $e^{-\frac{E}{kT}}$ as T increase the exponential term decreases meaning that states with higher energies become less probable.
This article introduces annealing temperature T which is a controlling parameter in simulated annealing for optimization and shows how to carry out the simulated annealing progresses to obtain a better heuristic solution to combinatorial optimization. Spin glasses are disordered magnetic systems where the interactions between spins are random, leading to frustration. This randomness and the resulting frustration cause a highly irregular energy landscape with many local minima, making the system's behavior complex and difficult to predict. Annealing as implemented by the metropolis procedure differs from iterative implementation because annealing is not stuck in low temperatures and also frustration does not show up.
Design of the computer is also discussed in [14] which is related to the simulated annealing framework that arrives in the design of computer circuits. This process is usually divided to several steps [14]:

- Partitioning: partitioning must be done in such a way that the number of circuits in each partition is small enough to fit easily into the available packages.

- Placement: The major focus in placement is on minimizing the length of connection in order to finish the system.

- Wiring: in wiring, it is desirable to maintain the minimum possible length while minimizing the source of electromagnetic noise.

If we have connectivity information in a matrix a_{ij} which indicates the number of signals passing between circuit i and j, and if we indicate which circuit i is placed by a two variable $\mu_{ij} = \pm 1$ we can have our objective function. λ is expressed as the relative costs of imbalance and boundary condition.

$$f = \sum_{i>j} \left(\lambda - \frac{a_{ij}}{2}\right) \mu_i \mu_j \tag{3.2}$$

This objective function has the form of a Hamiltonian, and it can be shown that this Hamiltonian has a spin glass phase at low temperatures. We understand from [14] that freezing must be slow enough to reach the lowest possible energy. The metropolis algorithm proceeds in controlled steps from one configuration to the other that prevent our algorithm to get stuck in local minima.

3.3 Trafic signal optimization Example

Another study by [1] focuses on a square lattice by means of a quantum annealing machine, namely the D-Wave quantum annealer in order to solve the traffic problems. This article formulates the signal operation problem as a combinatorial optimization problem and the objective function of which is Hamiltonian. A simplified situation

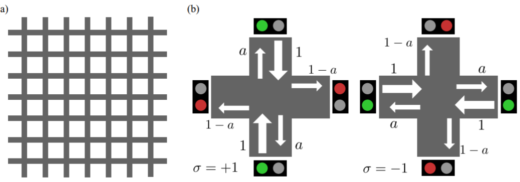


Figure 3.1: (a) Grid pattern of roads. (b) The two states of traffic signals at each intersection [1].

is considered in which two states are assumed for each signal: traffic is allowed in either the north-south direction or the east-west direction. The objective function related to this problem is [1] :

$$H(\sigma(t)) = \sigma(t)^T J_0(t) + h\sigma(t) + c(t) \tag{3.3}$$

which is a quadratic form with variables ±1, and it matches the Hamiltonian form of the Ising model [15]. Hence, solving the signal optimization problem of the objective function is regarded as equivalent to the problem of finding the spin direction $\sigma_i \in \{\pm 1\}$ that minimizes the Ising Hamiltonian.

According to this model which can be seen in Fig 3.1, at each node i, the signal can be in one of two states: σ_i = +1, allowing vehicle fow only in the north-south direction, or σi = −1, permitting vehicle fow only in the east-west direction. Traffic control is compared in three models of numerical experiments.

- Local control, which determines the signal display at each time step with a local rule. There is θ as a threshold, which is the value of the switching parameter and it is determined such that the objective function is minimized.

- Optimal control with simulated annealing: it reduces in equation 3.3 at each time step using the simulated annealing. The simulated annealing is an algorithm for finding a solution by examining the vicinity of the current solution at each step and probabilistically determining whether it should stay in the current state or switch to a vicinity state.

- Optimal control with quantum annealing: it reduces which can be seen in equation 3.3 by using the quantum annealing with the D-Wave 2000Q. Because the problem size exceeds the size of problems that 2000Q can solve, it is subdivided by the graph partitioning technique.

According to [1] Quantum Annealing performs the best among the three methods. This superiority is primarily due to its ability to achieve a smaller Hamiltonian value in steady state compared to the other methods, which indicates a more optimized solution. The Quantum Annealing method, employing the D-Wave 2000Q, demonstrated better synchronization and lower oscillation in control states, particularly effective in larger cities with traffic lights with a grid-like intersection layout.

The findings show that Quantum Annealing can more effectively handle the complex optimization problems associated with traffic signal control, providing superior performance, particularly in minimizing the objective function over large-scale networks [1]. The study also highlighted the potential benefits of newer quantum hardware which could further enhance performance, and suggested that Quantum Annealing is particularly advantageous when prioritizing reduction in signal switching frequency and achieving steady control states.

3.4 Quantum Bridge Analytics I: A Tutorial on Formulating and Using QUBO Models

The QUBO model has emerged as an underpinning of the quantum computing area known as quantum annealing and Fujitsu's digital annealing and has become a subject of study in neuromorphic computing [16]. Through these connections, QUBO models lie at the heart of experimentation carried out with quantum computers developed by D-Wave Systems and neuromorphic computers developed by IBM. The model expressed by the optimization problem:

$$QUBO : minimize/maximizey = \mathbf{x}^T \mathbf{Q} \mathbf{x}. \tag{3.4}$$

where x is a vector of binary decision variables and Q is a square matrix of constants [15]. It is common to assume that the Q matrix is symmetric or in upper triangular form, which can be achieved without loss of generality simply as follows:

Symmetric form: For all i and j except $i = j$, replace q_{ij} by $(q_{ij} + q_{ji})/2$.

Upper triangular form For all i and j with $j > i$, replace q_{ij} by $q_{ij} + q_{ji}$. Then replace all q_{ji} for $j < i$ by 0. (If the matrix is already symmetric, this just doubles the q_{ij} values above the main diagonal, and then sets all values below the main diagonal to 0).

In our project we have many elements that come together and cooperate with frequencies in the conductor. Therefore, each of these sites interacts according to its configuration if it has a ferrite or not. So binary relations are paying the way for the overall ability to surpass certain frequencies. Now, our aim is to balance the binary cooperation while reducing the number of sites, reminiscent of paramagnetic to ferrimagnetic transform for the Ising materials

4

Application to passive filters

4.1 Introduction to Filters

A filter is a circuit or a component capable of passing (or amplifying) certain frequencies while attenuating other frequencies. Thus, a filter can extract important frequencies from signals that also contain undesirable or irrelevant frequencies. Electromagnetic (EMI) filters are essential for power conversion systems interfaced with the utility grid since they suppress conducted noise and aid in complying with regulatory conducted emissions (CE) standards such as FCC Part 15 or CISPR 11/32, etc. Common mode (CM) filters are an integral part of EMI filters and play a key role since they are used for mitigating the CM noise generated by the fast switching action of semiconductors used in power converters [17].

4.1.1 Passive and Active Filters

Filters can be placed in one of two categories [2]:

Passive filters play an integral role in harmonic elimination in power systems. These filters operate on the basic principles of electrical circuit theory, harnessing the characteristic behaviors of resistors, inductors, and capacitors to selectively eliminate unwanted frequencies. A passive filter is a combination of resistors, capacitors, and inductors arranged in various configurations to achieve a desired filtering effect [2]. The main function of a passive filter is to allow certain frequencies to pass while blocking or attenuating others. This is primarily achieved by exploiting the frequency-dependent impedance characteristics of capacitors and inductors. Passive filters are most responsive to a frequency range from roughly 100 Hz to 300 MHz. The limitation on the lower end results from the fact that the inductance or capacitance would have to be quite large at low frequencies. The upper-frequency limit is due to the effect of parasitic capacitances and inductances. Careful design practices can extend the use of passive circuits well into the gigahertz range [2].

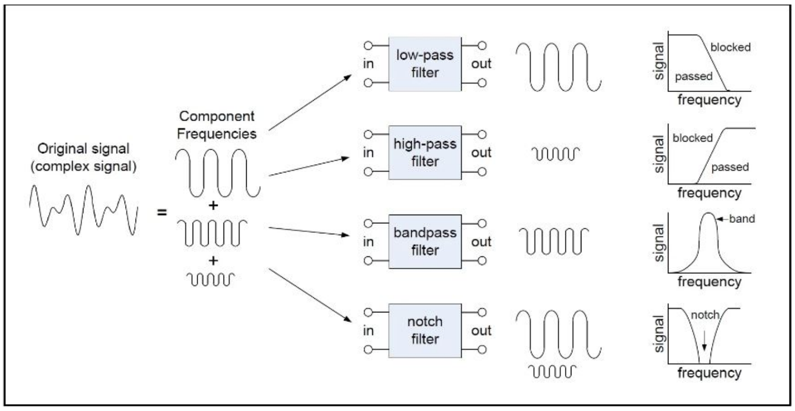


Figure 4.1: A basic depiction of the four major filter types [2].

Active filters are capable of dealing with very low frequencies (approaching 0 Hz), and they can provide voltage gain (passive filters cannot). Active filters can be used to design high-order filters without the use of inductors; this is important because inductors are problematic in the context of integrated-circuit manufacturing techniques. However, active filters are less suitable for very high-frequency applications because of amplifier bandwidth limitations. Radio-frequency circuits must often utilize passive filters [2].

There are four types of filters which are and can be seen in Fig 4.1:

- Low-Pass Filter (LPF): Allows frequencies below a certain cutoff frequency to pass through while attenuating frequencies above the cutoff [2].
- High-Pass Filter (HPF): Allows frequencies above a certain cutoff frequency to pass through while attenuating frequencies below the cutoff [2].
- Band-Pass Filter (BPF): Allows frequencies within a certain range (band) to pass through while attenuating frequencies outside this range [2].
- Band-Stop Filter (BSF) or Notch Filter: Attenuates frequencies within a certain range while allowing frequencies outside this range to pass through [2].

5
Finite Element Method

The finite element method (FEM) is a numerical method for solving partial differential equations arising in engineering [18]. The FEM constitutes a general tool for the numerical solution of partial differential equations in engineering and applied science. Historically, all major practical advances of the method have taken place since the early 1950s in conjunction with the development of digital computers. However, interest in approximate solutions of field equations dates as far back in time as the development of the classical field theories (e.g. elasticity, electro-magnetism) themselves [18].
To solve a problem, the FEM subdivides a large system into smaller, simpler parts called (finite) elements. This is achieved by a particular space discretization in the space dimensions, which is implemented by the construction of a mesh of the object. We aim to perform an optimization on a continuous domain, but solving it directly is infeasible. By linearizing the problem, we can achieve solvable solutions. The closer our mesh approximation is to the actual car components, the more reliable our results will be. However, the computations with refined mesh will lead to a temporal overhead. While discrete solutions offer a practical approach, our problem fundamentally lies in a continuous domain. Therefore, we need to express our problem using integral equations over the domain. So we discretized everything using mesh because we propagate the solution linearly or quadratically, depending on the mesh element type chosen.

5.1 Maxwell's equations

Detailed solutions for magnetic fields in most situations of practical engineering interest involves the solution of Maxwell's equations along with various constitutive relationships which describe material properties. Maxwell's equations illustrate, with succinct beauty, the unique coexistence in nature of the electric field and the magnetic field [19].

The Maxwell equations are given by [20] :

$$\nabla \cdot \mathbf{D} = \rho \qquad \text{(Gauss's law for electric field)} \tag{5.1}$$

$$\nabla \cdot \mathbf{B} = 0 \qquad \text{(Gauss's law for magnetic field)} \tag{5.2}$$

$$\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t} \qquad \text{(Faraday's law of induction)} \tag{5.3}$$

$$\nabla \times \mathbf{H} = \mathbf{J} + \frac{\partial \mathbf{D}}{\partial t} \qquad \text{(Ampere's circuital law)} \tag{5.4}$$

where Maxwell defined **D** as the displacement current vector, **B** as the magnetic flux density, ρ as the volume charge density (in C/m^3), **J** as the electric current density (in A/m^2), and ∇ as the nabla (i.e., del) operator (i.e., the first-order differential operator).

The electric flux always comes from the positive charge and passes to a negative charge. This is expressed mathematically by the divergence equation 5.1 and known as Gauss's law for electric fields. The magnetic flux is the magnetic field crossing an area and is always circulating with no starting and end point. The characteristic of magnetic flux is described by the divergence equation 5.2 of the magnetic flux density and is known as Gauss's law for magnetic fields. The continuity equation of the electric current describes the current flowing from the observation point reduces the charge of the point and can be calculated as [9]:

$$\nabla \cdot \mathbf{J} = -\frac{\partial \rho}{\partial t} \tag{5.5}$$

According to Faraday's law of induction, any change in the magnetic field induces electromotive force. The Faradays' law of induction states that change of magnetic flux crossing an open surface S is equivalent to a negative line integral of the electric field strength along the line l around the surface and can be calculated as [9]:

$$\oint \mathbf{E} \cdot d\mathbf{l} = -\frac{d}{dt} \int \mathbf{B} \cdot d\mathbf{S} = -\frac{d\varphi}{dt} \tag{5.6}$$

The association of Faraday's law and Ampere's law is essential as these laws determine the induced voltage in the coil (windings) of the electrical machines. Faraday's and Ampere's law is also crucial in determining losses caused by eddy currents in the magnetic circuit [21].

5.1.1 Eddy-currents

In electromagnetism, an eddy current (also called Foucault's current) is a loop of electric current induced within conductors by a changing magnetic field in the conductor according to Faraday's law of induction or by the relative motion of a conductor in a magnetic field [22]. Eddy currents flow in closed loops within conductors, in planes perpendicular to the magnetic field. They can be induced within nearby stationary conductors by a time-varying magnetic field created by an AC electromagnet or transformer, or by relative motion between a magnet and a nearby conductor. The magnitude of the current in a given loop is proportional to the strength of the magnetic field, the area of the loop, and the rate of change of flux, and inversely proportional to the resistivity of the material. When graphed, these circular currents within a piece of metal look vaguely like eddies or whirlpools in a liquid [23].

By Lenz's law, an eddy current creates a magnetic field that opposes the change in the magnetic field that created it, and thus eddy currents react back on the source of the magnetic field. For example, a nearby conductive surface will exert a drag force on a moving magnet that opposes its motion, due to eddy currents induced in the surface by the moving magnetic field. This effect is employed in eddy current brakes which are used to stop rotating power tools quickly when they are turned off. The current flowing through the resistance of the conductor also dissipates energy as heat in the material [24].

5.1.2 Flux linkage

In electrical engineering, the term flux linkage is used to define the interaction of a multi-turn inductor with the magnetic flux as described by Faraday's law of induction [25]. Since the contributions of all turn in the coil add up, in the over-simplified situation of the same flux Φ passing through all the turns, the flux linkage (also known as flux linked) is [25] :

$$\lambda = n\Phi \tag{5.7}$$

In a typical application, the term "flux linkage" is used when the flux is created by the electric current flowing through the coil itself. Per Hopkinson's law,

$$\lambda = n\frac{\mathrm{MMF}}{R} \tag{5.8}$$

where MMF is the magnetomotive force and R is the total reluctance of the coil. Since

$$\mathrm{MMF} = nI \tag{5.9}$$

where I is the current, the equation can be rewritten as

$$\lambda = LI \tag{5.10}$$

where L is called the inductance [26].

So in this thesis, by having λ and current we can get the L for each body.

5.1.3 Magnetic coenergy

In physics and engineering, Coenergy (or co-energy) is a non-physical quantity, measured in energy units, used in theoretical analysis of energy in physical systems [3]. Consider a system with a single coil and a non-moving

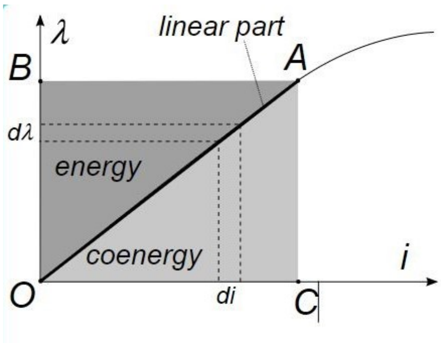


Figure 5.1: Graphical definition of coenergy [3]

armature (i.e., no mechanical work is done). Hence, all of the electric energy supplied to the device is stored in the magnetic field [3].

$$dW_{\text{input}} = dW_{\text{stored}} \quad (dW_{\text{mechanical}} = 0) \tag{5.11}$$

where (e is the voltage, i is the current, and λ is the flux linkage):

$$dW_{\text{input}} = eidt \tag{5.12}$$

$$dW_{\text{stored}} = id\lambda \tag{5.13}$$

For calculations either the flux linkage λ or the current i can be used as the independent variable.

The total energy stored in the system is equal to the area OABO in Fig 5.1, which is in turn equal to OACO, therefore [3]:

$$\text{Energy} = \text{Area}(OABO) = W_{\text{stored}} = \int_0^{\lambda} i(\lambda)d\lambda \tag{5.14}$$

$$\text{Coenergy} = \text{Area}(OACO) = W'_{\text{stored}} = \int_0^{i} \lambda(i)\, di \tag{5.15}$$

For linear lossless systems, the coenergy is equal in value to the stored energy. the coenergy has no real physical meaning, but it is useful in calculating mechanical forces in electromagnetic systems. The self inductance is defined as flux linkage over current:

$$L = \frac{\lambda}{i} \tag{5.16}$$

and the energy stored in a coil is:

$$W_{\text{energy}} = \frac{1}{2}\frac{\lambda^2}{L} = \frac{1}{2}Li^2 \tag{5.17}$$

In a magnetic circuit with a movable armature the inductance L(x) will be a function of position x.

It can be therefore written that the field energy is a function of two mathematically independent variables λ

and x:

$$W(\lambda, x)_{\text{energy}} = \frac{1}{2}\frac{\lambda^2}{L(x)} \tag{5.18}$$

$$W'(i, x)_{\text{coenergy}} = \frac{1}{2}L(x)i^2 \tag{5.19}$$

The last two expressions are general equations for energy and coenergy in magnetostatic system.
Elmer computations report the magnetic co-energy as lumped parameter at each time step. This value used to quantity L.

5.1.4 Inductance

Inductance is the tendency of an electrical conductor to oppose a change in the electric current flowing through it [9]. The electric current produces a magnetic field around the conductor. The magnetic field strength depends on the magnitude of the electric current and follows any changes in the magnitude of the current. From Faraday's law of induction, any change in magnetic field through a circuit induces an electromotive force (EMF) (voltage) in the conductors, a process known as electromagnetic induction. This induced voltage created by the changing current has the effect of opposing the change in current. This is stated by Lenz's law [9].
In our project inductances (L) is computed from FEM and fed to the objective function which is explained in equation 8.1. All conductors have some inductance, which may have either desirable or detrimental effects in practical electrical devices. The inductance of a circuit depends on the geometry of the current path, and on the magnetic permeability of nearby materials; ferromagnetic materials with a higher permeability like iron near a conductor tend to increase the magnetic field and inductance. Any alteration to a circuit that increases the flux (total magnetic field) through the circuit produced by a given current increases the inductance, because inductance is also equal to the ratio of magnetic flux to current which can be indicated in eq 5.20 [27] [28] [29] [30].

$$L = \frac{\Phi(i)}{i} \tag{5.20}$$

5.2 Elmer software

In our simulation work using Elmer, a critical aspect involves the meticulous definition of boundary conditions to accurately model the physical interactions at different locations of our component. For instance, boundary condition 2 is specified to model a current density at a particular boundary, essential for studying electromagnetic effects. Meanwhile, boundary condition 3, known as `Adiabatic_boundary`, ensures no heat transfer across the boundary by setting a zero heat transfer coefficient, simulating an insulated condition. Additionally, Boundary Condition 4, labeled `Heat_transfer_environment`, is defined to simulate heat exchange with an external environment by assigning a specific heat transfer coefficient. These conditions allow us to replicate and analyze real-world physical behaviors accurately, leveraging Elmer's capabilities to handle complex multi physical interactions efficiently.

Elmer software consists of different parts which are :

- ElmerGUI : ElmerGUI is the graphical user interface for Elmer and it serves as a user-friendly front end that facilitates the setup, execution, and post-processing of simulations within the Elmer software suite. It allows users to create simulation models by defining the geometry, selecting material properties, setting boundary conditions, and configuring solver settings through a graphical interface, rather than requiring manual edits of text files or script-based setups [31]. In our project, we use this tool to present the grid domains.
- ElmerGrid: ElmerGrid is a simple mesh generator and mesh manipulation utility. It is an appropriate tool for generating structured meshes for simple 1D, 2D, and 3D geometries. It can also read meshes generated by other mesh generators and manipulate them. Among the possible operations are scaling, changing of element type, defining boundaries, or partitioning the mesh for parallel solution, for example. ElmerGrid was originally a side product of research done in the area of silicon carbide crystal growth. The main goal of ElmerGrid was to write a simple mesh generator for multiphysical problems where different meshes based on the same geometry were needed for different physical problems. Since that the software has been modified to meet the needs of Elmer's development [32].
- ElmerSolver: The numerical solver that performs the finite element calculations, using the mesh and case files [33].
- ElmerPost: We use Paraview as a postprocessor for the visualization module [34].

As it can be seen in Fig 5.2 we are able to show our Grid domain with bodies using ElmerGui. Our mesh is consisted of 32 bodies each with the value 1 representing the ferrite. It is worth to mention that the body in the middle is Cooper and the domain around the boundary is air. We can see from the Fig 5.3 that each body is unique with their labels as Fig 5.4. Visualizing the unique bodies is going to help us to understand the phase change which is the main goal our project. The phase change from ferrite to air, leading to lighter CM filter designs..

5.3 Elmer FEM Solver

In this thesis, we utilize Elmer, an open-source finite element software initially developed through collaboration among Finnish universities, research institutes, and industry. Elmer facilitates the solving of multi-physical problems including fluid dynamics, structural mechanics, electromagnetic, heat transfer, and acoustics [35]. These are modeled by partial differential equations solved via the FEM.

Solving partial differential equation (PDE) models with the solver of Elmer requires that a precise description of the problem is given using the so-called solver input file, briefly referred to as the `sif.file`. This file contains user-prepared input data that specify the location of mesh files and control the selection of physical models, material parameters, boundary conditions, initial conditions, stopping tolerances for iterative solvers, etc.

5.3.1 The solver input file

The material of the solver input file is organized into different sections. Each section is generally started with a row containing the name of the section, followed by a number of keyword commands. The names of the sections that we have in our template file are :

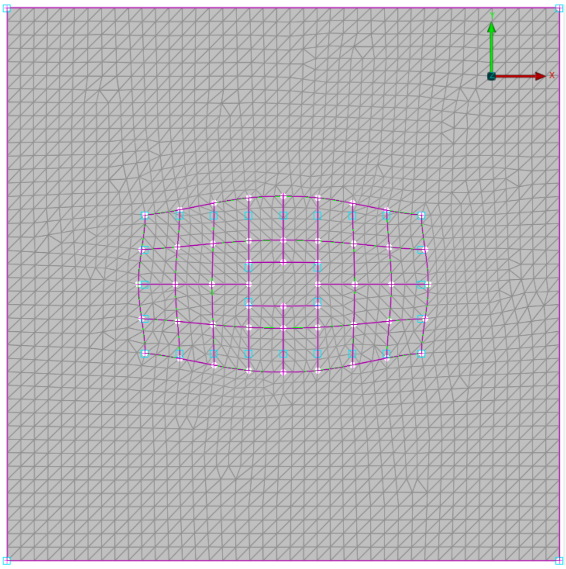


Figure 5.2: Grid domain with bodies, as it can be seen we have 32 bodies with conductor in the middle. The domain around the bodies is air.

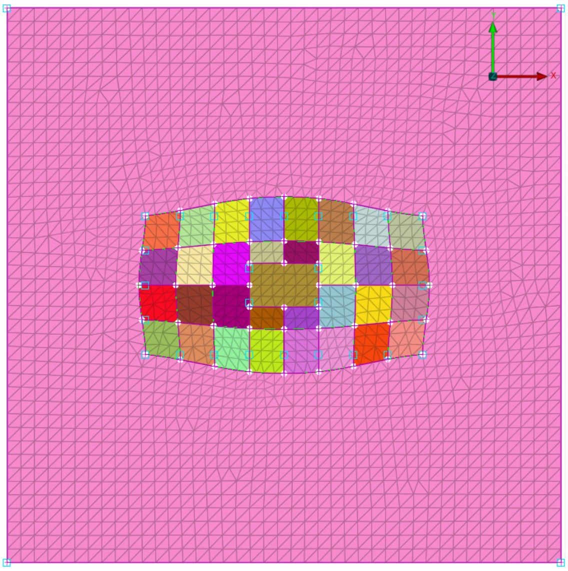
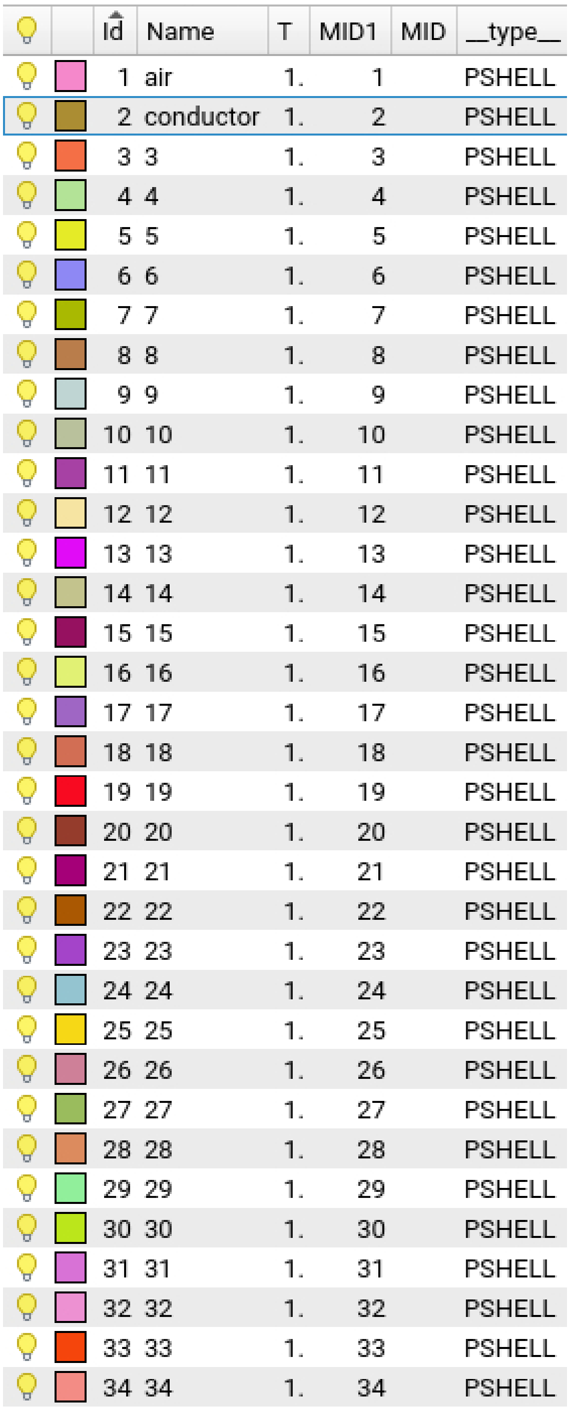


Figure 5.3: Grid domain with bodies. It indicates a finite element mesh generated for a computational simulation. Within this grid, there are 32 distinct bodies, each uniquely colored to differentiate them from one another



💡		Id	Name	T	MID1	MID	__type__
💡	🟪	1	air	1.	1		PSHELL
💡	🟫	2	conductor	1.	2		PSHELL
💡	🟧	3	3	1.	3		PSHELL
💡	🟩	4	4	1.	4		PSHELL
💡	🟨	5	5	1.	5		PSHELL
💡	🟦	6	6	1.	6		PSHELL
💡	🟩	7	7	1.	7		PSHELL
💡	🟫	8	8	1.	8		PSHELL
💡	⬜	9	9	1.	9		PSHELL
💡	🟩	10	10	1.	10		PSHELL
💡	🟪	11	11	1.	11		PSHELL
💡	🟨	12	12	1.	12		PSHELL
💡	🟪	13	13	1.	13		PSHELL
💡	🟨	14	14	1.	14		PSHELL
💡	🟥	15	15	1.	15		PSHELL
💡	🟨	16	16	1.	16		PSHELL
💡	🟪	17	17	1.	17		PSHELL
💡	🟧	18	18	1.	18		PSHELL
💡	🟥	19	19	1.	19		PSHELL
💡	🟫	20	20	1.	20		PSHELL
💡	🟪	21	21	1.	21		PSHELL
💡	🟫	22	22	1.	22		PSHELL
💡	🟪	23	23	1.	23		PSHELL
💡	🟦	24	24	1.	24		PSHELL
💡	🟨	25	25	1.	25		PSHELL
💡	🟥	26	26	1.	26		PSHELL
💡	🟩	27	27	1.	27		PSHELL
💡	🟧	28	28	1.	28		PSHELL
💡	🟩	29	29	1.	29		PSHELL
💡	🟩	30	30	1.	30		PSHELL
💡	🟪	31	31	1.	31		PSHELL
💡	🟪	32	32	1.	32		PSHELL
💡	🟧	33	33	1.	33		PSHELL
💡	🟥	34	34	1.	34		PSHELL

Figure 5.4: The label of 32 bodies. According to these labels we can understand which body is going to experience the phase change from ferrite to air. It is worth to mention that Mesh manipulation software ANSA is used for visualization [4]

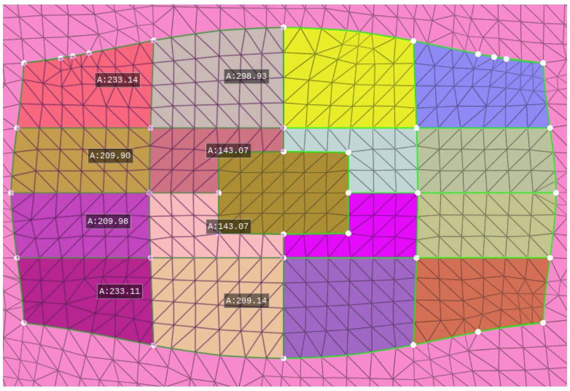


Figure 5.5: The areas of each domain in mm^2, which is a way to show the weight of each domain.

Header

We locate the location of mesh files in the header section

Simulation

The simulation section is used for giving general information that is not specific to a particular PDE model involved in the simulation. This information describes the coordinate system used, indicates whether the problem is stationary or evolutionary, defines the file names for output, etc. We've configured several key aspects to ensure the simulation accurately reflects the dynamic behavior of the system over time.
The simulation is conducted within a quasi-3D filter with 1 meter of depth by using symmetric boundary conditions. The conventional quasi-3D analysis is a method that can equalize a 3D analysis with multiple 2D analyses and is widely used for 3D models that take a long analysis time. We utilize a transient simulation type, which allows us to observe how variables evolve, rather than assuming steady conditions. To achieve this, we employ a timestepping method using the second-order Backward Differentiation Formula (BDF), known for its stability and effectiveness in solving stiff equations. The simulation parameters are set with time steps of 0.1 units and run for several intervals to capture the necessary data, all defined in our solver input file.

Constants

The constants section is used for defining certain physical constants. Like Boltzman, permeability of vacuum.

Body, Material, Body Force, and Initial Conditions

The Elmer mesh files contain information on how the computational region is divided into parts referred to as bodies. A body section associates each body with an equation set, material properties, body forces, and initial conditions by referring to definitions given in a specified equation section, material section, body force section, and initial condition section. To do this, the different sections of the same type are distinguished by integer identifiers that are parts of the section names. It is worth mentioning that we have 3 different materials and their physical properties are used in simulations. The first material considered in our simulation is air at room temperature, pivotal for fluid dynamics and thermal transfer analyses. Its dynamic viscosity of 1.983e-5 Pa·s reflects air's internal friction, crucial for predicting flow behaviors in fluid dynamics simulations. Other essential properties include a heat capacity of 1005.0 J/kg·K, the sound speed of 343.0 m/s, and air's heat expansion coefficient of 3.43e-3 per Kelvin which are key for modeling thermal expansion, Additionally, air's density of 1.205 kg/m^3 affects buoyancy and pressure calculations, and its thermal conductivity is considered as 0.0257 W/m·K. These properties are fundamental for accurately simulating air's behavior under various conditions and ensuring the reliability of our simulation outcomes. Following air, the second material explored in our study is a specific type of ferrite, designated ferromagnetic. This material is crucial as we aim to optimize and minimize the number of meshes required in our simulations. The third material in our study is copper, serving as the conductor centrally located within the component previously described. We have carefully defined its properties to ensure accurate simulation and analysis within our model. Copper's selection is crucial due to its excellent conductive attributes, both thermal and electrical, which are integral to the component's performance.

Equation and solver sections

Equation section provides us a way to associate each body with a set of solvers, where each solver is typically associated with the ability to solve a certain physical model. This section provides a way to associate each body with a set of solvers, where each solver is typically associated with the ability to solve a certain physical model. We have 6 solvers in our file.:

- **Solver 1: MagnetoDynamics2D**
  The first solver is designed to solve for the magnetic vector potential **A** in a 2D magnetodynamics problem. The magnetic vector potential **A** is used in electromagnetism to describe the magnetic field **B**. The equation being solved is typically derived from Maxwell's equations, focusing on the magnetostatic or magnetodynamic behavior in two dimensions. This solver employs an iterative method (BiCGStab) for solving the linear system with appropriate preconditioning. Key parameters include nonlinear system convergence criteria, maximum iterations, and details of the linear system solver.

- **Solver 2: MagnetoDynamicsCalcFields**
  Purpose: Calculating various electromagnetic field quantities based on the magnetic vector potential solved by Solver 1.
  Theory: This solver computes derived fields such as magnetic field strength **H**, magnetic flux density **B**, electric field **E**, current density **J**, and more. These calculations are crucial for understanding the detailed electromagnetic behavior of the system, including Joule heating, nodal forces, and Maxwell stress.

- **Solver 3: Heat Equation**
  Purpose: Solving the heat equation to model thermal effects in the material.
  Theory: The heat equation, in the form

$$\rho c_p \frac{\partial T}{\partial t} = \nabla \cdot (k \nabla T) + Q$$

describes the distribution of temperature T in a material over time, where ρ is density, c_p is specific heat, k is thermal conductivity, and Q is the heat source term. This solver models the thermal response of the material, taking into account thermal conductivity and potential heat generation from electromagnetic effects (e.g., Joule heating).

- **Solver 4: Result Output**
  Purpose: Outputting the results of the simulation in a specified format.
  Theory: This solver formats and writes the computed simulation results to a file, typically used for post-processing and visualization. The VTU format (VTK unstructured grid) is commonly used for representing mesh-based data, allowing detailed visualization of the simulation results.

- **Solver 5: SaveScalars**
  Purpose:Saving scalar quantities for post-processing or further analysis.
  Theory: This solver extracts specific scalar quantities such as time, CPU time, current density, and magnetic flux density. These scalar values can be used for analyzing the performance and behavior of the system over time or at specific locations.
  Key Parameters:Variables to save, operators for calculating averages or integrals, and masks for selecting specific regions or bodies.

- **Solver 6: Density** purpose: The captured data is essential for post-processing activities, where detailed analysis or optimization of materials based on their responses in various conditions is conducted. Theory: This parameter is highlighted as a key variable, indicating that the density of materials is specifically recorded by this solver. We have 3 material and with 3 different densities which are air with the density of 1.205, ferrite with the density of 7925.0, and Copper with 8960.0. The aim is to visualize 3 materials distinctly on an image.

Boundary Condition

Boundary condition sections define the boundary conditions for the different models. The Elmer mesh files contain information on how the boundaries of the bodies are divided into parts distinguished by their own boundary numbers [36]. The setup provides insights into the complexity and multidisciplinary nature of the simulations that can be handled within a FEM framework, where electrical, thermal, and other physical phenomena can be independently or jointly modeled depending on the requirements of the study or design process. The boundary condition in our case actively enforces an electrical grounding (zero potential) at boundary 1.

6
Optimization methods

6.1 Optimization

In mathematics and computer science, an optimization problem is the problem of finding the best solution from all feasible solutions. In the simplest terms, an optimization problem consists of maximizing or minimizing a real function by systematically choosing input values from within an allowed set and computing the value of the function.

Optimization problems can be divided into two categories, depending on whether the variables are continuous or discrete [37]:

- An optimization problem with discrete variables is known as a discrete optimization, in which an object such as an integer, permutation or graph must be found from a countable set.

- A problem with continuous variables is known as a continuous optimization, in which an optimal value from a continuous function must be found. They can include constrained problems and multimodal problems.

We would like to use passive filter design as a combinatorial optimization problem which is a component functioning in the vehicle as a Low Pass Filter (100MHz). Looping the optimization candidates designed by the different algorithms with FEM, we will be able to see how the resulting filter performs towards human-designed alternatives.

6.2 Np-hardness

In computational complexity theory, a complexity class is a set of computational problems "of related resource-based complexity" [38]. The two most commonly analyzed resources are time and memory. Therefore, understanding the complexity class of a problem is a crucial aspect of every research. This knowledge enables researchers to select the most appropriate algorithm, effectively manage time, and achieve more accurate results. By classifying a problem's complexity, we can tailor our computational approaches to be both efficient and effective, thereby enhancing the quality and reliability of the outcomes. The following complexity classes are relevant for the optimization problems considered here:

- **P**: complexity class of problems that can be solved by a deterministic Turing machine in polynomial time. Following Cobham-Edmonds thesis [39], problems belonging to this class are in practice efficiently solvable. P contains many important nontrivial problems, including the decision versions of linear programming, of the greatest common divisor problem, and of finding a maximum matching in a graph, as well as deciding if an integer is prime [38].
- **NP** : complexity class of problems that can be solved by a non-deterministic Turing machine in polynomial time.
- **NP-hard** : complexity class of problems to which every problem in NP can be reduced in polynomial time. The resulting problems can also be in NP, but not necessarily. Informally these problems can be viewed as the hardest problems of NP and problems that are even harder. This class contains many problems, including Traveling Sales Person and other routing problems, finding a minimum vertex cover in a graph, or the graph coloring problem. There are even NP-hard problems that are not decidable, for example the Halting problem [38] [40].
- **NP-complete**: complexity class of problems that are NP-hard and in NP. This class is very important since every NP-complete problem represents the whole class in terms of complexity characterization, i.e. general findings on a single NP-complete problem can be applied to every NP-complete problem. A number of NP-complete problems are known [39], including the boolean satisfiability problem, the knapsack problem, or the decision version of TSP.

Fig 6.1 depicts the correlation of the described complexity classes under the assumption that P and NP are not equivalent. Since it has not yet been proven that the complexity class P is not equivalent to the complexity class NP, it cannot be excluded that P = NP. The consequences would be enormous for practical computer science, due to the fact that the equivalence of P and NP implies the equivalence of the class of NP-complete problems and P. Therefore algorithms would exist that solve every NP-complete problem in polynomial time. This would be of great value for the solution of many optimization problems, and a tremendous threat for cryptography. Nevertheless, strong evidence exists that $P \neq NP$, since nobody has found an algorithm yet that solves any of the 3000 known NP-complete problems in polynomial time. Therefore it is reasonable as well as practical to research the application of heuristics or metaheuristic algorithms to NP-complete and N P-hard problems.

Our project involves optimizing the configuration of ferromagnetic bodies on a grid to minimize energy. This problem is classified as NP-hard due to the combinatorial nature of the task. With 32 ferrite bodies, there are 32!

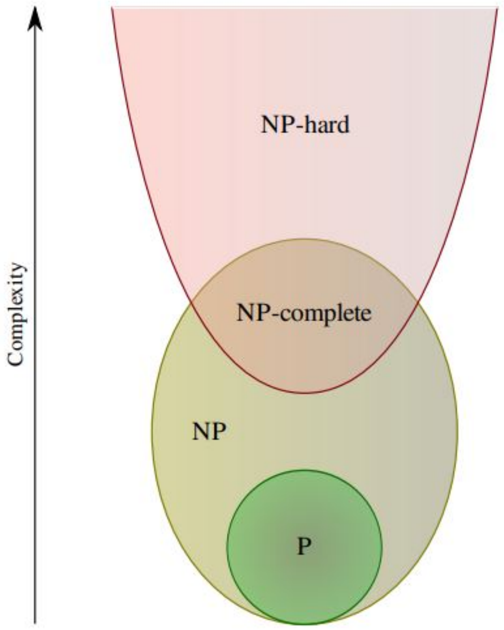


Figure 6.1: Euler diagram of problem complexity classes, under the assumption that $P \neq NP$ [5]

(approximately 2.63×10^{35}) feasible solutions. Given the vast number of possible configurations, this optimization problem is characterized by numerous local minima, which we can see in the parts of the results, making the identification of the global minimum a computationally intensive task.

6.3 Statistical mechanics

Statistical mechanics is a branch of physics that uses statistical methods to explain and predict the behavior of systems with a large number of particles. It bridges the microscopic laws of physics, which govern individual particles, and the macroscopic properties of materials that we observe, such as temperature, pressure, and volume [41]. Statistical mechanics contains many useful tricks for extracting properties of a macroscopic system from microscopic averages. The partition function Z is a central concept in this context which can be seen in equation 6.1. In this equation, β is the thermodynamic beta, defined as $\frac{1}{k_B T}$, where k_B is the Boltzmann constant, and E_i is the total energy of the system in the respective microstate.

In statistics and statistical physics, the Metropolis-Hastings algorithm is a Markov chain Monte Carlo (MCMC) method for obtaining a sequence of random samples from a probability distribution from which direct sampling is difficult [42]. New samples are added to the sequence in two steps: first a new sample is proposed based on the previous sample, then the proposed sample is either added to the sequence or rejected depending on the value of the probability distribution at that point. This algorithm is an essential tool in computational statistical mechanics and allows for the efficient sampling of the Boltzmann distribution. It generates a sequence of states for the system, where each state is generated with a probability P proportional to $e^{-\beta E_i}$. The algorithm proceeds as follows:

- Start with an initial state s with energy $E(s)$.
- Propose a new state s' with energy $E(s')$.
- Calculate the acceptance probability $A = \min\left(1, e^{-\beta(E(s')-E(s))}\right)$.
- Generate a random number r uniformly distributed between 0 and 1.
- If $r \leq A$, accept the new state s'; otherwise, retain the current state s.

By iterating this process, the Metropolis algorithm ensures that the states are sampled according to the Boltzmann distribution, allowing the computation of thermodynamic averages and other properties [41].

$$Z = \sum_i e^{-\beta E_i} \tag{6.1}$$

6.4 Ising Model

To address combinatorial optimization problems with a quantum algorithm, there is a way that is accomplished by expressing the optimization problem in terms of quantum variables. In many scientific situations, models are essential tools for understanding the world around us. The universe's inflation, general circulation models of the earth's climate, the double-helix model of DNA, evolutionary models in biology, agent-based models in social science, and equilibrium models of markets are just a few of the many phenomena they have documented [43]. Models of nature that can encompass a wide range of completely different systems are the most fruitful ones, and understanding how these models work leads to understanding all the physical systems the model can represent. The Ising model is one of the most often used models in physics which is a model consisting of discrete variables that represent magnetic dipole moments of atomic "spins" that can be in one of two states (+1 or −1). The Ising Model is a mathematical model; however, it can be considered as a model for a magnet. Mathematically, the fact that a problem is NP-complete means we can find a mapping to the decision form of the Ising model with a polynomial number of steps. This mapping can be re-interpreted as a pseudo-Boolean optimization problem [43]. The function that represents the energy of every single possible configuration (microstate) of the spins in the magnet is called the Hamiltonian which is shown as [44] :

$$\mathcal{H} = -J_{ij} \sum_{\langle i,j \rangle} S_i S_j - h_i \sum_i S_i \tag{6.2}$$

In the Ising model, the Hamiltonian includes two types of interactions:

- the **external field** term. The energies of the "spin down" and "spin up" states can be divided by an external magnetic field h, making one state more energetic than the other. The size of h represents how strong the field is, and the sign of which is describing whether it's spin up or spin down that's preferred. Since every individual spin feels the external field, it needs to sum over all sites to find the total contribution to the energy.

- the **interaction** term between neighboring spins. The reason for this interaction is that every spin in the magnet functions as a tiny magnetic dipole, creating a magnetic field that is felt by its neighbors. The size of J shows how strongly neighboring spins are coupled to each other, and the sign of it represents whether neighbors prefer to align or to anti-align [44].
  - $J_{ij} < 0$ is usually called the ferromagnetic coupling because the classical spins prefer to align with each other.
  - $J_{ij} > 0$ is usually called the anti-ferromagnetic coupling be cause the spins prefer to anti-align with each other.

We are using the Ising model since we want to minimize the weight of our ferrites by knowing the interactions between them. For a quantum form, we use spin operators σ_z^j, giving an Ising Hamiltonian, whose eigenvalues correspond to the previous cost. What we try to achieve for minimization, is to find the state minimizing the Ising model, Which corresponds to finding the minimal eigenvalue or ground state of the Hamiltonian. And this is what research/applications in combinatorial optimization using quantum computers are focusing on.

6.4.1 Phase change in Ising model

The Ising model of a ferromagnet is one of the simplest models displaying the paramagnetic ferromagnetic phase transition, that is, the spontaneous emergence of magnetization in zero external fields as the temperature is lowered below a certain critical temperature [45]. This model undergoes a phase transition between an ordered and a disordered phase in 2 dimensions or more. This was first proven by Rudolf Peierls in 1936, using what is now called a Peierls argument. The same phase change occurs in our project. In the end, Ising's model proved to be effective in simulating a variety of distinct physical systems. Any system composed of several independent components that communicate with one another pairwise can be described using this paradigm.

- High Temperature (Paramagnetic Phase): At high temperatures, thermal fluctuations dominate, and the spins are randomly oriented. The net magnetization M (the average spin) is close to zero. This disordered state is called the paramagnetic phase.

- Low Temperature (Ferromagnetic Phase): At low temperatures, the interaction energy dominates, and the spins tend to align either all up or all down, resulting in a non-zero net magnetization M. This ordered state is called the ferromagnetic phase.

- Critical Temperature Tc: The phase transition between the paramagnetic and ferromagnetic phases occurs at a specific temperature called the critical temperature Tc at T=Tc , the system undergoes a second-order phase transition.

6.5 Objective function formulation for Ising type problems

In order to navigate the intricacies of our algorithms for ferrite design, we must first establish a clear objective function. The objective function is prominently used to represent and solve the optimization problems of linear programming. Assuming the form Z = ax + by, where x and y are the decision variables. This function serves as the metric for quantifying the weight of the component based on the arrangement of mesh points (ie. distribution of ferromagnetic material). The primary aim of quantum algorithms is to minimize this objective function, a pivotal element in optimization problems representing what we want to minimize or maximize. Drawing an analogy to a common problem like the traveling salesman dilemma, where the objective function could signify the total distance traveled, our focus is on minimizing it. In the case of passive filter design, we adopt a straightforward model. Here, each mesh point contributes a fixed weight when set to 1 and no weight when set to 0. The overarching objective is to minimize the total weight, all while potentially adhering to specific constraints. In the realm of QC and optimization, it's intriguing to note that the objective function and the Hamiltonian can assume comparable roles. In quantum computing, particularly in algorithms like Quantum Annealing or the Quantum Approximate Optimization Algorithm (QAOA), the Hamiltonian often serves as a direct counterpart to the problem's objective function but is expressed within the framework of quantum mechanics. Concretely, the objective function can be conceptualized as the sum of the weights associated with all mesh points. If w denotes the weight contributed by a mesh point when it's set to 1, and 'x[i]' represents the binary variable indicating the state of the i-th mesh point (either 0 or 1), the objective function captures the essence of the weight distribution across the entire configuration. The objective function can be expressed as :

$$f(x) = -\sum_{i=1}^{32} x_i(\alpha w_i + \beta l_i) \tag{6.3}$$

α and β are constants that weigh the influence of w_i and l_i, respectively for a mesh with 32 domains representing the ferrite.

6.6 Random Agent (RA)

In the pursuit of optimizing passive component configurations, we employ a Random Agent algorithm to generate a substantial number of candidate configurations which is shown in Fig 6.2. The primary goal is to identify configurations that exhibit both minimal weight and maximal inductance, which are critical parameters in our project. Initially, a template file is read into the Python environment. The Random Agent algorithm then proceeds to generate 30000 unique candidate configurations. Each configuration consists of 32 binary variables, represented by either 1 or 0. These binary variables correspond to ferrite and air respectively. For each candidate configuration generated during an iteration, a new "modified.sif" file is created. This file, which is a key input for Finite Element Method (FEM) simulations, is stored in a separate working directory to maintain organizational clarity and computational efficiency. Subsequently, FEM simulations are conducted for each candidate configuration to determine the flux linkage and current. These simulations are critical as they provide the necessary data to calculate the inductance of each configuration which is explained in equation 5.19. Alongside inductance, the weight of each configuration is also calculated. The weight is a function of the material properties and the geometric arrangement of the components within each configuration. Both inductance and weight are crucial parameters; therefore, each candidate configuration is evaluated to form a tuple consisting of its weight and inductance values.

$$Tuple = (Weight, Inductance) \tag{6.4}$$

The optimization process seeks to identify the candidate configuration that exhibits the minimum weight and the maximum inductance, two key performance metrics for the project. This dual-objective approach ensures that the selected configuration not only minimizes the material usage but also maximizes the electromagnetic efficiency, resulting in a highly efficient and cost-effective design. To achieve this, the process begins with the generation and evaluation of a large number of candidate configurations, each representing a potential solution to the optimization problem. After the evaluation phase, a meticulous selection process is undertaken to determine the most optimal configuration from the generated pool. This selection is guided by two fundamental criteria. First, minimum weight, prioritizes configurations that use the least amount of ferrite material. By minimizing the material usage, the design reduces overall costs and contributes to a lighter, more practical solution for real-world applications. Moreover, reducing the weight can have additional benefits, such as ease of manufacturing, better thermal performance, and lower operational constraints in certain environments. Second, the maximum inductance criterion ensures that the selected configuration achieves the highest possible inductance. Inductance is a critical parameter as it reflects the effectiveness of the electromagnetic design. A higher inductance indicates superior electromagnetic performance, which is essential for meeting the technical requirements of the project. This criterion guarantees that the solution not only meets but exceeds the performance benchmarks, paving the way for a robust and reliable outcome. Ultimately, the configuration that best satisfies these two criteria, minimum weight and maximum inductance is chosen as the optimal solution. The rigorous nature of this process underscores the importance of achieving the dual objectives, laying a strong foundation for the success of the overall project.

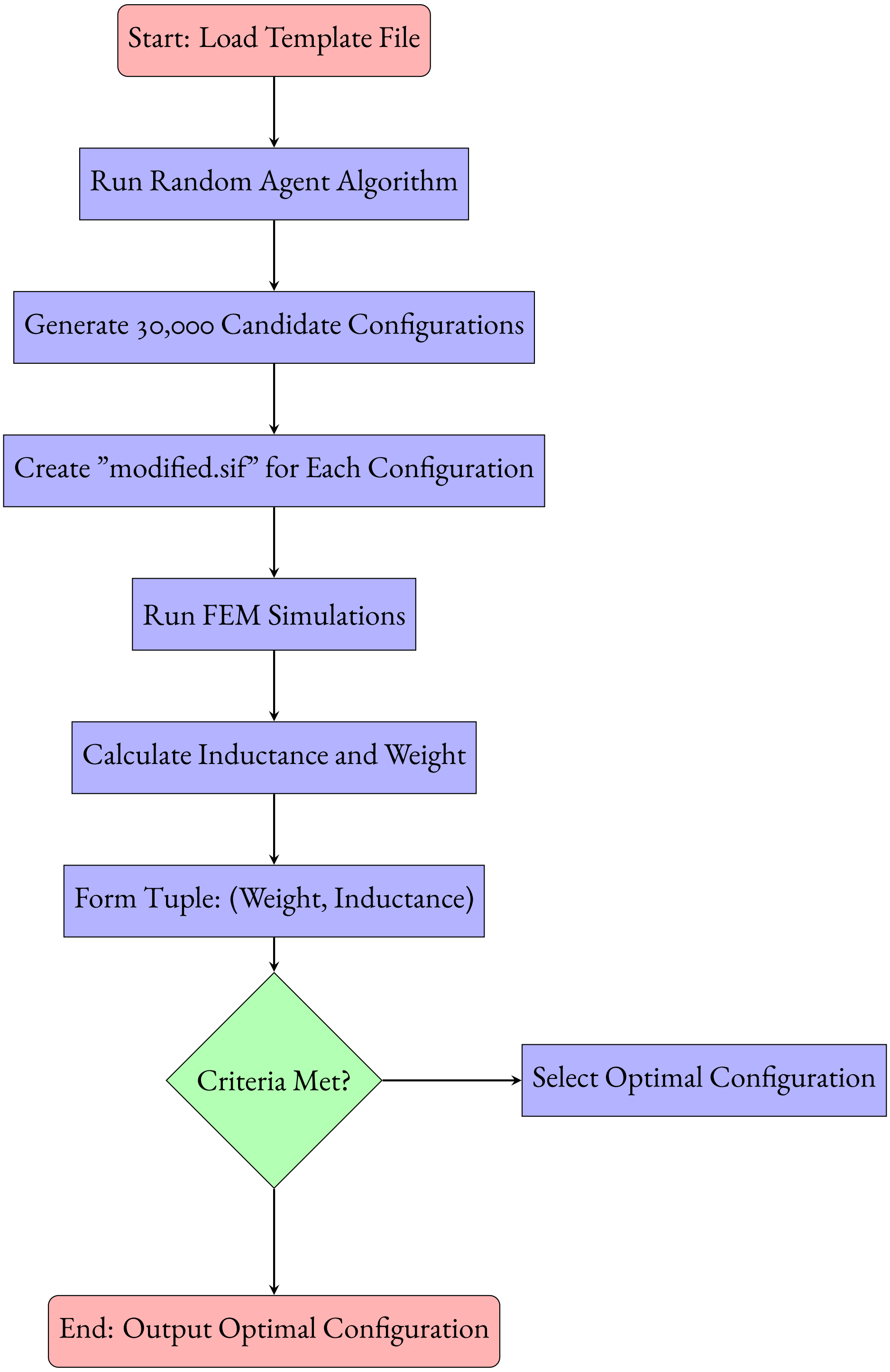

Start: Load Template File
Run Random Agent Algorithm
Generate 30,000 Candidate Configurations
Create "modified.sif" for Each Configuration
Run FEM Simulations
Calculate Inductance and Weight
Form Tuple: (Weight, Inductance)
Criteria Met?
Select Optimal Configuration
End: Output Optimal Configuration

Figure 6.2: Flowchart illustrating the optimization process of RA for passive component configurations. It includes key steps from template loading to the selection of the optimal configuration.

6.7 Simulated Annealing (SA)

Simulated annealing (SA) is a probabilistic technique for approximating the global optimum of a given function. Specifically, it is a metaheuristic to approximate global optimization in a large search space for an optimization problem. For large numbers of local optima, SA can find the global optimum [46]. SA is closely related to combinatorial optimization as it provides a robust technique for finding high-quality solutions in problems where the search space is discrete and vast [47].

In this process, temperature has a crucial role which is a controlling parameter that dictates the probability of accepting worse solutions as the algorithm searches for an optimal solution. High temperatures allow the algorithm to explore the solution space more freely, accepting worse solutions to escape local minima. As the temperature decreases, the algorithm becomes more selective, focusing on refining and converging to the best solution which can be seen in algorithm 6.1. The gradual reduction of temperature is crucial for balancing exploration and exploitation in the search process. Simulated annealing extends two of the most widely used heuristic techniques. The temperature distinguishes classes of rearrangements, so that rearrangements causing large changes in the objective function occur at high temperatures, while the small changes are deferred until low temperatures. This is an adaptive form of the divide-and-conquer approach. Like most iterative improvement schemes, the Metropolis algorithm proceeds in small steps from one configuration to the next, but the temperature keeps the algorithm from getting stuck by permitting uphill moves [14].

The algorithm is :

- we start with a random configuration of 1 and 0 as a list with 32 index and with the initial temperature of 50000.
- we set the parameter for the cooling schedule as a cooling rate of 0.9999.
- we compute the inductance of each body with FEM according to the eq 5.19 of the initial configuration.
- we calculate the objective function of the initial configuration with eq 8.1 and save it as the energy. The α is set to be 60 and β as 0.4. This is due to the prioritizing of the weight to be minimized.
- Generate a new solution, in the neighborhood of the current solution by flipping one of the spins and then calculate the energy of the new solution.
- If the new solution has a lower cost, we accept it as the new current solution. Otherwise, we accept it with a probability of $P = e^{-\frac{\Delta E}{T}}$. This probabilistic acceptance allows the algorithm to escape local minima.
- Reduce the temperature according to the cooling schedule.
- Repeat the iteration steps until the system has sufficiently cooled (i.e., the temperature is very low or a predefined number of iterations is reached). The current solution at the end of the process is taken as the optimal or near-optimal solution.

We choose SA because combinatorial optimization problems often have numerous local minima. The goal of this algorithm in this context is to minimize the 'energy' of the Ising model, where 'energy' is a metaphor for the objective or cost function of the optimization problem. The global minimum energy state corresponds to the optimal solution to the combinatorial optimization problem. Simple algorithms may get trapped in these suboptimal solutions. SA addresses this by allowing worse solutions to be accepted with a certain probability, controlled by a temperature parameter. This helps the algorithm escape local minima and explore a broader area of the solution space. In SA, we appear to have found a richer framework for constructing heuristic algorithms, since the extra control provided by introducing a temperature allows us to separate problems on different scales [14].

Algorithm 6.1 Simulated Annealing

```
Initialize: $T = T_0$, current solution $s$, best solution $s^*$
while stopping criterion not met
    Generate a neighbor solution $s'$
    Compute $\Delta E = f(s') - f(s)$
    if $\Delta E < 0$ or $\exp\left(-\frac{\Delta E}{T}\right) > \text{random}(0, 1)$
        $s \leftarrow s'$
    end if
    if $f(s) < f(s^*)$
        $s^* \leftarrow s$
    end if
    Update $T$ according to the cooling rate
end while
return $s^*$ =0
```

6.8 Simulated Annealing with Tunneling (tSA)

Quantum-inspired annealing introduces a nuanced enhancement to classical simulated annealing by incorporating principles from quantum mechanics [48]. In classical simulated annealing, a system transitions from its current energy state to a lower adjacent energy state by overcoming potential barriers through thermal fluctuations. However, quantum mechanics introduces the concept of quantum tunneling, where a particle can sometimes pass through a potential barrier, reaching a non-adjacent lower energy state without needing the thermal energy to overcome the barrier.

Incorporating this idea, quantum-inspired annealing allows the system to explore the solution space more effectively. This is achieved by occasionally permitting the simultaneous flipping of multiple spins in the new configuration, mimicking the quantum tunneling process [49]. Additionally, the algorithm increases the temperature suddenly when it reaches very low values. This strategic increase in temperature helps the system escape local minima and further expands the search space, enhancing the probability of finding a global optimum.

By integrating these mechanisms, quantum-inspired annealing leverages both the broad exploration capabilities of higher temperatures and the efficient traversal of energy landscapes via tunneling. This combination paves the way for more effective optimization and potentially superior results compared to classical simulated annealing.

Algorithm 6.2 Simulated Annealing with Tunneling

```
Initialize: $T = T_0$, current solution $s$, best solution $s^*$
while stopping criterion not met
    Generate a neighbor solution $s'$
    Compute $\Delta E = f(s') - f(s)$
    if $\Delta E < 0$ or $\exp\left(-\frac{\Delta E}{T}\right) > \mathrm{random}(0, 1)$
        $s \leftarrow s'$
    end if
    if $f(s) < f(s^*)$
        $s^* \leftarrow s$
    end if
    if $T \leq T_0 \times 0.00001$
        $T \leftarrow T_0$ {Reset temperature if it drops too low}
    else
        Update $T$ according to the cooling rate
    end if
end while
return $s^*$ =0
```

6.9 Knapsack Optimization

The Knapsack problem is a typical combinatorial optimization and NP-completely difficult problem. It can be described as follows: Given a set of m items, where item j has weight w_j and profit v_j, and an integer C is the total capacity of the knapsack. The problem is to choose a set of items such that their total profit is maximized, while the total weight is not larger than capacity C [50].

$$\begin{aligned} &\max f(x) = \sum_{j=1}^{m} v_j x_j \\ &\text{subject to } \sum_{j=1}^{m} w_j x_j \leq C, \\ &x_j \in \{0, 1\}, \quad j = 1, 2, \ldots, m. \end{aligned} \tag{6.5}$$

Here x_j represents the number of instances of the item j to include in the knapsack. In other words, if it is 1, we include the item and if it is 0 we exclude it. This process is illustrated in algorithm 6.3. We map the Knapsack Optimization to Ising Hamiltonian because quantum computers (particularly those using quantum annealing) are designed to minimize energy functions [13]. The Ising model describes such an energy landscape. In this approach, we leverage the quantum computer's natural ability to find the minimum of the system, solving complex combinatorial problems faster than classical methods.

It is worth mentioning that to map a knapsack problem into an Ising model, we need to transform the binary variables x_j into spin variables $(z_j = 2x_j - 1,\ \text{where}\, z_j \in \{-1, 1\})$. After which, we promote this new decision variables to Pauli spin operators with $Z_i|x_i\rangle = (-1)^{x_i}|x_i\rangle$ In our project, items are the 16 bodies of ferromagnets that we want to choose the best configuration for them according to their values which are the inductances that we get from FEM.

We want to maximize the total value ,inductance in our term, of the selected bodies. This can be written as a Hamiltonian H_{obj}:

$$H_{\text{obj}} = -\sum_{i=1}^{n} v_i \sigma_z^i \tag{6.6}$$

Where: σ_z^i is the Pauli-Z matrix applied to qubit i, which represents the binary variable x_i. The negative sign ensures that minimizing the Hamiltonian corresponds to maximizing the total value.

The constraint on the total weight is incorporated as a penalty term in the Hamiltonian. If the total weight exceeds the knapsack capacity W, a penalty is added. This can be shown as:

$$H_{\text{pen}} = \lambda \left(\sum_{i=1}^{n} w_i \sigma_z^i - W \right)^2 \tag{6.7}$$

The total Hamiltonian is a combination of the objective function and the penalty term:

$$H = H_{\text{obj}} + H_{\text{pen}} \tag{6.8}$$

$$H = -\sum_{i=1}^{n} v_i \sigma_z^i + \lambda \left(\sum_{i=1}^{n} w_i \sigma_z^i - W \right)^2 \tag{6.9}$$

The ground state of this Hamiltonian corresponds to the optimal solution of the knapsack problem.

Then we reformulate the Ising Hamiltonian into QUBO form. Quadratic unconstrained binary optimization (QUBO), is a combinatorial optimization problem with a wide range of applications from finance and economics to machine learning [51]. QUBO is an NP hard problem and we convert the Ising model to a QUBO, since it is suitable for quantum and hybrid algorithms in Qiskit. The function f_Q that assigns a value to each binary vector through :

$$f_Q(x) = x^\top Q x = \sum_{i=1}^{n} \sum_{j=i}^{n} Q_{ij} x_i x_j \tag{6.10}$$

Intuitively, the weight Q_{ij} is added if both x_i and x_j have value 1. When $i = j$, the values Q_{ii} are added if $x_i = 1$, as $x_i x_i = x_i$ for all $x_i \in \mathbb{B}$.

In our case converting Knapsack Hamiltonian to QUBO form will be as:

$$H_{\text{QUBO}} = -\sum_{i=1}^{n} v_i x_i + \lambda \left(\sum_{i=1}^{n} w_i x_i - W \right)^2 \tag{6.11}$$

Expanding the penalty term gives:

$$\lambda \left(\sum_{i=1}^{n} w_i x_i - W \right)^2 = \lambda \left(\sum_{i=1}^{n} w_i^2 x_i + 2 \sum_{i<j} w_i w_j x_i x_j - 2W \sum_{i=1}^{n} w_i x_i + W^2 \right) \tag{6.12}$$

Now, combining the value function and expanded penalty term:

$$H_{\text{QUBO}} = -\sum_{i=1}^{n} v_i x_i + \lambda \left(\sum_{i=1}^{n} w_i^2 x_i + 2 \sum_{i<j} w_i w_j x_i x_j - 2W \sum_{i=1}^{n} w_i x_i + W^2 \right) \tag{6.13}$$

The W^2 term is a constant and can be omitted since it doesn't affect the optimization process.

Algorithm 6.3 0/1 Knapsack Problem

```
Input: Array of weights $w_1, w_2, \ldots, w_n$, array of values $v_1, v_2, \ldots, v_n$, knapsack capacity $C$
Output: Maximum value that can be obtained in the knapsack
Create a 2D array $K[0..n][0..C]$
for each $i$ from 0 to $n$
    for each $w$ from 0 to $C$
        if $i = 0$ or $w = 0$
            $K[i][w] \leftarrow 0$
        else if $w_i \leq w$
            $K[i][w] \leftarrow \max(K[i-1][w], v_i + K[i-1][w - w_i])$
        else
            $K[i][w] \leftarrow K[i-1][C]$
        end if
    end for
end for
return $K[n][W]$ =0
```

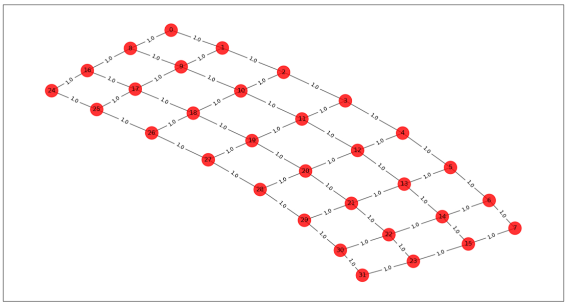


Figure 6.3: Our project with the max cut problem. As it can be seen from the figure the nodes represent the ferromagnetic bodies that we want to minimize them. Python module networkx is used. [6]

7
Quantum Computing

7.1 Hamiltonian

In the realm of quantum annealing, crafting the Hamiltonian formula involves tailoring it to encapsulate the essence of our optimization challenge. Generally, the Hamiltonian in a quantum annealing scenario comprises two principal components:

- The problem Hamiltonian (H_p), which encodes the optimization problem
- The driver Hamiltonian (H_0), which is used to initiate the quantum state in a superposition of all possible states.

Quantum annealing aims to uncover the ground state of Hp, where we seek to minimize the total energy (and consequently, the overall weight in this context). This process aligns with finding the most favorable configuration of mesh points that imparts the least weight contribution. Considering the goal of minimizing weight, let's assume that each mesh point i contributes a weight wi when set to '1' and imparts zero weight when set to '0'. In this light, the problem Hamiltonian can be articulated as follows:

$$H_p = \sum_{i=1}^{32} w_i \cdot Z_i \tag{7.1}$$

Here, Zi is the Pauli-Z operator applied to the i-th qubit. The Pauli-Z operator has eigenvalues +1 and -1, corresponding to the qubit states |0⟩ and |1⟩ respectively. Thus, each term in the sum contributes to the energy (and thereby the weight) only if the corresponding qubit (mesh point) is in the |1⟩ state.

7.1.1 Contribution to the hamiltonian's energy

The energy of a quantum system in a particular state is determined by applying the Hamiltonian to that state. Specifically, for the problem Hamiltonian Hp:

$$H_p = \sum_{i=1}^{32} w_i \cdot Z_i \tag{7.2}$$

In the context of a qubit (representing a mesh point), when the qubit is in the state |0⟩, the term wi · Zi effectively reduces to wi × (+1). In this case, it contributes nothing to the overall energy, signifying that the mesh point adds zero weight. Conversely, when the qubit is in the state |1⟩, the term becomes wi × (−1), actively contributing to the total energy. This scenario signifies that the mesh point adds its weight to the overall energy.

In the realm of quantum mechanics, the energy levels denoted by the eigenvalues of the Hamiltonian don't always have a straightforward correspondence to tangible aspects like weight. Yet, when delving into optimization problems tackled through quantum annealing, we construe lower energy states as more favorable solutions. Consequently, a qubit existing in the |1⟩ state (which imparts a negative contribution to the Hamiltonian's energy) implies that the associated configuration, where the mesh point adds weight, is less preferable concerning the pursuit of the minimum weight objective.

7.1.2 Driver hamiltonian H_0:

The design of the driver Hamiltonian usually aims to establish a uniform superposition of all conceivable states as the annealing process initiates. A widely adopted selection is:

$$H_0 = \sum_{i=1}^{32} X_i \tag{7.3}$$

where Xi is the Pauli-X operator on the i-th qubit. The Pauli-X operator acts like a bit-flip, taking |0⟩ to |1⟩ and vice versa. The use of the Pauli-X operator in the driver Hamiltonian for quantum annealing serves a purpose similar to the Hadamard gate in other quantum computing contexts – to prepare the system in a superposition state. However, it does so as part of a continuous, dynamic process unique to quantum annealing, rather than the discrete application of a gate.

7.2 Quantum Approximate Optimization Algorithm (QAOA)

Here we demonstrate an approach that is based on the Quantum Approximate Optimization Algorithm (QAOA) by Farhi, Goldstone, and Gutmann [52]. This algorithm is designed to solve combinatorial optimization problems and it is a variation form of form, which is based on quantum adiabatic annealing with a classical optimization loop on top of it. For combinatorial optimization, the quantum approximate optimization algorithm (QAOA)

[52] briefly had a better approximation ratio than any known polynomial-time classical algorithm (for a certain problem),[7] until a more effective classical algorithm was proposed [52]. The relative speed-up of the quantum algorithm is an open research question. A time evolution of a Hamiltonian is described by the Schrödinger equation which is :

$$i\hbar\frac{\partial}{\partial t}|\Upsilon(t)\rangle = \hat{H}|\Upsilon(t)\rangle \tag{7.4}$$

and it follows the following steps:

1. Defining a *cost Hamiltonian* H_C such that its ground state encodes the solution to the optimization problem.

2. Defining a *mixer Hamiltonian* H_M.

3. Defining the oracles $U_C(\gamma) = \exp(-i\gamma H_C)$ and $U_M(\alpha) = \exp(-i\alpha H_M)$, with parameters γ and α.

4. Repeated application of the oracles U_C and U_M, in the order:

$$U(\gamma, \alpha) = \prod_{i=1}^{N} \left(U_C(\gamma_i)U_M(\alpha_i)\right) \tag{7.5}$$

5. Preparing an initial state, that is a superposition of all possible states and applying $U(\gamma, \alpha)$ to the state.

6. Using classical methods to optimize the parameters γ, α and measure the output state of the optimized circuit to obtain the approximate optimal solution to the cost Hamiltonian. An optimal solution will be one that maximizes the expectation value of the cost Hamiltonian H_C [53].

7.3 Quantum Annealing Process

In quantum annealing, the system starts in the ground state of the driver Hamiltonian and evolves towards the ground state of the problem Hamiltonian. This process is governed by the time-dependent Hamiltonian:

$$H(t) = \left(1 - \frac{t}{T}\right) H_0 + \frac{t}{T}H_P \tag{7.6}$$

where t is the current time, and T is the total annealing time.

- Initial State and the Driver Hamiltonian (H_0) : The system starts in a simple quantum state that is easy to prepare. This state is typically the ground state of an initial Hamiltonian, known as the driver Hamiltonian (H_0). In this state, the quantum system is in a superposition of all possible states (solutions).

- Final State and the Problem Hamiltonian (H_p): The goal is to evolve this system into the ground state of the problem Hamiltonian (H_p), which encodes the solution to the optimization problem.

- The Role of $1 - \frac{t}{T}$ in the Annealing Schedule The quantum annealing process is governed by a time-dependent Hamiltonian that gradually transitions from H_0 to H_p over a period T (total annealing time).

- At the beginning ($t = 0$):

  $$1 - \frac{t}{T} = 1. \tag{7.7}$$

  The Hamiltonian is entirely H_d. The system is in a superposition of all possible states.

- During the process ($0 < t < T$):

  $$1 - \frac{t}{T} \tag{7.8}$$

  smoothly decreases from 1 to 0, and t/T increases from 0 to 1. The influence of H_0 decreases while that of H_p increases. This gradual change allows the quantum system to evolve naturally from the initial state towards the final state.

- 6- At the end ($t = T$):

  $$1 - \frac{t}{T} = 0.$$

  The Hamiltonian is entirely H_p. Ideally, the system is now in the ground state of H_p, which encodes the solution to the problem.

8 Results

The main question of our research is how we can make the component of the study (passive filter) lighter along with keeping the same performance. Inspired by the Ising model we indicate the model with discrete variables 1 and 0 representing the ferrite and air respectively. By considering their interaction terms we want to find the optimal solution that maximizes the number of zeros while having the most inductance that plays as a constraint in our objective function.

Fig 8.1 illustrates the initial design created manually. For visualizing the results, we use the Paraview tool, which is part of the Elmer suite used for Finite Element Method (FEM) simulations. In this figure, the red mesh elements represent the ferrite bodies, the darker red region in the center corresponds to the conductor, and the surrounding blue region indicates the air mesh elements. This human-designed model, while effective, is relatively heavy, motivating the need for an optimized, lighter alternative.

The magnetic field flux density for the human design can be seen in Fig 8.2. The red arrows illustrate the magnetic field flux density and evident is the fact that we have more density in the center than in the corners. In other words, the magnetic field flux density is continuous around the conductor, this observation suggests that the mesh elements in the corners may be reduced or removed without significantly affecting performance.
Considering the human-designed model, we apply three agents which are Random, Simulated Annealing, and Simulated Annealing with tunneling to optimize the weight. After this, we can compare the results and conclude the best candidate for the agent. We also try QA, a faster way to solve NP-hard problems. It should be noted that in this case, due to CPU limitations, a 16-domain case is investigated. We use Qiskit Optimization which is an open-source framework [12]. Built-in Python, the Qiskit Optimization module enables easy, efficient modeling of optimization problems for developers and optimization experts without quantum expertise. It uses classical optimization best practices and masks complex quantum programming. In order to find a solution for our problem on a quantum computer, we need first to map it to an Ising Hamiltonian and the Qiskit optimization module can generate it. After mapping we convert to QUBO to be solved with the quantum computers [54].

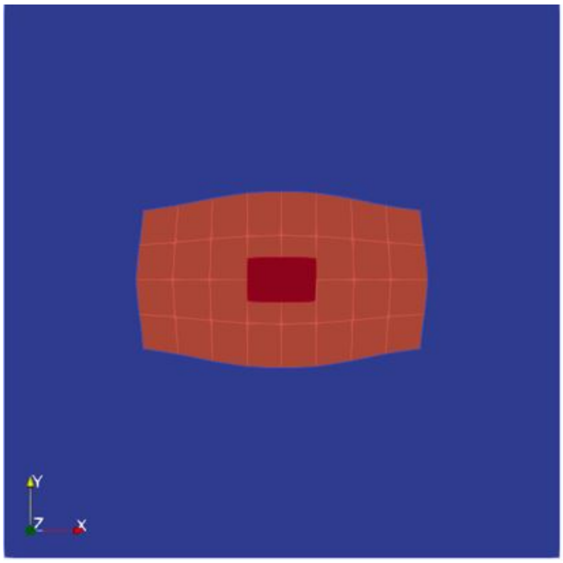


Figure 8.1: Visualization of the initial human-designed passive filter. The red mesh represents the ferrite material, the dark red in the center indicates the conductor, and the surrounding blue region depicts the air mesh. This design is effective but relatively heavy, prompting optimization efforts.

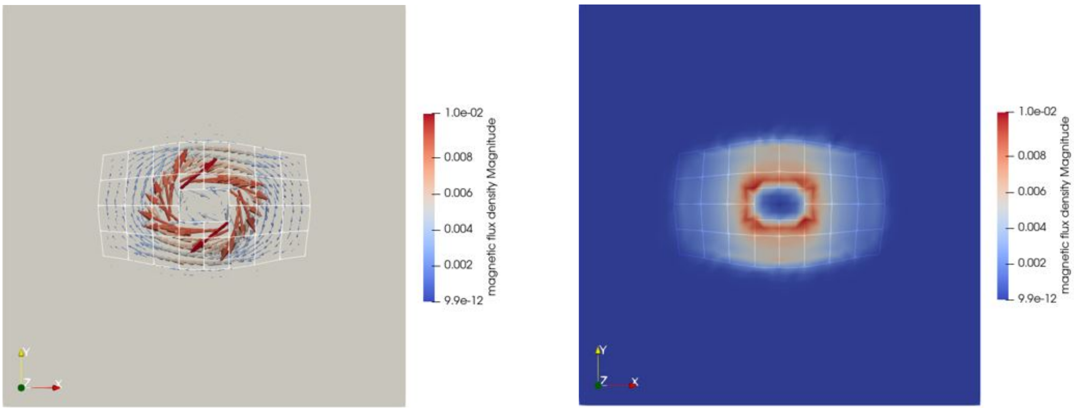


Figure 8.2: magnetic field flux density magnitude for the initial human-designed passive filter configuration. The red arrows in the left picture indicate the magnetic field flux density, showing a higher concentration in the center around the conductor compared to the corners.

8.1 Random Agent (RA)

The best candidate that our random Agent chooses is illustrated in Fig 8.3. After 30000 iterations, the configuration with the lowest weight and the highest inductance was determined to be [1 0 1 1 1 0 1 1 1 0 1 1 1 1 1 1 1 1 0 1 1 0 1 1 1 0 0 0 0 0 0]. By considering the weight of each body as 1, the weight of the best candidate of this agent is 21, with the total inductance of $5.7094831880985096 \times 10^{-11}$ provided in equation 5.20. By having the total flux linkage of the configuration which is gathered from the FEM, and divided by the current i, we have the total inductance. Using Python, we created a mapping that links each configuration's weight to its total inductance. Notably, we prioritized minimizing weight in this model. An analysis of these results indicates that while the Random Agent successfully minimized the weight of the passive filter, the outcome is not entirely satisfactory. This is due to the discontinuity in the magnetic field flux density observed in the center of the structure, as shown in Fig 8.3. Although the RA effectively reduced the overall functionality of the filter, this result was anticipated since the optimization was not fully implemented in this aspect.

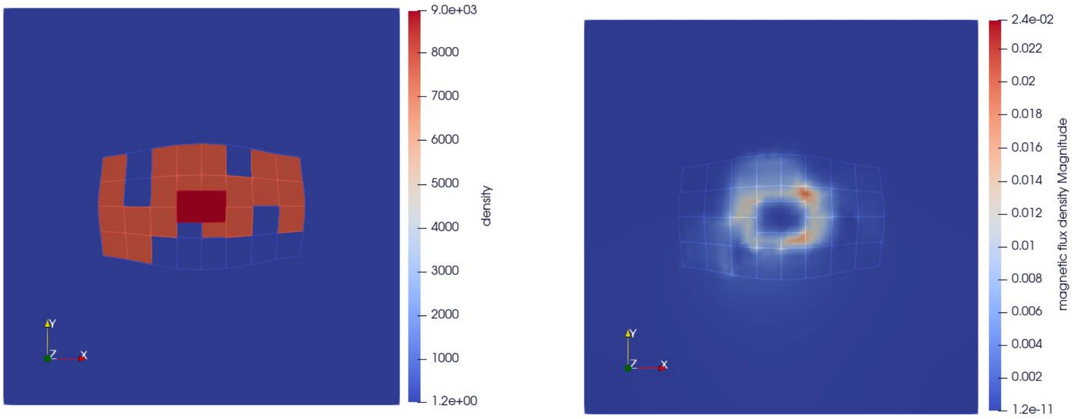


Figure 8.3: Visualization of the optimal configuration (left) and its resulting magnetic field flux density for the Random Agent (right). The elimination of bodies 20, 10, and 23 creates a discontinuity in the magnetic field flux density at the center.

8.2 Simulated Annealing Agent (SA)

SA uses the idea of annealing to escape from the local minima and reach to the best configuration. In this case, the process begins at a high initial temperature of 50,000 with a cooling rate of 0.9999 which is shown in Fig 8.7. Over 30,000 iterations, SA explores various configurations in search of the optimal solution according to the objective function:

$$f(x) = \sum_{i=1}^{32} x_i(\alpha w_i + \beta l_i) \tag{8.1}$$

So according to this agent, in each iteration, the objective function of the configuration is calculated and it compares it with the last candidate. If the current configuration has a lower energy, it is accepted as the new candidate. If the energy is higher, the acceptance of the new candidate depends on a probability that decreases as the temperature is reduced. The probability of accepting a new configuration is determined by the difference in energy between the current configuration and the new configuration, relative to the current temperature. This probability is computed using the following formula:

$$P_{\text{accept}} = \exp\left(\frac{E_{\text{current}} - E_{\text{new}}}{T_{\text{current}}}\right) \tag{8.2}$$

where E_{current} is the energy of the current configuration, E_{new} is the energy of the new configuration, and T_{current} is the current temperature. Once this probability P_{accept} is calculated, a random number p is drawn from a uniform distribution between 0 and 1. The new configuration is accepted if $p < P_{\text{accept}}$. In other words, the lower the energy of the new configuration relative to the current one, and the higher the temperature, the more likely the new configuration is to be accepted. The corresponding code logic is as follows:

```
accept_p = np.exp((current_energy - new_energy) / curr_temperature)
p = rnd.random()
if p < accept_p:
    curr_config = new_config
    config.append(curr_config)
    current_energy = new_energy
```

This code first calculates the acceptance probability P_{accept}. Then, it generates a random number p and compares it to P_{accept}. If p is less than P_{accept}, the new configuration is accepted, and the algorithm updates its current state to this new configuration. The best candidate for this Agent is [1 0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0] with the best energy of best energy = -156.589838481275 14 which is illustrated in Fig 8.4. Evident is the justification that this candidate is heavier than the candidate of the RA, but this configuration is better for us because we can see the continuity in our model.

Fig 8.6 illustrates the logarithm of the energy values across 30,000 iterations during the Simulated Annealing (SA) process. This analysis provides insights into the behavior of the SA algorithm as it searches for the optimal configuration. Throughout the entire process, significant fluctuations in the energy values are observed. This behavior is characteristic of the early stages of the SA algorithm, where the system is designed to accept a wide range of energy states, including higher energy configurations. The purpose of these fluctuations is to enable the algorithm to escape local minima and continue exploring the solution space effectively. As the number of iterations increases, the expectation is that the energy fluctuations should decrease, reflecting the algorithm's gradual convergence toward a global minimum. However, the plot shows that energy values continue to fluctuate with considerable amplitude throughout the entire process. This indicates that the cooling schedule, which is governed by a cooling rate of 0.9999, is relatively slow. Consequently, the system maintains a high degree of exploration even at later stages of the SA process, preventing premature convergence to a local minimum. The use of a logarithmic scale for energy, magnifies the perception of fluctuations, making even minor variations in energy appear more pronounced. The visible spikes and dips correspond to moments where the algorithm accepts suboptimal configurations, a deliberate feature of SA that allows the process to bypass local minima. Despite the apparent noise in the data, there is a subtle downward trend in the energy values, suggesting that the SA algorithm is gradually moving towards configurations with lower energy.

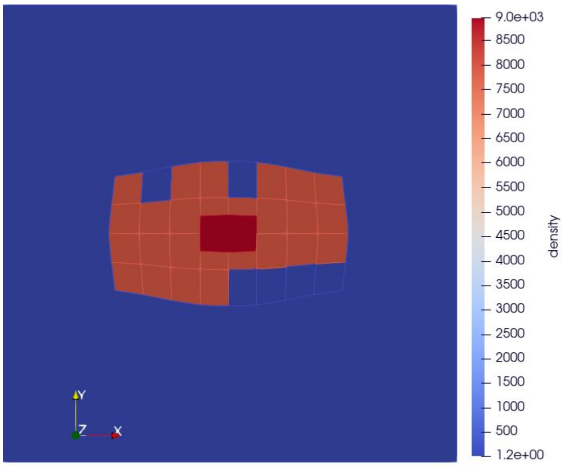


Figure 8.4: The best candidate of our Simulated Annealing Agent. It can be seen that this agent has conserved the continuity of magnetic field flux density and eliminated the last bodies.

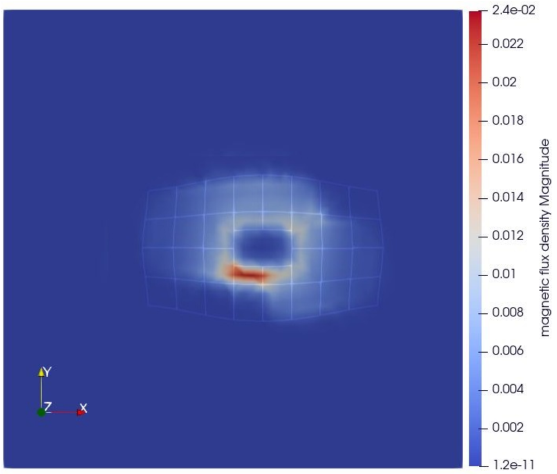


Figure 8.5: Magnetic field flux density for SA. Compared to RA, we have more density in the center which shows the impact of the optimization.

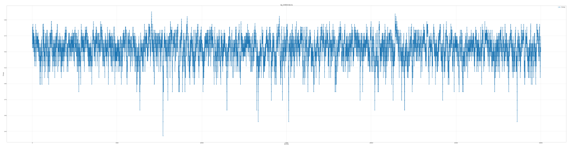


Figure 8.6: Logarithm of energy values versus the number of iterations in the Simulated Annealing process. The plot shows significant fluctuations with a gradual trend towards lower energy states over 30,000 iterations.

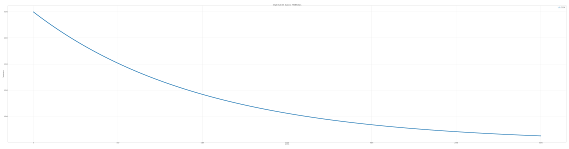


Figure 8.7: Temperature decreases during the Simulated Annealing process, starting from an initial temperature of 50,000 with a cooling rate of 0.9999, illustrating the gradual reduction in temperature over 30,000 iterations

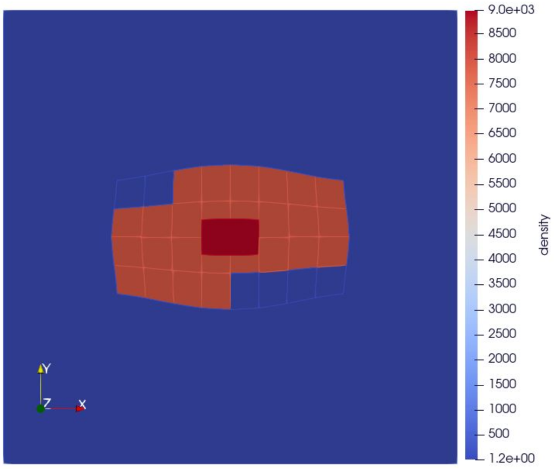


Figure 8.8: The best candidate of the tSA.

8.3 Tunneling Simulated Annealing Agent (tSA)

The approach of tSA is to convert standard simulated annealing to quantum-inspired simulated annealing by adding ideas from quantum tunneling. The idea is to raise the temperature when the agent has reached the temperature's end and flip multiple spins at each iteration 8.10. This approach is effective because it helps to find a better solution than classical simulated annealing. The temperature reduction is controlled by the cooling rate which is alpha and it is set to be 0.9999 the same as the SA agent. The best candidate that tSA finds is [1 0 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 0 0 1 1 1 0 1 1 0 1 1 1 0 1 0] and the best energy of best energy = -158.5898384812751 4 which is shown in Fig 8.8. This is what we expected to see better results than SA. We can see a complete continuity in the center and lower energy of our objective function than SA. Moreover, in Fig 8.9 it can be seen that we have a higher magnetic field flux density in the center compared with the SA agent and since it plays a key role, it should not be taken for granted.

8.4 Knapsack Optimization via QUBO formulation

The Knapsack Problem goal is to find a combination of items such that the total weight is within the capacity of the knapsack and maximize the total value of the items. In this thesis, we employ Qiskit, an open-source quan-

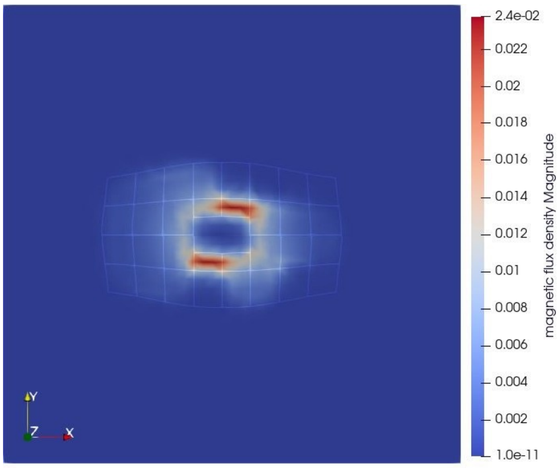


Figure 8.9: The magnetic field flux density for the tSA Agent with 30000 iterations

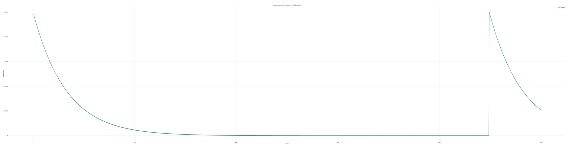


Figure 8.10: The temperature decreases gradually during the tSA process, starting at an initial value of 50,000 and following a cooling rate of 0.9999 over 30,000 iterations. At the end of the process, the temperature is then increased.

tum computing framework, to tackle the knapsack problem. we extend the classical Knapsack problem to a two-dimensional (2D) lattice configuration where each element of the lattice represents a binary variable (0 or 1), and they correspond to the inclusion or exclusion of each element in the optimal configuration. Each element of the lattice has associated physical properties such as weight and inductance, which we seek to optimize using quantum computation. Since we are using Qiskit, first we convert the problem into a Quadratic Unconstrained Binary Optimization (QUBO) format. This involves defining a Hamiltonian that combines the objective function and the penalty for violating constraints. To solve this, We both use Classical (Numpy Eigensolver), and Quantum Approximate Optimization Algorithm (QAOA). In both approaches, we arrive at the same solution with an objective function value of 2.768917890655 8642 and the selected items corresponding to the variables. In knapsack optimization the whole :

0.1006869263238621	0.3272448299581979	0.3295748209048719
0.09665066758164843	0.10610910693765153	0.1874750208847037
0.18321187336958628	0.10549461186016788	0.10547543668393612
0.18221514205585024	0.18799900418328772	0.10569193436011937
0.09703606079014983	0.3243472572576163	0.32641801288426925
0.09993785220159371		

and we want to maximize the values:

$$\begin{aligned}&0.1006869263238621 \cdot x_0 \ + \ 0.3272448299581979 \cdot x_1 \ + \ 0.3295748209048719 \cdot x_2 \ + \ 0.09665066758164843 \cdot x_3 \\ &+ \ 0.10610910693765153 \cdot x_4 \ + \ 0.1874750208847037 \cdot x_5 \ + \ 0.18321187336958628 \cdot x_6 \ + \ 0.10549461186016788 \cdot x_7 \\ &+ \ 0.10547543668393612 \cdot x_8 \ + \ 0.18221514205585024 \cdot x_9 \ + \ 0.18799900418328772 \cdot x_{10} \ + \ 0.10569193436011937 \cdot x_{11} \\ &+ \ 0.09703606079014983 \cdot x_{12} \ + \ 0.3243472572576163 \cdot x_{13} \ + \ 0.32641801288426925 \cdot x_{14} \ + \ 0.09993785220159371 \cdot x_{15}\end{aligned} \tag{8.3}$$

The optimization is subject to the weight constraint of the knapsack, which can be written as:

$$\begin{aligned}&2 \cdot x_0 + 3 \cdot x_1 + x_{10} + 2 \cdot x_{11} + 2 \cdot x_{12} + 3 \cdot x_{13} + 3 \cdot x_{14} + 2 \cdot x_{15} \\ &+ 3 \cdot x_2 + 2 \cdot x_3 + 2 \cdot x_4 + x_5 + x_6 + 2 \cdot x_7 + 2 \cdot x_8 + x_9 \leq 30\end{aligned} \tag{8.4}$$

here the x_i are the binary variable and the the coefficients are the weights of each body we want to The solution reached is:

$$[0, 1, 2, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]$$

In this 16-bodies case, it can be seen that the body number 3 is removed. In this scenario, the approach results in the exclusion of a single item from the optimal selection. While this method may be considered simplistic, it effectively reduces the overall weight of the knapsack while minimizing the associated loss in total value. This balance between weight efficiency and value retention highlights a pragmatic solution in the context of optimization. It can be seen that the total capacity of the knapsack is considered as 30 and we have an inequality in equation 8.4. In order to solve this equation first we need to convert inequality constraints into equality constraints with additional slack variables to remove inequality constraints from QuadraticProgram. In other words, by introducing the slack variable $\mathbf{s} \geq 0$, the inequality $\mathbf{Ax} \leq \mathbf{b}$ can be converted to the equation $\mathbf{Ax} + \mathbf{s} = \mathbf{b}$. In the knapsack problem, slack variables help enforce our constraints. In our case, by introducing S_0 we convert equation 8.4 to :

$$\begin{aligned}&2 \cdot x_0 + 3 \cdot x_1 + x_{10} + 2 \cdot x_{11} + 2 \cdot x_{12} + 3 \cdot x_{13} + 3 \cdot x_{14} + 2 \cdot x_{15} \\ &+ 3 \cdot x_2 + 2 \cdot x_3 + 2 \cdot x_4 + x_5 + x_6 + 2 \cdot x_7 + 2 \cdot x_8 + x_9 + s_0 = 30\end{aligned} \tag{8.5}$$

S_0 represents the difference between the left-hand side of the inequality and the constant (30) on the right-hand side. Essentially, it takes up any unused capacity in the constraint. In mathematical terms:

$$\begin{aligned}&s_0 = 30 - (2 \cdot x_0 + 3 \cdot x_1 + x_{10} + 2 \cdot x_{11} + 2 \cdot x_{12} + 3 \cdot x_{13} + 3 \cdot x_{14} + 2 \cdot x_{15} + 3 \cdot x_2 \\ &+ 2 \cdot x_3 + 2 \cdot x_4 + x_5 + x_6 + 2 \cdot x_7 + 2 \cdot x_8 + x_9 \ )\end{aligned} \tag{8.6}$$

The quantum circuit elements are shown in Fig 8.11. It is a variational quantum algorithm designed for solving our optimization problem using parameterized quantum gates. The circuit consists of multiple layers of parameterized rotations R_Y and R_Z which act on individual qubits, and entangling gates (CNOT gates) that connect qubits across different layers. The role of the quantum circuit is to explore the solution space more efficiently than classical algorithms can. By leveraging quantum superposition and entanglement, the circuit can represent and manipulate multiple potential solutions simultaneously, providing a potential speedup for solving complex optimization problems. After running the quantum algorithm, the final state of the qubits is measured, and the resulting bitstring represents a candidate solution to the knapsack problem. Through iterative optimization, the QAOA adjusts the quantum circuit parameters to find the optimal or near-optimal solution. The final verdict of the approach is still to be tested on real quantum hardware.

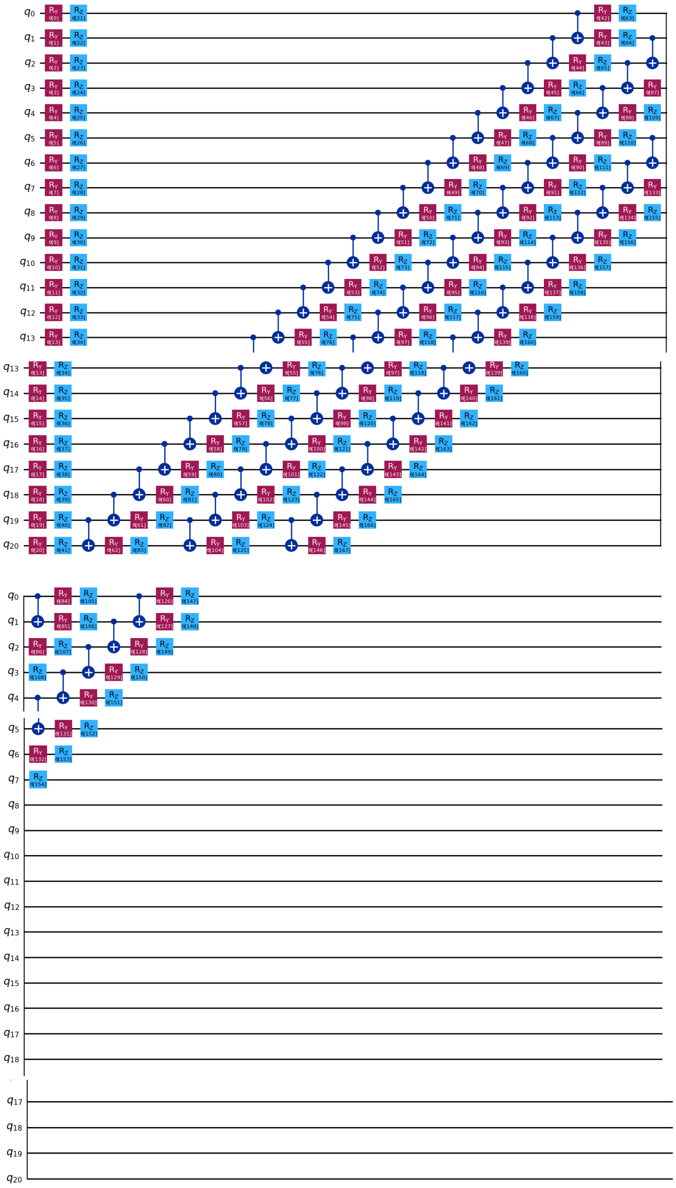


Figure 8.11: Overview of the quantum circuits used to solve the Knapsack problem. The circuits show different stages of qubit preparation and entanglement, from initial setup to complex entanglement across multiple qubits.

9 Conclusion

An industry-relevant component design problem is tackled. The optimization problem of a CM filter is prescribed using FEM software, coupled with Python based pipeline, designed specifically for this MSc. The optimization problem is designed to decrease the weight of a CM passive filter while its function(L) is preserved. 3+1 agents (RA,SA, QT-SA, and a QISKIT version) were designed. RA has worse results, SA improves this, and tSA is even better. Through these four approaches, the project demonstrates a progression in solution quality, with each agent incrementally improving upon the last. The results highlight the feasibility of applying classical and quantum-inspired algorithms to problems, suggesting pathways for integrating emerging quantum methods in real-world engineering challenges.

References

[1] D. Inoue, A. Okada, T. Matsumori, K. Aihara, and H. Yoshida, "Traffic signal optimization on a square lattice with quantum annealing," *Nature Scientific Reports*, vol. 7, no. 1, p. 11799, 2017.

[2] N. Davis, "An introduction to filters," September 30 2023.

[3] A. Bakshi and M. V. Bakshi, *Electrical Machines - I.* Pune, India: Technical Publications Pune, India, May 2006.

[4] BETA CAE Systems, "BETA CAE Systems," https://www.beta-cae.com/, 2024, accessed: 2024-11-17.

[5] D. Trummer, "Applying ant colony optimization to the periodic vehicle routing problem with time windows," Diplomarbeit, University of Vienna, Vienna, Austria, 2024, diplomarbeit zur Erlangung des akademischen Grades Diplom-Ingenieur im Rahmen des Studiums Computational Intelligence.

[6] NetworkX developers, "Networkx," n.d., accessed: 2023-11-17. [Online]. Available: https://networkx.org/

[7] V. Cars, "Our story," 2024, accessed: 2024-11-16. [Online]. Available: https://www.volvocars.com/intl/our-story/

[8] ——, "Sustainability highlights," 2024, accessed: 2024-11-16. [Online]. Available: https://www.volvocars.com/intl/sustainability/highlights/

[9] C. Kingsley, *Electric Machinery.* McGraw-Hill, 1992.

[10] A. Schrijver, *Combinatorial Optimization: Polyhedra and Efficiency*, ser. Algorithms and Combinatorics. Springer, 2003, vol. 24.

[11] T. Mariani and S. R. Vergilio, "Title of the article," *Information and Software Technology*, vol. 91, pp. 94–106, 2017.

[12] Q. Community, "Max-cut and traveling salesman problem examples," 2024, accessed: 2024-11-16. [Online]. Available: https://qiskit-community.github.io/qiskit-optimization/tutorials/06_examples_max_cut_and_tsp.html

[13] A. Lucas, "Ising formulations of many np problems," *Frontiers in Physics*, vol. 2, p. 5, 2014.

[14] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983.

[15] C. N. Yang, "The spontaneous magnetization of a two-dimensional ising model," *Physical Review*, vol. 85, p. 808, Mar. 1952. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRev.85.808

[16] E. F. Combarro, S. González-Castillo, and A. Di Meglio, *A Practical Guide to Quantum Machine Learning and Quantum Optimization: Hands-on Approach to Modern Quantum Algorithms*. Packt Publishing, 2023.

[17] S. Gulur, V. M. Iyer, and S. Bhattacharya, "Passive cm filter configuration for a multistage grid-tied solid state transformer," *IEEE Journal of Emerging and Selected Topics in Industrial Electronics*, vol. 4, no. 3, pp. 710–717, 2023.

[18] P. Papadopoulos, *ME280A: Introduction to the Finite Element Method*. Berkeley, CA: Department of Mechanical Engineering, University of California, Berkeley, 2015.

[19] F. J. Duarte, *Tunable Laser Optics*. San Diego: Academic Press, 2003.

[20] D. P. Hampshire, "A derivation of maxwell's equations using the heaviside notation," *arXiv preprint arXiv:1810.12061*, October 2018. [Online]. Available: https://arxiv.org/abs/1810.12061

[21] J. Pyrhonen, T. Jokinen, and V. Hrabovcova, *Design of Rotating Electrical Machines*. United Kingdom: Wiley, 2013.

[22] I. D. Vagner, B. I. Lembrikov, and P. R. Wyder, *Electrodynamics of Magnetoactive Media*. Springer Science & Business Media, 2003.

[23] H. Johnson, H. W. Johnson, and M. Graham, *High-speed Signal Propagation: Advanced Black Magic*. Prentice Hall Professional, 2003.

[24] I. D. Vagner, B. I. Lembrikov, and P. R. Wyder, *Electrodynamics of Magnetoactive Media*. Springer Science & Business Media, 2003.

[25] V. P. Bhatnagar, *A Complete Course in ISC Physics*. Pitambar Publishing, 1997, retrieved 2023-07-03.

[26] Y. Liang and H. Chen, "Circuit-based flux linkage measurement method with the automated resistance correction for srm sensorless position control," *IET Electric Power Applications*, September 2018, received on 15th April 2018, Revised on 03rd August 2018, Accepted on 24th August 2018, E-First on 27th September 2018. [Online]. Available: https://www.ietdl.org

[27] Y. Singh, *Electro Magnetic Field Theory*. Pearson Education India, 2011.

[28] C. L. Wadhwa, *Electrical Power Systems*. New Age International, 2005.

[29] R. A. Pelcovits and J. Farkas, *Barron's AP Physics C*. Barron's Educational Series, 2007.

[30] E. M. Purcell and D. J. Morin, *Electricity and Magnetism*. Cambridge University Press, 2013.

[31] Elmer Development Team, *ElmerGUI Manual*, CSC - IT Center for Science, Espoo, Finland, 2023.

[32] CSC – IT Center for Science, *ElmerGrid Documentation*, [Year], accessed: 2024-11-17. [Online]. Available: https://www.csc.fi/web/elmer/documentation

[33] J. Ruokolainen, M. Malinen, P. Råback, T. Zwinger, A. Pursula, and M. Byckling, *ElmerSolver Manual*, CSC – IT Center for Science, April 2023.

[34] Kitware Inc., "ParaView," https://www.paraview.org/, 2024, version X.X.

[35] E. D. Team, *ElmerSolver Manual*, n.d., accessed: 2023-11-17. [Online]. Available: https://www.nic.funet.fi/pub/sci/physics/elmer/doc/ElmerSolverManual.pdf

[36] J. Ruokolainen, M. Malinen, P. Råback, T. Zwinger, A. Pursula, and M. Byckling, *ElmerSolver Manual*, CSC – IT Center for Science, April 2023, 6.

[37] INFORMS Computing Society, "The Nature of Mathematical Programming," Archived March 5, 2014, at the Wayback Machine, 2014, accessed: 2024-11-17. [Online]. Available: https://web.archive.org/

[38] S. Aaronson, "P=?np," *Electronic Colloquium on Computational Complexity*, January 8 2017, archived from the original on June 17, 2020. [Online]. Available: https://www.example.com/archive/Aaronson_P_NP

[39] R. M. Karp, "Reducibility among combinatorial problems," in *Complexity of Computer Computations*, R. E. Miller and J. W. Thatcher, Eds. New York: Plenum, 1972, pp. 85–104.

[40] R. Sedgewick and K. Wayne, "Introduction to computability," https://introcs.cs.princeton.edu/java/54computability/, accessed: 2024-11-17.

[41] K. Huang, *Introduction to Statistical Physics*, 2nd ed. CRC Press, 2009.

[42] M. N. Rosenbluth, "Genesis of the monte carlo algorithm for statistical mechanics," in *AIP Conference Proceedings*, vol. 690, 2003, pp. 22–30.

[43] E. Boros and P. L. Hammer, "Pseudo-boolean optimization," *Discrete Applied Mathematics*, vol. 123, p. 155, 2002.

[44] G. Gallavotti, *Statistical Mechanics*, ser. Texts and Monographs in Physics. Berlin: Springer-Verlag, 1999.

[45] R. Peierls and M. Born, "On ising's model of ferromagnetism," *Mathematical Proceedings of the Cambridge Philosophical Society*, vol. 32, no. 3, p. 477, 1936.

[46] *What is Simulated Annealing?*, retrieved 2023-05-13. [Online]. Available: http://www.cs.cmu.edu

[47] C. M. University, "What is simulated annealing?" 2023, retrieved: 2023-05-13. [Online]. Available: https://www.cs.cmu.edu

[48] S. Yarkoni, E. Raponi, T. Bäck, and S. Schmitt, "Quantum annealing for industry applications: Introduction and review," *Journal Name*.

[49] J. McCaffrey, "Simulated annealing optimization using c# or python," Microsoft Research, n.d., accessed: 2023-05-13. [Online]. Available: https://www.microsoft.com/en-us/research/publication/simulated-annealing-optimization-using-c-or-python/

[50] H. Kellerer, U. Pferschy, and D. Pisinger, *Knapsack Problems*. Berlin: Springer, 2004, retrieved 5 May 2022.

[51] G. Kochenberger, J.-K. Hao, F. Glover, M. Lewis, Z. Lu, H. Wang, and Y. Wang, "The unconstrained binary quadratic programming problem: A survey," *Journal of Combinatorial Optimization*, vol. 28, pp. 58–81, 2014. [Online]. Available: https://doi.org/10.1007/s10878-014-9734-0

[52] E. Farhi, J. Goldstone, and S. Gutmann, "Quantum approximate optimization algorithm," *arXiv preprint arXiv:1411.4028*, 2014. [Online]. Available: https://arxiv.org/abs/1411.4028

[53] P. Chandarana, N. N. Hegade, K. Paul, F. Albarrán-Arriagada, E. Solano, A. del Campo, and X. Chen, "Digitized-counterdiabatic quantum approximate optimization algorithm," *Physical Review Research*, vol. 4, no. 1, p. 013141, 2022. [Online]. Available: https://doi.org/10.1103/PhysRevResearch.4.013141

[54] Q. D. Team, "Qiskit.optimization.converters.quadraticprogramtoqubo," Online, 2023, accessed: 2023-11-17. [Online]. Available: https://docs.quantum.ibm.com/api/qiskit/0.27/qiskit.optimization.converters.QuadraticProgramToQubo

Acknowledgments

I would like to express my sincere thanks to my supervisor, Prof. Marco Zanetti, for his valuable support throughout my academic journey. The opportunity to work on my thesis project at Volvo Cars has been a truly defining experience in my career. I am especially grateful to my thesis supervisor, Dr. Raik Orbay, for his continuous guidance, encouragement, and unwavering belief in my abilities. His daily mentorship was essential to the successful completion of this work. Working with him, I learned an important lesson: when we approach our work with passion and enjoyment, success naturally follows. This idea often reminds me of a quote by Hegel: ”Nothing great has been and nothing great can be accomplished without passion and continuity,” a belief I deeply resonate with and try to live by every day.

I also want to thank my co-supervisor, Dr. Bahram Ganjipour, whose expertise in physics helped bring crucial insights to our project. This journey wouldn’t have been possible without his support. I will always remember our meetings, where we discussed the project from both physics and engineering perspectives, enriching the work and expanding my understanding.