

DESIGN OF MAGNETIC SOFT ROBOTS
USING EVOLUTIONARY ALGORITHMS

DESIGN OF SMALL SCALE MAGNETIC SOFT ROBOTS USING
CMA-ES AND MATERIAL POINT METHOD

By FATEMEH NOROUZIANI, B.Sc

A Thesis Submitted to the School of Graduate Studies in Partial
Fulfillment of the Requirements for
the Degree Master of Applied Science

McMaster University © Copyright by Fatemeh Norouziani, April 2025

McMaster University

MASTER OF APPLIED SCIENCE (2025)

Hamilton, Ontario, Canada (Computing and Software)

TITLE: Design of Small Scale Magnetic Soft Robots Using CMA-
ES and Material Point Method

AUTHOR: Fatemeh Norouziani
B.Sc (Software Engineering),
Qazvin Islamic Azad University, Qazvin, Iran

SUPERVISOR: Dr. Onaizah Onaizah

NUMBER OF PAGES: xix, 64

Lay Abstract

Small scale magnetic soft robots are small machines with flexible bodies that can be remotely controlled by a magnetic field. Due to their small size and flexible body, they have a wide range of applications, especially in the medical field. However, designing these robots is a complex task that requires a vast range of knowledge and often relies on a trial and error approach. This research explores how artificial intelligence (AI) techniques can be used to automate this process. To achieve this, two AI-based methods were implemented: Genetic Algorithm (GA) and Covariance Matrix Adaptation Evolution Strategy (CMA-ES). These two methods are different dialects of Evolutionary Algorithms (EA), which are a group of algorithms that are inspired by natural evolution and improve a population of solutions through iterations, similar to how organisms evolve in nature. The results show that CMA-ES successfully designs and optimizes the magnetization profile of a particular magnetic soft robot structure and outperforms human design configuration by 45.5% in terms of the horizontal speed of the robot. However, GA failed to achieve comparable performance and was unable to compete with the efficiency of human-designed robots. CMA-ES performs better likely due to its ability to adjust its search strategy during the optimization process, while GA relies on a fixed approach that cannot adapt to the problem. This highlights the efficiency and superiority of CMA-ES in designing and optimizing

magnetic soft robots.

Abstract

Small scale untethered magnetic soft robots, due to their flexible body, small size, and remote actuation capabilities, have a wide range of applications in fields such as bioengineering and healthcare. However, designing these robots is a highly challenging task requiring precise optimization of material and magnetic configurations. Traditionally, this process relies on trial and error, which can limit innovation due to designer bias towards traditional methods, and provides an inefficient means of exploring the vast space of possible design solutions.

To address this challenge, this study presents an AI-driven optimization that utilizes Covariance Matrix Adaptation Evolution Strategy (CMA-ES) combined with a Material Point Method (MPM) based simulation environment, to successfully design and optimize the magnetic profile of a strip-shaped walking robot, and outperform the human-designed magnetization profile as shown by the increase in the robot's horizontal speed by 45.5%.

Additionally, we compared the CMA-ES with the Genetic Algorithm (GA), which is a widely used evolutionary optimization method. The result demonstrates that CMA-ES significantly outperforms GA in terms of convergence speed. While CMA-ES designed robots significantly outperformed the human-designed configuration, GA was unable to come close to the original design performance. This performance gap is

likely because CMA-ES dynamically adapts its search distribution by using a covariance matrix. This allows for more efficient exploration of design space and increases the performance of this algorithm in solving complex problems, while GA relies on fixed mutation and crossover strategies that limit its ability.

These findings highlight the potential of AI-driven optimization in the design of the robots, enabling more innovative and efficient designs. Future work can explore multi-objective optimization to balance competing goals such as maximizing locomotion speed, improving motion accuracy, and minimizing manufacturing complexity. Additionally, this approach can be applied to different robots and locomotion modes. Extra experimental validation involving physical fabrication and testing can be done to confirm the effectiveness of the optimized designs in real-world conditions. This research represents a small step towards a fully automated design process for a magnetic soft robot.

To my husband, Alireza, my constant source of strength and love. Despite the distance, you were always by my side, reminding me that I was never alone.

Acknowledgements

I would like to sincerely express my deepest appreciation to my supervisor, Dr. Onaizah Onaizah, for all her support and guidance, and for generously sharing her knowledge with me throughout my research. I am also deeply grateful to Dr. Stephen Kelly, whose help and advice were crucial at every stage of this work; this thesis would not have been possible without his guidance. I am extremely grateful to Dr. Jacques Carette for reviewing my thesis and for all of the valuable feedback and support he provided. Additionally, I would like to thank Stefanie for her kindness and support, especially during the challenging times.

A very special thank you goes to my parents, who have always encouraged me to pursue my dreams and shown great patience, allowing me to prioritize my studies without ever making me feel guilty. Thank you to my brother, who has always believed in me and constantly encouraged me to believe in myself.

I am forever grateful to my husband, who always supported me from thousands of miles away. His love, understanding, patience, and unconditional support were the source of my strength throughout this journey. I would also like to thank my husband's family for their kindness, support, and encouragement, which meant so much during this time.

Finally, I would like to thank my friends Melika, Atiyeh, Amir Hossein, Tanya,

Maryam, and Toktam, as well as my labmates, especially Shivam, for their friendship, support, encouragement, and all the wonderful memories we created together.

Thank you all for being part of this important chapter of my life.

Table of Contents

Lay Abstract	iii
Abstract	v
Acknowledgements	viii
Abbreviations	xviii
Declaration of Academic Achievement	xx
1 Introduction	1
2 Background	4
2.1 Introduction to Soft Robotics	4
2.2 Magnetic Actuation and Fundamental Principles	9
2.3 Optimization of Programmable Domains in Soft Magnetic Robots . .	13
2.4 Introduction to Evolutionary Algorithms	16
3 Methodology	30
3.1 Algorithm	31
3.2 Representation	35

3.3	Evaluation Function	35
3.4	Simulation Environment	39
4	Results	45
5	Conclusion	52

List of Figures

2.1	Types of Magnetic Soft Materials. Magnetic soft materials can be categorized into a) Discrete Systems, which have rigid magnets inside a flexible structure, and b) Continuous Systems, which have micro/nano magnetic particles in a flexible matrix.	6
2.2	Magnetic Hysteresis of Hard and Soft Magnetic Materials. a) Hard magnetic materials show high coercivity (H_c) and remanence (M_r) while b) soft magnetic materials exhibit low coercivity (H_c) and remanence (M_r) with a narrow hysteresis loop. Figure recreated based on [1].	8
2.3	Magnetic torque applied on a dipole in the presence of an external magnetic field.	10
2.4	Magnetic force applied on a magnetic dipole in the presence of a nonuniform magnetic field	11
2.5	Deformation of a magnetic soft robot under a uniform magnetic field. a) The robot remains in its natural position when no external magnetic field is present. b) In a uniform magnetic field, the magnetic particles of each segment experience torque and align with the external field which leads to the deformation of the robot.	12

2.6	The voxel-encoding and development process of magnetic soft robots. Reproduced from [2] under the Creative Commons Attribution (CC BY) license. "Fabrication" subfigure taken from [3]	13
2.7	Flowchart of evolutionary algorithms process. The process involves initializing a population, selecting parents, applying crossover and mutation to generate new offspring, and selecting survivors and updating the population based on that. This cycle continues for a set number of generations or until a specific condition, such as reaching an optimal solution, is met. Figure recreated based on [4].	17
2.8	Programmable MSCR with varied remanent magnetization and rigidity along its body. a) Each voxel of the robot has a specific remanent magnetization that is set by adjusting magnetic particle volume fraction (ϕ). b) The relation between magnetic particle volume fraction (ϕ) and normalized magnetization strength and shear modulus is shown in black and red, respectively. Reproduced from [5] under the PNAS license.	24
2.9	a) The flowchart of the optimization algorithm, illustrating the representation of individuals in the initial population, stochastic universal sampling, crossover, and mutation. b) The highest and mean normalized half workspace of 100 individuals plotted over 50 generations. c) Magnetic distribution and fitted function of the optimized design at generation 40. d) Normalized half workspace of the robot based on the fitted distribution of the optimized design in generation 40. Reproduced from [5] under the PNAS license.	26

2.10	Voxel-Encoding Printing of Hard-Magnetic Soft Active Materials. a) special curvature of a Sidewinder snake, which leads to its unique sidewinder motion. b) The structure of a three-layered printed voxel with programmed density and direction. d) The encoding of voxels with three printed layers. e) Magnetic density of voxels with a varied number of layers. The dashed line represents the ideal magnetic density for this type of robot. Reproduced from [6] under the Creative Commons Attribution License (CC BY 4.0).	27
2.11	Evolutionary algorithm applied to voxel-encoding printing. a) The algorithm gets the desired deformation as input, and EA finds the required distribution to achieve the targeted deformation. b) Flowchart of the implemented EA. Reproduced from [6] under the Creative Commons Attribution License (CC BY 4.0).	29
3.1	This figure illustrates the position measurement for robots with different final shapes. a) A robot ending the simulation with a straight shape. b) A robot ending the simulation with a bent shape. The Blue, yellow, and red arrows represent displacement measurements taken from the corresponding colored segments.	37
3.2	The best fitness values over 100 generations without height restriction.	38

3.3	This figure illustrates the behavior of the robot optimized with and without height restriction over time. a) The robot optimized with a height restriction shows a stable and continuous walking behavior by bending and straightening in a cycle, while the robot optimized without a height restriction b) demonstrates a flipping behavior that is not continuous, and the robot gets stuck and is unable to progress further.	41
3.4	Comparison of simulation stability. a) Stable simulation with sufficient particle density and a small time step b) Unstable simulation resulting from reduced particle density and increased time step causes particles to disperse and loss of structure	42
3.5	Plot of the B_y and B_z components of the magnetic field over six cycles.	43
4.1	The best fitness values over 100 generations for five independent GA runs.	46
4.2	The best fitness values over 100 generations for five independent CMA-ES runs.	48
4.3	Mean of the best fitness achieved over 100 generations for five independent runs of GA and CMA-ES.	49
4.4	The best fitness values over 100 generations for five independent CMA-ES runs with the restart strategy.	50

4.5 This figure illustrates the magnetic configurations of the best individuals from the best generations of each seed. The original robot's magnetic configuration is also included for reference. The direction of magnetization of each segment is represented by an arrow, while the color of the arrows indicates its magnitude. 51

List of Tables

2.1	Summary of Simple Genetic Algorithm (SGA) components. Reproduced from [4].	19
2.2	Summary of Evolution Strategy (ES) components. Reproduced from [4].	20

Abbreviations

Abbreviations

AI	Artificial intelligence
GA	Genetic Algorithm
CMA-ES	Covariance Matrix Adaptation Evolution Strategy
EA	Evolutionary Algorithm
SGA	Simple Genetic Algorithm
ES	Evolution Strategies
PSO	Particle Swarm Optimization
DE	Differential Evolution
LCS	Learning Classifier Systems
EDA	Estimation of Distribution Algorithms
GP	Genetic Programming

MPM	Material Point Method
FEM	Finite Element Method
MSCR	Magnetic Soft Continuum Robot
DIW	Direct Ink Writing

Declaration of Academic Achievement

I, Fatemeh Norouziani, declare that this thesis is the result of my own research conducted under the supervision of Dr. Kelly and Dr. Onaizah. All sources of information used or referred to in this thesis have been properly acknowledged within the text and are fully listed in the references section.

Chapter 1

Introduction

Small scale robots range from a few centimeters to nanometer scale [7–9]. The special characteristics of these robots, such as their small size and precision, makes them suitable tools in many different fields, such as bioengineering [10, 11], microfabrication [12–14], and healthcare [15–22]. These robots can be actuated using different methods, such as light, electricity, heat, chemicals, and magnetism. However, among these methods, magnetic actuation stands out due to its unique characteristics. One of the key advantages of magnetic actuation is its biocompatibility, which makes it a suitable method for medical applications. This is because it can interact with biological tissues without causing physical damage or altering DNA. It can also penetrate varied tissues and surfaces, providing precise remote control. The potential applications of these robots can be increased by integrating the soft materials. The untethered soft robots have more advantages than the rigid ones, as they can adapt their body shape to fit tight spaces and navigate paths with high curvatures [23]. As there is no direct physical attachment to an external point, they are able to access remote and hard to reach parts of the body for targeted drug delivery [17, 16, 15] and

minimally invasive surgeries [19, 18].

Designing these untethered small scale magnetic soft robots is a precise and complex process that involves optimizing many parameters. In recent years, many studies have been done on the use of artificial intelligence (AI) to assist and ease this process [5, 6, 24]. These studies have shown the promising possibility of using AI, especially evolutionary algorithms, in designing and optimizing small scale magnetic robots. This optimization can focus on different aspects of the robot, such as magnetization profile and material distribution. Despite the success of the mentioned studies, the objective functions that are used in these studies are based on some physical modeling that cannot represent all aspects of the complex behavior of these robots in real world environments and has limited the effectiveness of the algorithms in evolving more sophisticated designs.

To address the mentioned gap, this study integrates Covariance Matrix Adaptation Evolution Strategy (CMA-ES)[25], which is an evolutionary algorithm, with a Material Point Method (MPM) based simulation framework designed for magnetic soft robots [26] to optimize an untethered small scale magnetic soft robot to increase its walking speed. Using this MPM-based simulation as part of the fitness function enables the evolution of complex robots with more sophisticated behavior without the need for advanced physical modeling.

This study goes beyond mere optimization and can be considered a step towards automated design. The algorithm designs the magnetic configuration of the robot to have desirable movement, and the resulting robot demonstrates movement similar to the human-designed robot, despite having a significantly different magnetic configuration. The algorithm successfully achieves novel designs and significantly increases

the speed of the robot by 45.5%.

This study demonstrates the potential of combining CMA-ES with an MPM-based simulation, provides a framework for optimizing any segment-base robots that can be modeled within this simulation environment, and opens the door for future research that aims for evolving more sophisticated and complex microscale magnetic soft robots for diverse applications.

Chapter 2

Background

This chapter provides an overview of the fundamental concepts involved in this study. It begins with an introduction to soft robotics and highlights its significance and applications while providing details on the materials and fabrication involved. Then, the principle of magnetic actuation is explained to provide the foundational knowledge that is necessary for understanding the following sections. The chapter concludes with an introduction to evolutionary algorithms and their role in optimizing small scale soft magnetic robots.

2.1 Introduction to Soft Robotics

Soft robotics is a growing field that focuses on creating flexible and adaptable robots inspired by biological organisms [27]. These robots are characterized by their soft body made of flexible materials, allowing them to bend and deform to adapt to the environment and have better maneuverability. These characteristics give soft robots significant advantages over rigid ones, especially in circumstances where adaptability

and safe interaction are essential. Various actuation methods, such as light [28], electricity [29], heat [30], chemical [31] and magnetism [32–35], are used in soft robotics. Among these methods, magnetic actuation stands out due to several advantages. Firstly, it is wireless and can navigate through air, vacuum, and fluids, which eliminates the need for a direct physical connection, leading to the improvement of the robot’s mobility, an essential feature of soft robots. Secondly, it is safe, biocompatible, and quick in response, making it a desirable method for medical applications [36, 15–17]. Magnetic soft robots are widely used in various fields, such as small-scale manipulation [37], microfluidics [38, 11], and intelligent sensors [39]. These robots are moved by a magnetic field, which can generate impressive forces compared to their size. This, combined with the fact that they do not require a motor for movement, allows them to be scaled down to just a few nanometers to micrometers, which makes them ideal for operation with high precision.

Despite these advantages, magnetic actuation has limitations that are worth mentioning. The magnetic field strength decreases rapidly with distance, which can limit the depth of the operation, especially inside the body. Additionally, the magnetic field tends to interfere with surrounding materials and is affected by magnetic noise, which can lead to inaccuracy. In terms of safety, the magnetic field should be used with greater caution in individuals with cardiac pacemakers and electrically active implants since it can cause serious health issues [40, 41].

2.1.1 Magnetic Soft Materials for Soft Robotics

Small scale magnetic soft robots are constructed using magnetic soft materials, which consist of two components: the soft material that provides the flexible base and the

magnetic component that provides responses to the magnetic actuator [1]. These materials are categorized into discrete and continuous systems based on their structure. In discrete systems, the soft material is incorporated with one or multiple rigid magnets. As a result, the magnetic moment of this system is focused. On the contrary, the continuous system is composed of a soft matrix embedded with magnetic nano- or microparticles that are distributed throughout the material. This provides a distributed magnetic moment that can lead to more smooth deformations.

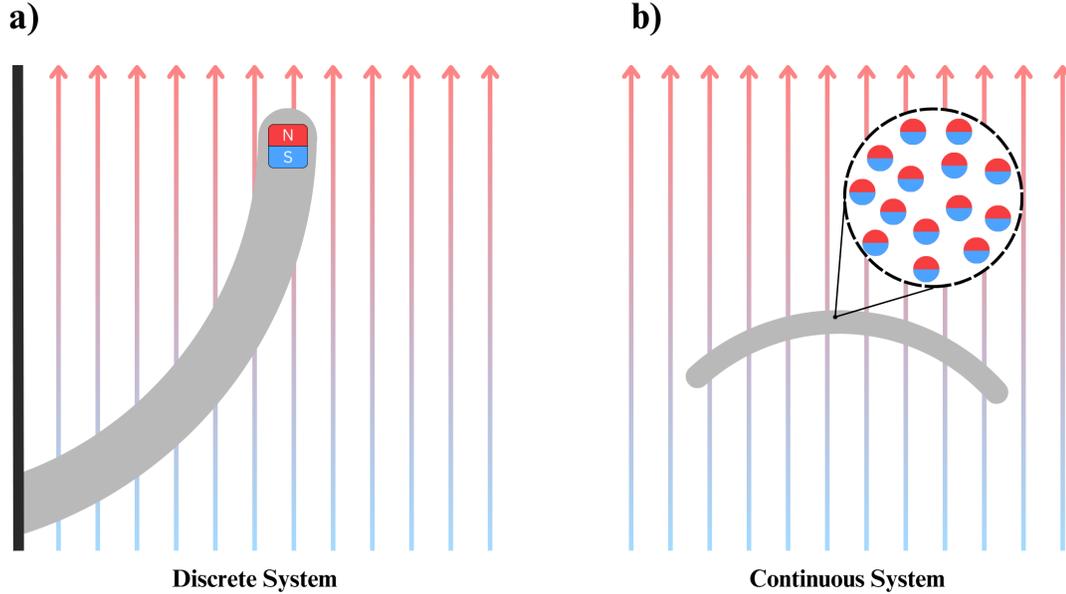


Figure 2.1: Types of Magnetic Soft Materials. Magnetic soft materials can be categorized into a) Discrete Systems, which have rigid magnets inside a flexible structure, and b) Continuous Systems, which have micro/nano magnetic particles in a flexible matrix.

When we refer to magnetic materials, we primarily focus on the ferromagnetic and ferrimagnetic materials, which are strongly responsive to external magnetic fields. The magnetic properties of these materials can be indicated using a magnetization curve. This curve provides important information about magnetic material behavior,

including remanence (M_r), coercivity (H_c), and saturation magnetization (M_s). The remanence (M_r) indicates the magnetic moment that remains in the material after being completely magnetized in the absence of an external magnetic field. Coercivity (H_c) is the strength of the magnetic field that is needed to demagnetize the material, and saturation magnetization (M_s) is the maximum magnetization that a material can have. Based on these properties, magnetic materials fall into two main categories: soft-magnetic materials and hard-magnetic materials. As shown in Fig. 2.1.b, soft-magnetic material exhibits high saturation magnetization (M_s) but has low coercivity (H_c), low remanence (M_r), and a small hysteresis loop. These characteristics show that soft-magnetic material can be easily magnetized and demagnetized with a small external magnetic field. While they can act as a strong magnet in the presence of an external field, their low remanence (M_r) results in a weak and negligible magnetic moment when the external magnetic field is removed. On the other hand, hard-magnetic materials (Fig. 2.2.a) have high coercivity (H_c) and high remanence (M_r), along with a large hysteresis loop. This indicates that they retain a strong magnetic moment even in the absence of an external magnetic field. Additionally, a stronger magnetic field is required to magnetize and demagnetize the materials that are usually used in permanent magnets.

2.1.2 Fabrication of Magnetic Soft Robots

Both soft and hard magnetic materials can be used as embedded particles within a soft matrix to create magnetically responsive soft robots. However, their fabrication processes are different. When using soft magnetic particles, the particles are mixed with a soft matrix and exposed to an external magnetic field while the material

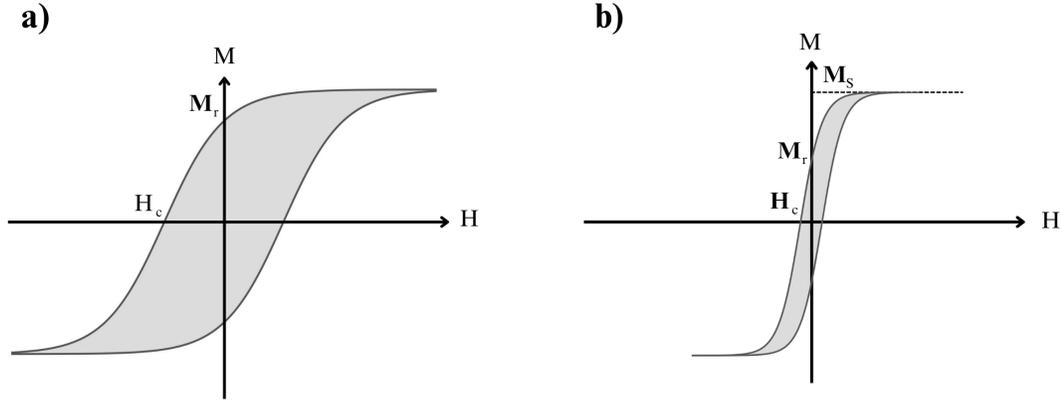


Figure 2.2: Magnetic Hysteresis of Hard and Soft Magnetic Materials. a) Hard magnetic materials show high coercivity (H_c) and remanence (M_r) while b) soft magnetic materials exhibit low coercivity (H_c) and remanence (M_r) with a narrow hysteresis loop. Figure recreated based on [1].

is still in a liquid state. Under the influence of this field, the particles align with the external magnetic field direction and preserve their arrangement after the soft matrix is cured. Although the particles lose most of their magnetization when the external magnetic field is removed, their fixed alignment in the cured matrix enables controlled deformation by magnetic actuators. When hard magnetic particles are used, the particles are mixed with the soft matrix and magnetized after the matrix is cured. Alternatively, it is also possible to magnetize the particles before mixing them in the soft matrix and reorient them again before curing to produce desirable patterns. Both of these methods lead to soft robots with strong magnetization in a specific direction. The latter method provides more control on creating magnetic patterns and needs a weaker magnetic field in the final step compared to the former. This study focuses on soft magnetic robots constructed by embedding hard magnetic

particles within a soft matrix. From this point, the term 'soft magnetic robots' will specifically refer to these structures.

2.2 Magnetic Actuation and Fundamental Principles

The movement of electrical charges, whether as a current in a wire or as free electrons in a magnetic material, can lead to the creation of a magnetic field [42]. Magnetic fields are vector fields that vary across space. In magnetic robots, the focus is on the magnetic fields projected into space from a source that can apply force and torque on magnetic objects to actuate them. By observing a magnetic field source from a far distance, it can be considered a dipole. Magnetic dipoles are similar to electric dipoles, but unlike electric charges, magnetic monopoles have not been observed in nature. By considering the source of the magnetic field as a dipole, the complex and subtle details of the field structure can be simplified, allowing it to be represented as a point with both magnitude and direction. The magnetic field \mathbf{B} that a dipole has is represented by Eq. (2.2.1) where \mathbf{r} is the distance from the central dipole, \mathbf{m} is the magnetic moment of the dipole and μ_0 represents the vacuum magnetic permeability.

$$\mathbf{B}(\mathbf{r}) = \frac{\mu_0}{4\pi} \left[\frac{3\mathbf{r}(\mathbf{m} \cdot \mathbf{r})}{r^5} - \frac{\mathbf{m}}{r^3} \right] \quad (2.2.1)$$

The magnetic field can apply magnetic torque and force on an object with a magnetic moment, which are the fundamental principles of magnetic actuation. By applying a magnetic field, the resulting magnetic torque and force can be used to manipulate the magnetic object remotely with high precision.

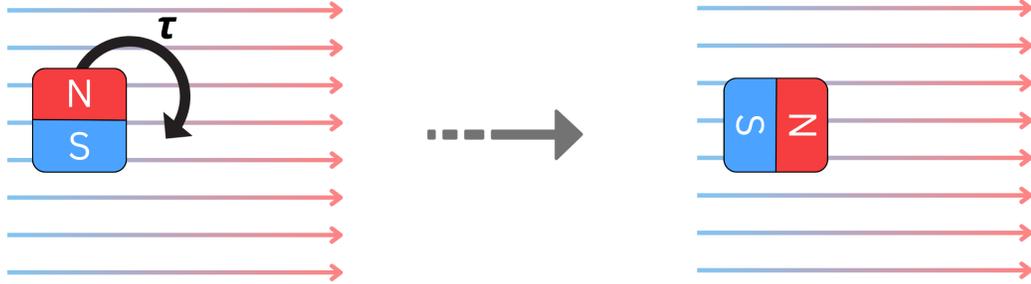


Figure 2.3: Magnetic torque applied on a dipole in the presence of an external magnetic field.

The torque $\boldsymbol{\tau}$ on a magnetic dipole \boldsymbol{m} due to an external magnetic field \boldsymbol{B} is given by Eq. (2.2.2).

$$\boldsymbol{\tau}_m = \boldsymbol{m} \times \boldsymbol{B} \quad (2.2.2)$$

Since magnetic torque is given by the cross product of the external magnetic field and magnetic moment of the object, the maximum torque is generated when they are perpendicular, while torque is zero when they are parallel. Fig. 2.3 illustrates the rotation of a magnet to align with the external magnetic field due to the applied magnetic torque.

If the external magnetic field is nonuniform, a magnetic force is applied to the magnetic object exposed to it. A nonuniform magnetic field varies across space and creates a spatial gradient. When a magnetic object is placed in such a field, a magnetic force is generated that drives it from a region with a weaker magnetic field toward the region with a stronger magnetic field. The magnetic field generated by a source weakens rapidly as the distance from the source increases. As a result, the magnetic object gets attracted to the magnetic field source. The magnetic force \boldsymbol{F} on

a magnetic dipole \mathbf{m} when it is placed in a nonuniform magnetic field \mathbf{B} is calculated by Eq. (2.2.3)

$$\mathbf{F}_m = (\mathbf{m} \cdot \nabla)\mathbf{B} = \begin{bmatrix} \frac{\partial B_x}{\partial x} & \frac{\partial B_x}{\partial y} & \frac{\partial B_x}{\partial z} \\ \frac{\partial B_y}{\partial x} & \frac{\partial B_y}{\partial y} & \frac{\partial B_y}{\partial z} \\ \frac{\partial B_z}{\partial x} & \frac{\partial B_z}{\partial y} & -\left(\frac{\partial B_x}{\partial x} + \frac{\partial B_y}{\partial y}\right) \end{bmatrix} \begin{bmatrix} m_x \\ m_y \\ m_z \end{bmatrix} \quad (2.2.3)$$

A magnetic object can simultaneously experience both magnetic torque and magnetic force. Fig. 2.4 illustrates a magnet placed in a nonuniform magnetic field where a magnetic force is exerted on it. This moves the object toward the region with a stronger magnetic field. In this case, the magnet is parallel to the external magnetic field, which results in zero torque.

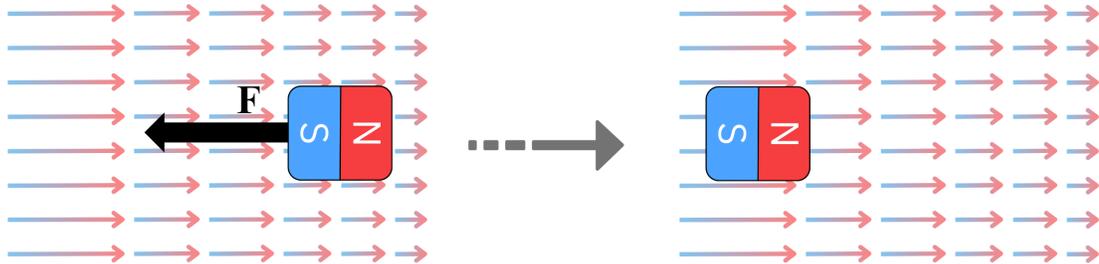


Figure 2.4: Magnetic force applied on a magnetic dipole in the presence of a nonuniform magnetic field

Soft magnetic robots embedded with hard magnetic particles operate using the same fundamental magnetic principles, which allow them to deform under applied magnetic force and torque. This deformation is achieved through the controlled alignment and distribution of magnetic particles within the soft matrix, enabling bending and twisting. Utilizing hard magnetic particles allows different parts of the robot

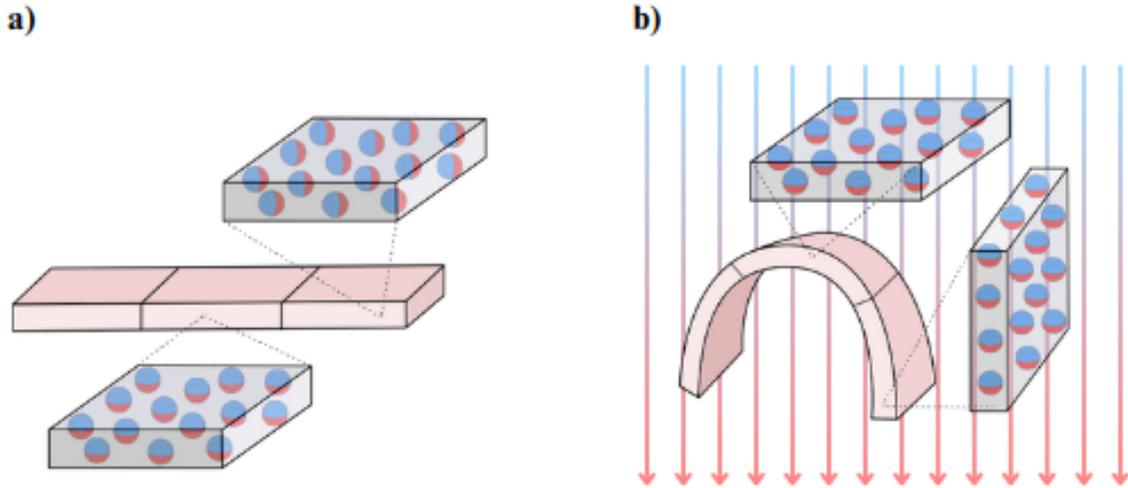


Figure 2.5: Deformation of a magnetic soft robot under a uniform magnetic field. a) The robot remains in its natural position when no external magnetic field is present. b) In a uniform magnetic field, the magnetic particles of each segment experience torque and align with the external field which leads to the deformation of the robot.

to have distinct magnetization alignments, proving more sophisticated and precise control over the deformation.

Fig. 2.5 illustrates a soft magnetic robot with varying magnetization alignment along its body. When it is not exposed to an external magnetic field, it is in its natural shape as seen in Fig. 2.5a. When placed in a uniform magnetic field, the magnetic particles experience a torque, causing it to align with the external magnetic field. As the particles are magnetized segment by segment, each segment responds differently to the external magnetic field based on its magnetic moments. This leads to the deformation of the robot as illustrated in Fig. 2.5b.

The deformation of these robots is dependent on many variables, such as the mechanical properties of the soft material, the shape of the robot, and the distribution

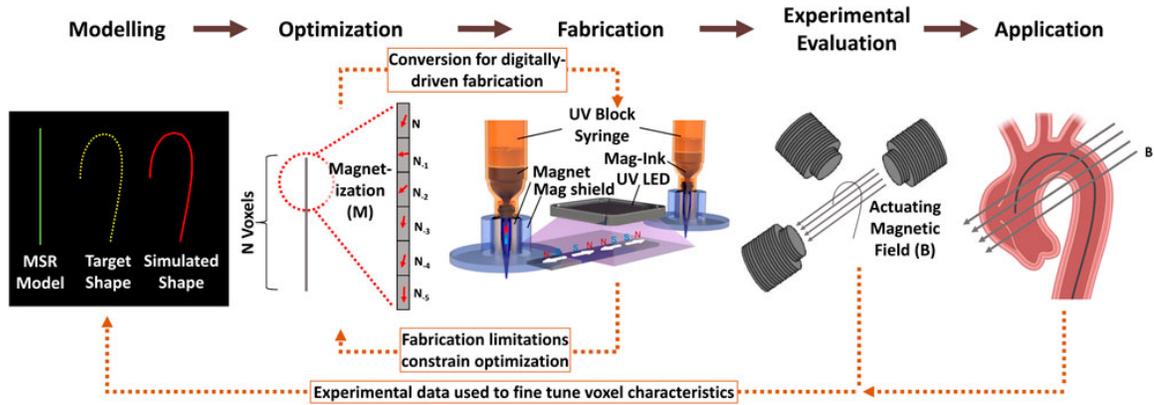


Figure 2.6: The voxel-encoding and development process of magnetic soft robots. Reproduced from [2] under the Creative Commons Attribution (CC BY) license. "Fabrication" subfigure taken from [3]

of the particles. Physical modeling of these robots' behavior is complex and goes far beyond the provided magnetic principles. To address it, this study combines an MPM-based simulation environment with an evolutionary algorithm to optimize robot performance without relying on explicit physical modeling.

2.3 Optimization of Programmable Domains in Soft Magnetic Robots

Soft robots embedded with hard magnetic particles can achieve precise and sophisticated deformation as their physical properties are derived from the soft matrix and the magnetic density and profile of different regions can be adjusted [2]. The ability to adjust the magnetic profile of different regions enables a wide range of design possibilities, but it increases the design and fabrication complexity. The detailed design workflow for such robots has been illustrated in Fig. 2.6. This figure demonstrates the design process of a soft magnetic robot for cardiac catheterization. The process

is initiated by modeling the robot, which is the foundation for the next step. In the optimization phase, the model is used alongside the desired target shape or application to determine the characteristic magnetization of each segment. The optimized robot can be fabricated in the next phase to be examined in a real world setting. The driven data can be used to further tune the model and repeat the process [2].

This research focuses on the optimization phase of this process, which can have a significant impact on the final result. As we proceed, we delve deeper into optimization techniques and their use in the field of magnetically actuated soft robots [2].

2.3.1 Optimization Using Physics-Based Modeling

Numerical, analytical, and data-driven methods can be used to produce physics-based modeling [2]. The modeling predicts the behaviour of the robot under changing magnetic fields. Using this model, control variables can be optimized to fine-tune the design and achieve desirable deformation [43–46].

Physics-based modeling comes with several disadvantages. The computational cost can be high, especially when dealing with high-resolution voxelization. Furthermore, this approach often uses simplification through static or quasi-static assumptions, which prevents the model from considering the robots' dynamic behavior [2].

2.3.2 Optimization Using deep learning algorithms

Deep learning algorithms can be used for optimization by identifying complex patterns from large data sets [47]. By training these algorithms on appropriate data sets, models can be developed to determine the magnetic configuration of magnetic soft robots to achieve desirable deformations [2].

Despite their power and popularity, these algorithms are not widely used in this field due to several disadvantages. These algorithms rely on large data sets that are difficult and costly to obtain in the magnetic soft robotics field. Data needs to represent the robot with high accuracy in various conditions that are challenging to achieve in both real-world and simulation environments, especially when the number of repetitions increases [2].

Despite these challenges, a few studies have used deep learning optimization with successful results. Lloyd et al. used an artificial neural network (ANN) to determine the magnetization of a soft continuum robot to achieve desirable deformations. The ANN was trained on a data set they built based on various magnetization and their corresponding deformations. The data was produced using a FEM modeling they introduced, where thousands of different magnetizations were fed to the FEM model to obtain the deformation responses. The trained ANN model was able to reverse the design process by getting the desired shape and producing the required magnetization profiles [48].

2.3.3 Optimization Using Evolutionary algorithms

Evolutionary algorithms are powerful optimization tools inspired by natural selection. These algorithms are able to search large solution spaces with multiple local optima, and unlike traditional gradient-based methods, they do not require derivative information. These algorithms operate based on stochastic selection and evolve selected solutions repetitively by mutation and recombination to gradually improve them [49].

In recent years, evolutionary algorithms have found their place in the optimization of magnetic soft robots. These algorithms can be used to optimize the magnetic

profile or rigidity pattern of the robot to achieve a higher workspace, the range of positions that the robot can physically reach, or a specific shape deformation. By widely exploring different configuration settings, the algorithms can develop robots that are specifically adapted to their intended tasks and capable of high-precision operation.

As this study heavily relies on evolutionary algorithms, we will explore this method in greater depth in the following section. Additionally, a separate section (Section 2.4.3) will be devoted to reviewing existing research and advancements in applying evolutionary algorithms for the optimization of magnetic soft robots.

2.4 Introduction to Evolutionary Algorithms

Evolutionary algorithms are a group of algorithms inspired by Darwin’s theory of evolution in nature. In this group of algorithms, the potential solutions are considered as individuals that evolve through generations. These algorithms consist of several steps that help them to mimic evolution in nature and gradually evolve solutions in a cyclical process. The process begins with a group of random individuals that form the initial generation. Each of these individuals gets evaluated and assigned a fitness value based on a fitness function in the evaluation phase. Next, some of these individuals are chosen based on their fitness value as the parents of the next generation, and they get combined together in a process called crossover and slightly changed in the process called mutation. These operations form new individuals, which are called the offspring. These offspring are evaluated and then compete with the existing individuals based on their fitness value, and in some cases age, to be selected to form the next generation. This cycle continues until an individual (solution) with

a specific fitness is achieved or for a set number of generations [4]. The flowchart of this cyclical process is shown in Fig. 2.7.

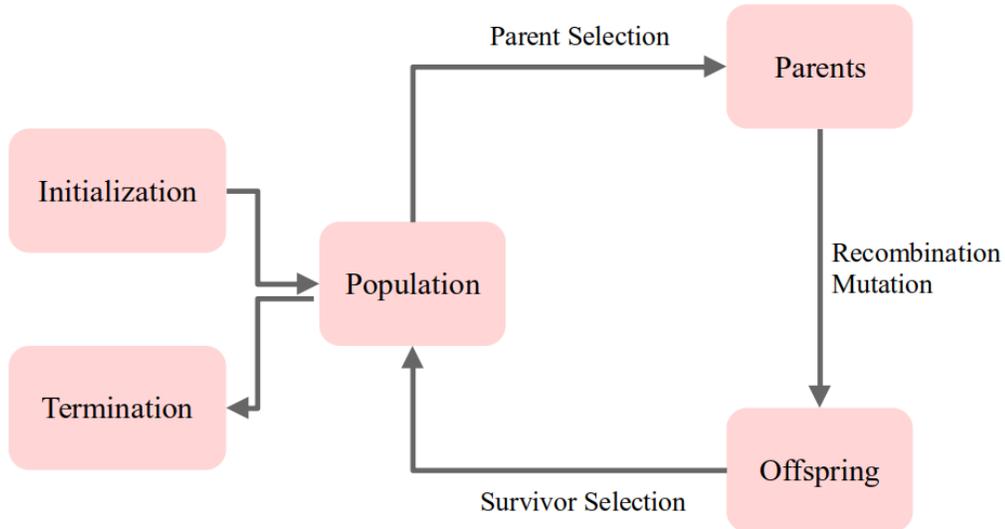


Figure 2.7: Flowchart of evolutionary algorithms process. The process involves initializing a population, selecting parents, applying crossover and mutation to generate new offspring, and selecting survivors and updating the population based on that. This cycle continues for a set number of generations or until a specific condition, such as reaching an optimal solution, is met. Figure recreated based on [4].

These algorithms are considered stochastic, as many parts of their processes rely on randomness. The initial population is often composed of random individuals. Selection is not solely based on fitness alone, and even less fit individuals have the chance of being selected. In recombination, it is randomness that determines which part of each parent contributes to form the offspring. Similarly, in mutation, the specific parts of the individual that undergo alteration are selected randomly.

Evolutionary algorithms act based on two factors: variation operators and selection. Variation operators include mutation and recombination, which cause the diversity among solutions and increase the novelty. Conversely, the selection increases the quality of the individuals by favouring the ones with higher fitness values. The gradual improvement of fitness value through generations is the result of these two features [4].

The process of these algorithms can be seen from two different perspectives. One perspective considers the process as a fitness function optimization, where values are iteratively refined to get closer to the optimal solution point gradually. On the other one, the process is considered as an adaptation problem. In this perspective, fitness is not considered a function to be optimized but considered an expression of the environmental condition, and the individuals that are more adapted to the environmental conditions are more likely to have more offspring, which gradually leads to populations that are generally more adapted to the environment [4].

Over time, different dialects of evolutionary algorithms have developed to address the diverse range of problems [4]. These include Genetic Algorithms (GA), Evolution Strategies (ES), Evolutionary Programming (EP), Genetic Programming (GP), Learning Classifier Systems (LCS), Differential Evolution (DE), Particle Swarm Optimization (PSO), and Estimation of Distribution Algorithms (EDA). Each of these has its unique representation for solutions and its unique approach to selection, reproduction, and mutation, which makes them suitable for solving specific problems. However, this study has a focus on Genetic Algorithms (GA) and Evolution Strategies (ES), and other dialects fall out of the scope of this research and will not be discussed further.

2.4.1 Genetic Algorithm

The genetic algorithm is the most well-known dialect of evolutionary algorithms and shows the most resemblance to the natural evolution process [50–52]. It was introduced by John Holland in the 1960s and early 1970s [53]. Around the same time, De Jong implemented it and used it for parameter optimization [54, 55]. This algorithm relies on both mutation and recombination to evolve solutions. The solutions are represented as a string of fixed length that traditionally was binary-encoded, but alternative encoding methods are also used, such as Octal, Hexadecimal, three, permutation, and value encodings [56–58].

Although the genetic algorithm was first introduced to study adaptive behavior, it is widely used as a powerful optimization tool. One of the earliest versions of GA is the Simple Genetic Algorithm (SGA), which is a process with fixed steps. It starts with a population of size μ , from which a sub-population of parents is created that has the same size (μ) but allows duplicates. These individuals pair randomly and combine with each other through crossover to create offspring. The offspring replace the parent population, and the cycle continues. This algorithm is based on a binary representation and fitness proportionate selection. The mutation rate is low, and the production of the new generation heavily relies on the crossover, while mutation has a low probability. This algorithm is summarized in table 2.1.

Representation	Bit-strings
Recombination	1-Point crossover
Mutation	Bit flip
Parent selection	Fitness proportional - implemented by Roulette Wheel
Survival selection	Generational

Table 2.1: Summary of Simple Genetic Algorithm (SGA) components. Reproduced from [4].

2.4.2 Evolution Strategy

In the early 1960s, Rechenberg and Schwefel introduced the evolutionary strategies (ES) while they were working on a shape optimization problem. ES is typically used for continuous optimization, where each solution is represented as a vector of real-valued parameters (see 2.2). $(1 + 1)$ ES is the earliest ES algorithm consisting of two members. This algorithm works with one individual, called parent, which produces one offspring. The offspring is created by adding a different random number to each element of the parent, and it replaces the parent if it has a higher fitness value. This ensures that the solution (individual) is always moving towards a higher fitness value in the search space. By removing this condition and replacing the parent with the offspring, whether it has a higher fitness or not, we reach the $(1, 1)$ ES algorithm. In this case, the individuals forget the previous solutions, their ancestors. The random numbers that are added to the elements of the parents are sampled from a Gaussian distribution with a mean of zero and a deviation equal to σ that is considered the mutation rate [4].

Representation	Real-valued vectors
Recombination	Discrete or intermediary
Mutation	Gaussian perturbation
Parent selection	Uniform random
Survivor selection	Deterministic elitist replacement by (μ, λ) or $(\mu + \lambda)$
Speciality	Self-adaptation of mutation step sizes

Table 2.2: Summary of Evolution Strategy (ES) components. Reproduced from [4].

Having more than one individual leads to the development of $(\mu + \lambda)$ and (μ, λ) ES's in the 1970s. In this case, μ determines the size of the population, and λ is the number of offspring that is produced in each generation. Following this, the concept

of self-adaption was introduced to enhance the effectiveness of the search process. To enable this, the ES's parameters are included in the individuals, allowing dynamic adaption through the optimization process. Nowadays, the mutation step size in most ESs is self-adaptive to provide flexibility and efficiency. One of the most significant algorithms in this area is Covariance Matrix Adaptation (CMA) [25, 4]

CMA introduces a more flexible search strategy, which makes it a successful approach for optimizing synthetic and real-world black-box problems [59, 58]. This algorithm acts based on adapting mean vector, step size, and covariance matrix to refine the search distribution through generation. The mean vector represents the center of search distribution and is updated in each generation based on the fittest offspring generated in the current generation. This leads to the shift of the distribution towards more promising areas of the search space. The step size determines the scale of exploration in the search space. By adapting this parameter, CMA-ES makes a balance between exploration and exploitation. One of the key differences between this algorithm and other ESs is the full covariance matrix update. This allows search distribution to align with problem structure and properly update the distribution in complex and non-separable problems.

Recombination in Evolution Strategies

The recombination process, which combines the parents to produce the offspring, can be done in two ways based on the selection of value for each gene: Discrete Recombination and Intermediate Recombination. In Discrete Recombination, the value of each gene is selected randomly from one of the parents to transfer to the offspring. On the contrary, in the Intermediate Recombination, the value of each gene is obtained

by averaging the corresponding genes of the parents. The number of contributing parents for each offspring can be more than the traditional two parents. This recombination is referred to as global recombination, whereas the two-parent approach is called local recombination [4]. Global recombination is commonly used in ESs, while both discrete and intermediate recombination are applied simultaneously to produce offspring. The part of individuals that is storing ES's parameters undergoes intermediate combination while the other part undergoes discrete recombination. This allows the algorithm to produce diverse solutions while keeping the average mutation step size.

2.4.3 Review of Evolutionary Algorithm Applications in Magnetic Soft Robot Optimization

In recent years, evolutionary algorithms have found their place as a powerful optimization tool for the small scale magnetic robot design. Evolutionary algorithms, unlike traditional methods, do not rely on gradient information and are capable of searching a large and complex design space efficiently. Their population-based structure allows for the exploration of diverse solutions, which makes them a suitable approach for optimizing small scale magnetic robots with complex and non-linear design characteristics. In this section, we cover key studies that have applied evolutionary algorithms for optimization of small scale robot.

Evolutionary design of magnetic soft continuum robots by Liu Wang et al. [5]

Wang et al. introduced an evolutionary approach for optimization of magnetic soft continuum robots (MSCRs) to increase their workspace. Traditional MSCRs consist of permanent magnets or a uniform distribution of magnetic particles, which limits the range of workspace of these robots. To overcome this problem, Wang et al. used genetic algorithm to optimize both magnetization and rigidity pattern of the robot to have more flexible and adaptable robots that leads to their larger workspace.

The robot is divided into 100 discrete voxels, and a magnetic particle volume fraction (ϕ) is assigned to each of these voxels as seen in Fig. 2.8. The (ϕ) value affects both remanent magnetization and rigidity. The remanent magnetization is obtained from Eq. (2.4.1), where M_0 is the remanent magnetization of the hard-magnetic particles. By increasing the ϕ , the remanent magnetization of the voxel also increases. Consequently, ϕ only determines the magnitude of the magnetization, while the direction of magnetization is considered the same for all voxels, pointing toward the tip of MSCR.

$$M = M_0\phi \tag{2.4.1}$$

Increasing ϕ also affects the rigidity of voxels. As magnetic particles are stiffer than the polymer matrix, increasing ϕ leads to an increased shear modulus, which can be obtained from Eq. (2.4.2), where G_0 is the shear modulus of the polymer matrix. It is also worth mentioning that in this study, ϕ is constrained between 0 and 0.4 as voxels with ϕ bigger than 0.4 cannot be fabricated properly.

$$G(\phi) = G_0 \exp \left[\frac{2.5\phi}{1 - 1.35\phi} \right] \tag{2.4.2}$$

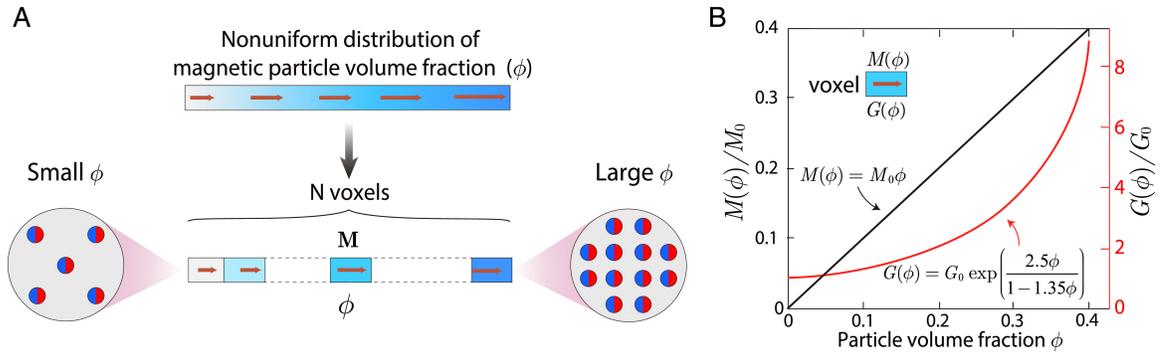


Figure 2.8: Programmable MSCR with varied remanent magnetization and rigidity along its body. a) Each voxel of the robot has a specific remanent magnetization that is set by adjusting magnetic particle volume fraction (ϕ). b) The relation between magnetic particle volume fraction (ϕ) and normalized magnetization strength and shear modulus is shown in black and red, respectively. Reproduced from [5] under the PNAS license.

To measure the workspace as the fitness value of the genetic algorithm, they developed a hard-magnetic elastica theory, which predicts the deformation of the robot with a given magnetization and rigidity pattern under the influence of a specific external magnetic field. As the workspace is asymmetric, they only analyzed half of the workspace for their study. The algorithm starts with 100 randomly generated individuals represented as an array of ϕ 's. The half workspace of these individuals is calculated using the developed model. Then, they choose 100 individuals based on stochastic universal sampling, which allows duplication, and individuals with higher fitness are more likely to be selected. The next generation is created by 5% elitism, 10% mutation, and 85% crossover. This means that 5% of the new generation is built by choosing the individuals with the highest fitness of a selected group, 10% of the new generation is built from randomly selected individuals from a selected group that has undergone mutation, and finally the remaining 85% of the new generation is built by random selection from the selected group and applying crossover on them. The

cycle continues until the difference between the workspace of the fittest individual and the mean of the workspaces of all individuals in a generation is smaller than a predefined threshold. This optimization process is demonstrated in Fig. 2.9.

Using a genetic algorithm, they successfully optimized an MSCR robot with non-uniform magnetization distribution and achieved a significantly larger workspace. They also validate their approach through finite element simulations and experimental testing and illustrate the strength of the evolutionary algorithm in the optimization of magnetic soft continuum robots.

Evolutionary Algorithm-Guided Voxel-Encoding Printing of Functional Hard-Magnetic Soft Active Materials by Wu et al. [6]

. This study introduces a voxel-encoding (Direct Ink Writing) DIW printing strategy that provides control on the direction and density of magnetization of voxels in the printing process. This flexibility is achieved by controlling the path of printing and the number of printing layers; as the number of layers increases, the magnetization density of the voxel increases.

By having m voxels and n layers per voxel, the search space is equal to $(2n + 1)^m$, which can grow significantly as the number of voxels and/or layers increases. This indicates the need for an efficient optimization tool to find an optimal design among a large number of possible solutions. As a result, Wu et al. introduced an evolutionary algorithm that is capable of finding the optimum magnetization of different beams' voxels, among possible configurations, to achieve desired deformations when they are exposed to specific external magnetic fields. The algorithm begins with an initial population of 32 randomly generated individuals ($\mu = 32$), where each individual

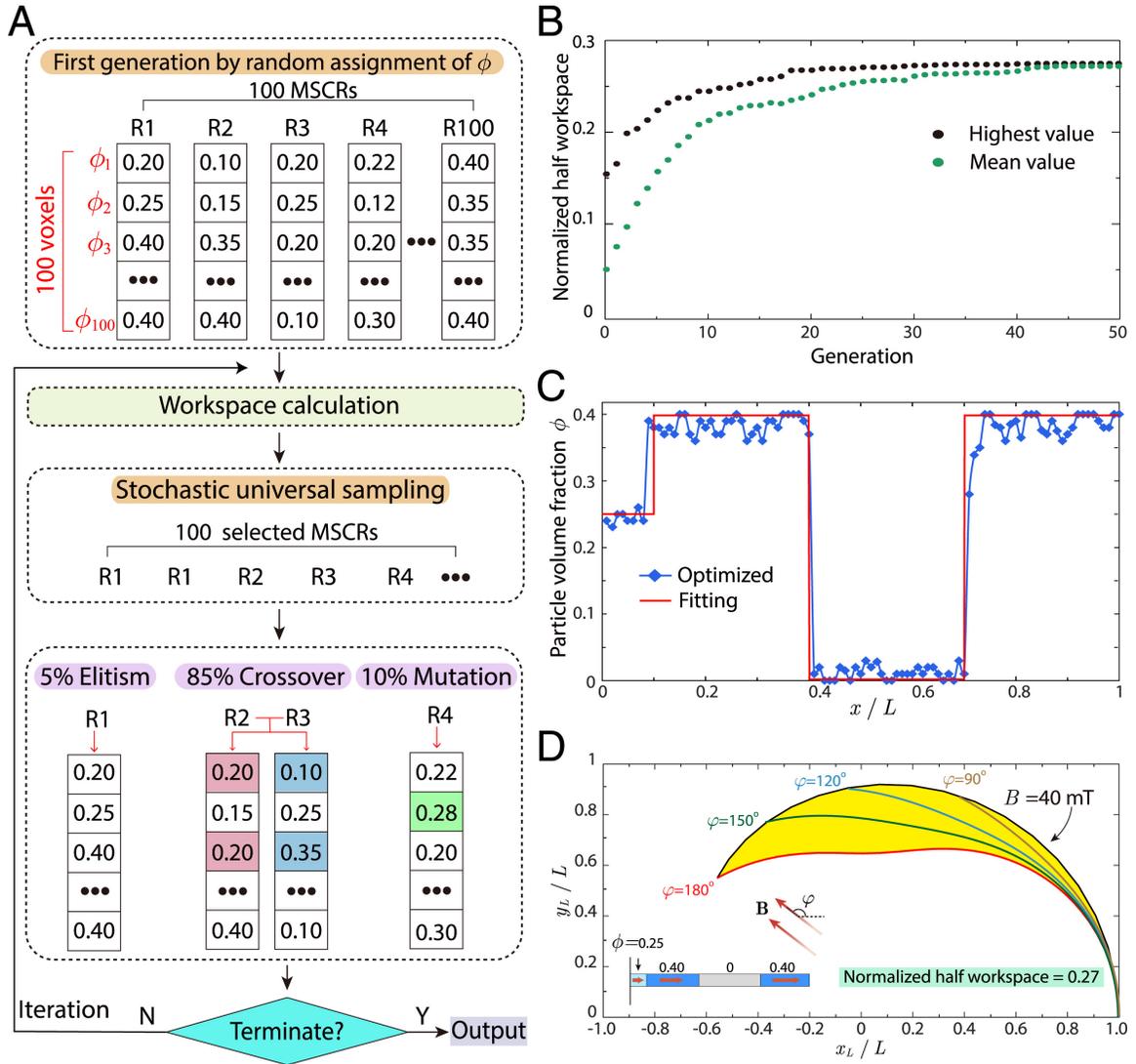


Figure 2.9: a) The flowchart of the optimization algorithm, illustrating the representation of individuals in the initial population, stochastic universal sampling, crossover, and mutation. b) The highest and mean normalized half workspace of 100 individuals plotted over 50 generations. c) Magnetic distribution and fitted function of the optimized design at generation 40. d) Normalized half workspace of the robot based on the fitted distribution of the optimized design in generation 40.

Reproduced from [5] under the PNAS license.

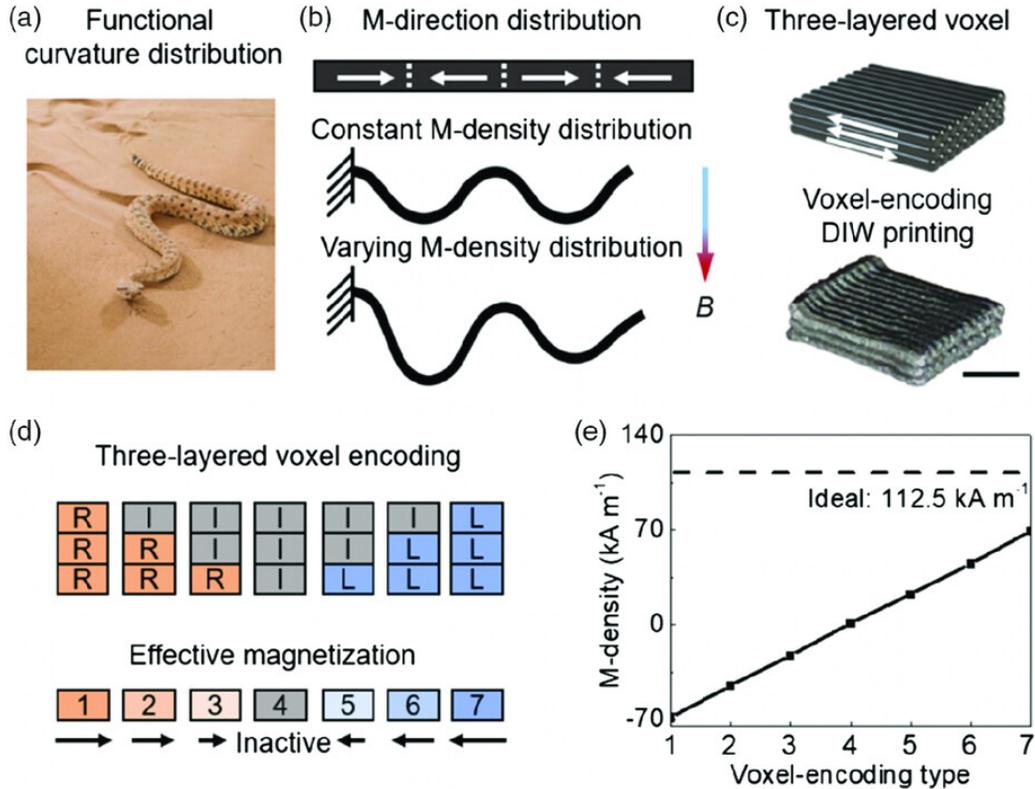


Figure 2.10: Voxel-Encoding Printing of Hard-Magnetic Soft Active Materials. a) special curvature of a Sidewinder snake, which leads to its unique sidewinder motion. b) The structure of a three-layered printed voxel with programmed density and direction. d) The encoding of voxels with three printed layers. e) Magnetic density of voxels with a varied number of layers. The dashed line represents the ideal magnetic density for this type of robot. Reproduced from [6] under the Creative Commons Attribution License (CC BY 4.0).

represents a unique magnetization pattern for the beam. In this representation, each element of individuals corresponds to a single voxel that determines the magnetization density and direction. The range of acceptable values for each voxel is determined based on the number of layers (n). For example, having $n = 3$, the variations can be encoded by $2n + 1$, which is equal to 7 possible magnetization states. The encoded values are stored in the individuals and are decoded when it is needed. Each individual is evaluated by comparing the deformation of the decoded individuals in the FEM

simulation to the desired deformation fed to the algorithm in the first step. In the next step, 38 individuals ($\lambda = 38$) are selected for the next generation by crossover and mutation operators. 70% of the new generation undergoes crossover, and 25% undergoes mutation, while the remaining 5% of the population does not experience either. This cycle continues until the fitness value is smaller than a predefined value or for a set number of generations, here 15 generations. This process is illustrated in Fig. 2.11.

Conclusion. The reviewed studies demonstrate the power of evolutionary algorithms in optimizing magnetic soft robots, especially through control over the material properties and magnetization pattern across segmented structures [5, 6]. However, a common limitation in both works is the focus on the static shape outcome under fixed magnetic field conditions. The optimization target is a specific final deformation rather than the robot’s complete dynamic behavior in an environment. This provides an opportunity for future research to go beyond shape-matching objectives toward behavior-driven fitness functions that reflect the robots’ behavior in real tasks. In this study, we address this gap by utilizing an MPM-based simulation environment as part of the fitness function to increase the robot’s walking speed and overall locomotion performance.

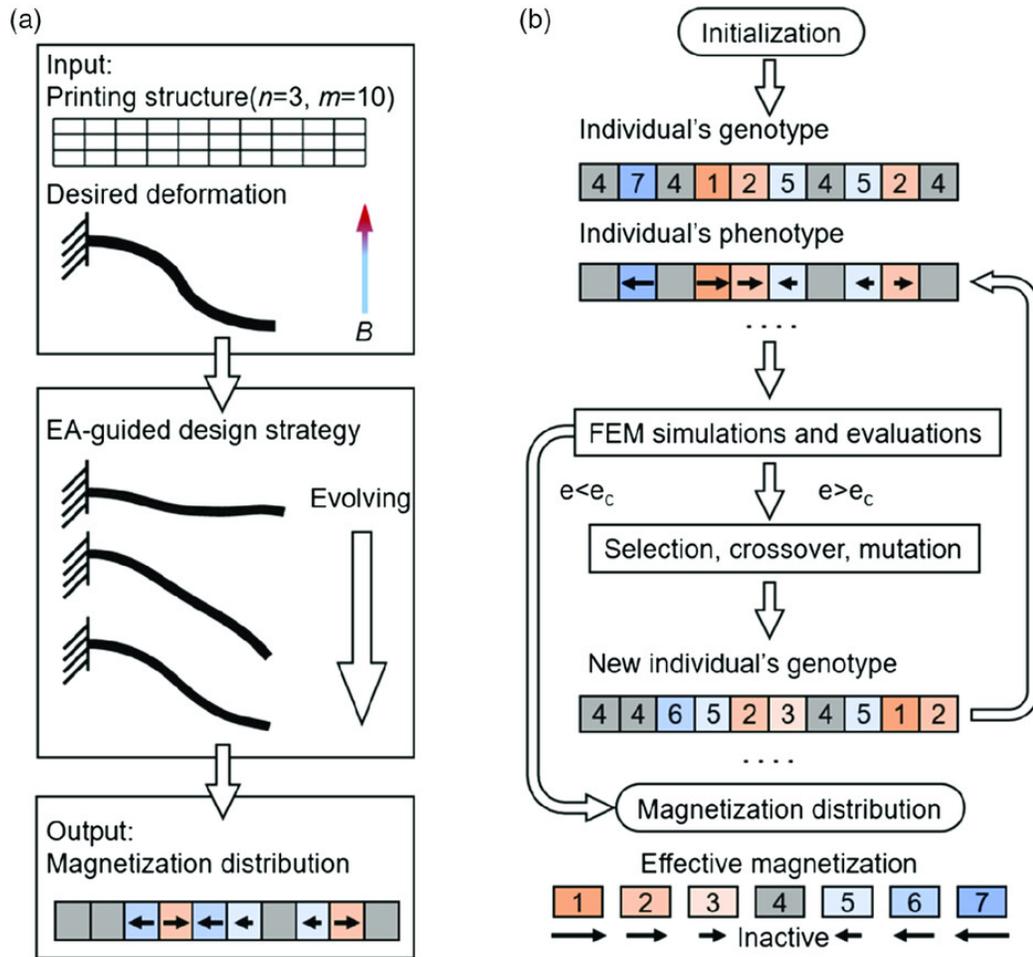


Figure 2.11: Evolutionary algorithm applied to voxel-encoding printing. a) The algorithm gets the desired deformation as input, and EA finds the required distribution to achieve the targeted deformation. b) Flowchart of the implemented EA. Reproduced from [6] under the Creative Commons Attribution License (CC BY 4.0).

Chapter 3

Methodology

A small-scale soft robot originally presented by Diller et al. [13], later adapted by Hu et al. [44], and reproduced by Davy et al. [26] is chosen as the foundation of this research. The proposed robot is capable of having multimodal locomotion, which consists of swimming inside or on the surface of a liquid, crawling, jumping, climbing, and walking. It is built from a mixture of silicon elastomer and hard magnetic particles that provide a flexible body that deforms under the influence of an external magnetic field. The shape and actuation method of the robot is similar to the one introduced by Zhang et al. [60], which focuses only on swimming movement. Although Zhang et al. have modeled the swimming speed of their swimmer robot, which can be used as part of the evaluation function for swimming movement, we have used the MPM-based simulation to optimize the robot with a focus on walking behavior, which paves the path for optimization of the soft-bodied robot by eliminating the need for hand-crafted mathematical models as the simulation captures the physical behavior needed for evaluation.

The reproduced robot by Davy et al. [26] consists of 9 sections aligned as an array,

as each section represents a discrete voxel. Each voxel can be magnetized differently, which means the direction and strength of the magnetization can be customized. These magnetization patterns determine how robots react to the external magnetic field, and by altering these patterns, we can actually design different robots with varied kinds of movements. In this study, the Covariance Matrix Adaptation Evolution Strategy (CMA-ES), a robust method for non-linear continuous optimization, is used as an optimization method to find the best magnetic configuration for each section of the robot to achieve a higher horizontal speed. The optimization process relies on an MPM-based Simulation that models magnetic soft robots to evaluate the performance of each individual in a rotating magnetic field in terms of the distance they travel in a specific amount of time.

Additionally, a genetic algorithm is also implemented for comparison. However, the GA fails to achieve optimization within the same number of generations and population size, which highlights the superiority of the CMA-ES to GA in this context.

The codebase developed for the optimization framework, including the GA and CMA-ES implementation and adapted MPM simulation environment, is available at <https://github.com/HeartLab-McMaster/MPM-Optimization>.

3.1 Algorithm

To optimize the magnetization pattern of the robot, two evolutionary algorithms, the Genetic Algorithm (GA) and the Covariance Matrix Adaptation Evolution Strategy (CMA-ES), are implemented and compared. The GA is selected due to its popularity in optimizing soft magnetic robots, and CMA-ES is chosen for its reputation

in continuous optimization. Both algorithms, GA and CMA-ES, use the same representation (3.2) and evaluation function (3.3), with identical population size and generation limit to ensure a fair comparison.

In this study, the external magnetic field applied to individuals during the evaluation phase was kept constant to enable comparison of different individuals in the identical condition. The individuals differ in the magnetic moments (both in direction and magnitude) of their segments. The algorithm seeks to find the optimal magnetic configuration for the robot by altering these magnetic moments and evaluating the resulting configuration.

3.1.1 Genetic Algorithm

The Genetic Algorithm (GA) has been discussed in detail in 2.4.1. As representation and evaluation functions are shared between GA and CMA-ES, they are discussed separately in Section 3.2 and Section 3.3, respectively. Here, we focus on the aspects that differentiate GA from CMA-ES, including crossover, mutation, and selection.

The algorithm starts with 12 randomly generated individuals. Each individual represents a unique magnetic configuration for the nine-segmented robot described earlier. Each individual is separately evaluated by constructing a robot based on its parameters, sending it to the simulation while exposing it to a fixed rotating magnetic field for a set period of time. As each robot has a unique magnetic configuration, they react differently to the magnetic actuation, resulting in varying traveled distances. This traveled distance is used as a fitness value to compare individuals (different magnetic configurations) with each other. Based on these assigned fitness values, 12 individuals are selected using tournament selection, a method that will be discussed in

detail later. In this selection process, some individuals may be selected multiple times, and some of them may not be selected at all. Then, some of these selected individuals undergo crossover, where two individuals blend their parameters, and mutation, where random small changes are introduced. Both operations ensure that the individuals' values remain within valid bounds. Only the individuals that are altered through crossover or mutation are reevaluated by running new simulations, and the resulting population replaces the previous one. This cycle repeats for 100 generations, as the algorithm proceeds without relying on convergence-based termination.

Selection: Tournament Selection

In this algorithm, tournament selection is chosen as the selection mechanism. In this method for selecting λ individuals from a population with the size of μ , a subgroup of k individuals is selected randomly, and the best individual in this subgroup is selected for reproduction. This allows the selection pressure to be adjustable and is a suitable approach when a subjective selection is needed by comparing the individuals to each other, which makes it an appropriate approach when EAs are applied in design and art. In this study, we use a tournament size of $k = 3$. The value of k determines the selection pressure. Increasing k increases the chance of selecting individuals with higher fitness, as they are more likely to be in the selected subset. This decreases the selection of individuals with lower fitness.

Crossover: Blend Crossover

In this study, we employ Blend Crossover ($BLX-\alpha$) as a recombination operator [61]. Unlike traditional crossover methods that generate offspring strictly inside the range

of parents' values, $BLX - \alpha$ allows offspring to take values both inside and outside that range. For each element of the solution representation (gene), if the parents have values x_i and y_i , the distance is defined as $d_i = |y_i - x_i|$. The offspring gene is then sampled uniformly from the range $[\min(x_i, y_i) - \alpha \cdot d_i, \max(x_i, y_i) + \alpha \cdot d_i]$. Here, the parameter α defines the exploration intensity. This method allows the offspring to allocate outside of the parents' range proportional to the distance between them and enables a diverse search of the solution space while providing refinement of current solutions. For our study, we set α to 0.5, which is the recommended value by the previous studies. Additionally, to keep the modified element within the defined range, the values exceeding the upper limit are set to the upper bound, and those falling below the lower limit are set to the lower bound.

Mutation: Gaussian Mutation

In the GA algorithm, we use a fixed-step mutation. Each gene in an individual has a 20% chance of being mutated by the step size of 0.1. This means that the mutation can increase or decrease the element values by 0.1. To keep the mutated element within the defined range, the same approach that is used in the crossover is applied.

3.1.2 CMA-ES

In this study, we use the CMA Python library to implement the CMA-ES optimization algorithm. The overall optimization process is shown in Algorithm 1. The algorithm is initialized with step size $\sigma = 1$, and all elements of the initial solutions are set to zero. This allows the algorithm to start the search from a neutral place and gradually shift toward more promising places by self-adaptation. Around this initial individual,

12 individuals are sampled from a multivariate normal distribution. Each individual is evaluated by constructing a robot based on its parameters and simulating its movement under a rotating magnetic field for a set time. Similar to the GA approach, the traveled distance is measured and used as the fitness value. After evaluation of all individuals, the fittest one is used to update the mean (center of sampling distribution), and the covariance matrix and step size are adjusted accordingly. This cycle also continues for 100 generations and does not rely on the convergence of fitness.

3.2 Representation

For determining the magnetization of each section, each individual is represented as an array of 9 pairs of y and z , which represent the vector of the magnetization. To avoid the algorithm exponentially growing, the y and z are constrained within a set boundary.

3.3 Evaluation Function

The evaluation function of our algorithm uses the MPM-based simulation as a critical tool to evaluate each robot's performance under the influence of a rotating magnetic field. The decoded individuals are sent to the simulation as robots with specific configurations. The simulation runs for a set period of time while the robot is exposed to a rotating magnetic field. The average position of the particles of the central section is considered the position of the robot, highlighted in red in Fig. 3.1.

This approach is necessary as the robot has the potential to bend during its movement. As shown in Fig. 3.1, considering the position of the first or last section

as the position of the robot can lead to inaccurate results. For instance, one robot may end the simulation in a straight shape (Fig. 3.1.a), while another finishes in a bent shape (Fig. 3.1.b). In such cases, even if their central sections are aligned at the same position, using the first section (blue) as the position of the robot would make it appear as though the straight robot traveled a shorter distance. Using the last segment (yellow) would suggest that the bent robot traveled a shorter distance. In both cases, the robots' final positions are the same, and only the final shapes of the robots are different. Therefore, using the central segment (red) ensures the correct displacement measurement that is independent of the robot's shape.

$$\text{Fitness} = y_{\text{final}} - y_{\text{initial}} \quad (3.3.1)$$

The evaluation function calculates the distance that each robot travels along the y -axis (forward) in a specific period of time and uses it as the fitness value that we are trying to increase.

3.3.1 Penalty Conditions

Some penalty conditions are applied during simulation to reduce computational cost while encouraging effective movements. If these conditions are violated, the simulation is terminated, and a penalty value, which is negative, is applied based on the severity of the violation. The fitness value in terms of violation is defined as below:

$$\text{Fitness} = \frac{\text{Penalty Value}}{\text{Time to Reach Threshold}} \quad (3.3.2)$$

This ensures that individuals who violate the constraints sooner receive worse fitness values than those which violate them more gradually. This enables the algorithm to rank the individuals even though they are not acceptable and provides a path for finding acceptable individuals gradually and optimizing their performance afterward. The penalty applies under the following conditions:

It is worth mentioning that the penalty value is not simply subtracted from the fitness value calculated using Eq. (3.3.1); instead, a new form of fitness calculation, given by Eq. (3.3.2), is employed. In fact, the Eq. (3.3.2) is used to calculate the fitness value when the simulation is terminated early due to the violation of penalty conditions, rather than to merely adjust the fitness with a penalty.

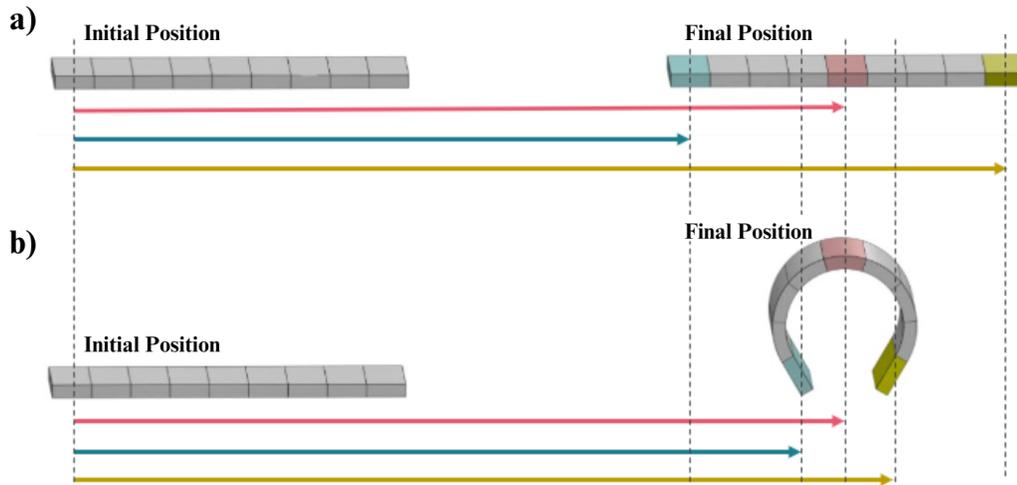


Figure 3.1: This figure illustrates the position measurement for robots with different final shapes. a) A robot ending the simulation with a straight shape. b) A robot ending the simulation with a bent shape. The Blue, yellow, and red arrows represent displacement measurements taken from the corresponding colored segments.

Backward Movement Threshold: If a robot moves backward beyond a defined threshold, the simulation ends, and the fitness value is calculated based on the provided equation. This prevents the waste of computation for individuals with poor forward movement while enabling the evolution of better individuals based on these poor ones that are finally capable of moving forward quickly.

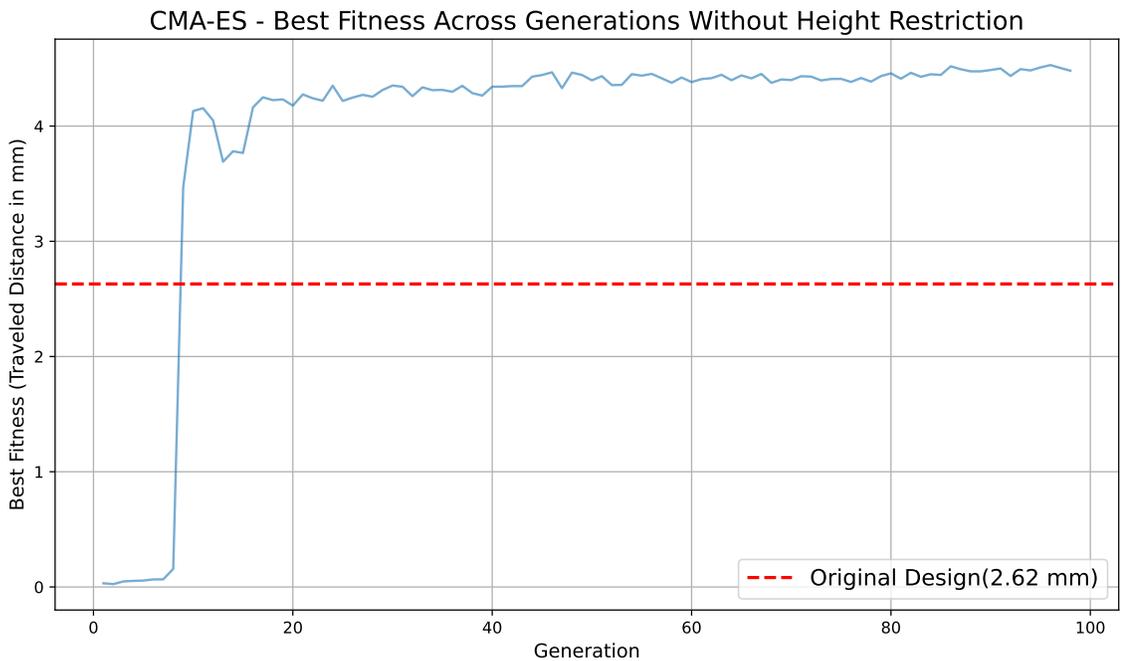


Figure 3.2: The best fitness values over 100 generations without height restriction.

Height Limitation Threshold: A height limitation is introduced to guide the algorithm through acceptable behavior. Without this constraint, the algorithm evolves robots that move by flipping instead of crawling. They rotate around the front part of the robot as the pivot point, which causes a large initial displacement, but this movement does not continue, and can be done for only one rotation. This particular behavior is shown in Fig. 3.3, and the performance of CMA-ES over generations

without height restriction is illustrated in Fig. 3.2. The best fitness achieved without height restriction increases rapidly, corresponding to the evolution of flipping behavior, and even outperforms the original design within the set simulation time. However, the flipping behavior is not a continuous locomotion. When the simulation time is extended, it is evident that the original design’s traveled distance increases, while the robot optimized by the algorithm is stuck after one flip and does not advance further. The algorithm considers the flip as a successful large displacement. Even with an extended simulation time, the algorithm fails to recognize the lack of continuous movement. By limiting the height that the central segment is allowed to exceed, this kind of movement receives a penalty and gets eliminated from later generations. This limitation can also be set to a specific height to ensure the resulting robot is capable of navigating through a specific narrow path.

3.4 Simulation Environment

While finite element method (FEM) modeling is widely used for optimization in previous studies, this study utilizes an MPM-based simulation as part of the evaluation function in this study [26]. FEM is a highly accurate modeling method that can be used for simulating the deformation of magnetic soft robots in the presence of a magnetic field and is capable of handling complex geometry. To simplify the FEM, the magnetic elements can be considered discrete, but under this assumption, the FEM loses its accuracy in large deformations. Zhao et al. addressed this issue by introducing a stressing factor. However, the non-linear FEM is computationally expensive because it relies on a numerical iterative minimization process.

Algorithm 1 CMA-ES Optimization for Magnetic Soft Robot Design

```
1: Initialize CMA-ES with:
2:   Population size
3:   Step size ( $\sigma$ )
4:   Bounds for the search space
5:   Maximum number of generations ( $gen_{max}$ )
6: for each generation do
7:   for each individual in the population do
8:      $fitness \leftarrow$  run_simulation (individual)
9:     Assign  $fitness$  to the individual
10:  end for
11:  Update CMA-ES model with:
12:    Fitness values
13:    Candidate solutions
14:  if termination criteria are met then
15:    Exit the loop
16:  end if
17: end for
18: return the best solution

19: Subroutine: run_simulation(individual)
20: Initialize the robot using the individual's parameters
21: Set simulation parameters (e.g., grid size, time step, duration, penalty value)
22: while simulation time  $\leq$  duration do
23:   Perform one MPM simulation step
24:   Update the magnetic field dynamically based on time and frequency
25:   if robot exits the grid or violates constraints then
26:      $fitness \leftarrow$  penalty/time
27:     Exit the simulation loop
28:   end if
29:   if intermediate checks (e.g., height threshold) fail then
30:      $fitness \leftarrow$  penalty/time
31:     Exit the simulation loop
32:   end if
33: end while
34: if simulation completes successfully without penalties then
35:    $fitness \leftarrow$  distance between the robot's initial and final positions
36: end if
37: return  $fitness$ 
```

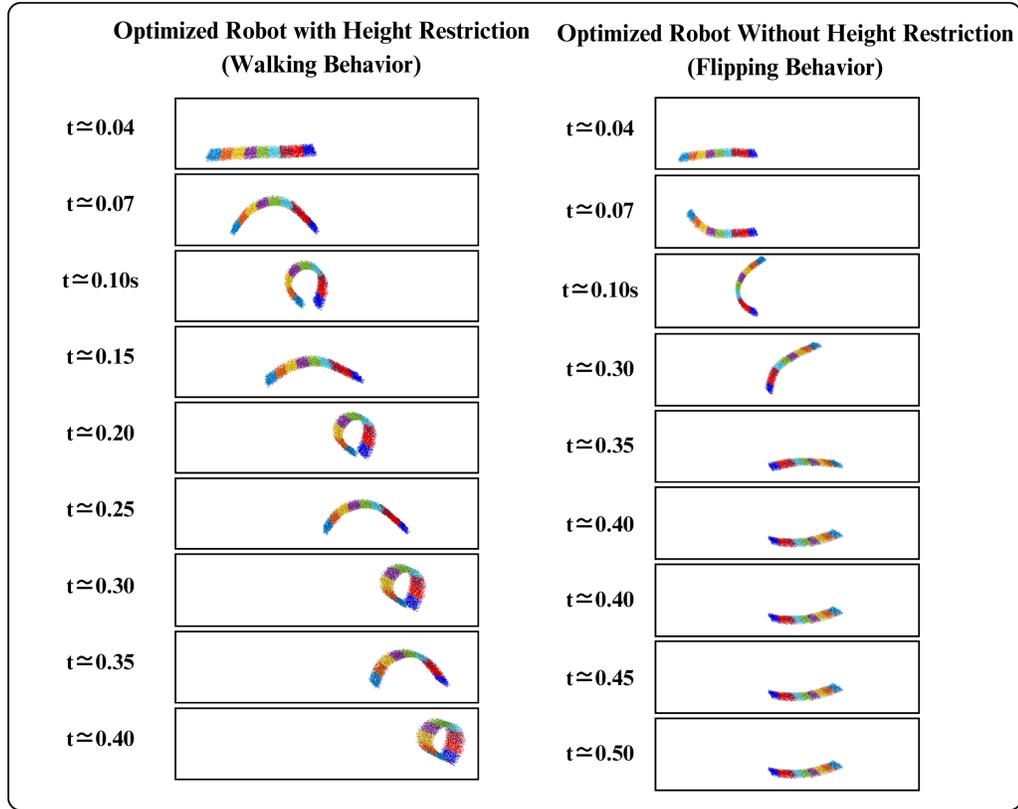


Figure 3.3: This figure illustrates the behavior of the robot optimized with and without height restriction over time. a) The robot optimized with a height restriction shows a stable and continuous walking behavior by bending and straightening in a cycle, while the robot optimized without a height restriction b) demonstrates a flipping behavior that is not continuous, and the robot gets stuck and is unable to progress further.

On the other hand, the Material Point Method (MPM) offers advantages in handling large deformations and self-collisions, which makes it a suitable tool for modeling magnetic soft robots. Additionally, MPM is well-suited for parallel computing, which makes it a faster alternative to FEM.

MPM consists of four cyclical steps. In the first step, the internal forces and magnetic stresses are calculated in the particle domain. Then, the properties of the particles are transferred to the grid domain to apply further calculations. In the next

step, the external field, such as the one applied due to the external magnetic field, is calculated in the grid domain. Finally, the calculated values are transferred back to the particle domain. Please refer to [26] for more detailed information.

A group of parameters needs to be set in this simulation environment to achieve the desired modeling of the robot. The particle density is set to 1×10^{13} particles/m³ to have a suitable resolution for the robots. The simulation grid size is set to 25 mm with a spatial resolution equal to 0.1 mm to have an accurate calculation of the deformation and movement.

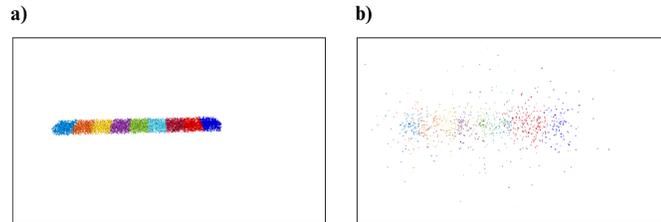


Figure 3.4: Comparison of simulation stability. a) Stable simulation with sufficient particle density and a small time step b) Unstable simulation resulting from reduced particle density and increased time step causes particles to disperse and loss of structure

Initially, we tried to decrease the computational cost of the simulation by decreasing the particle density and increasing the simulation step size, with the assumption that high precision is not needed in the evolving phase, and the final optimized robot can be validated by a high-precision setting at the end. However, this approach was unsuccessful, as reducing the particle density and/or increasing the simulation caused the simulation to become unstable and fail to calculate the particles' position properly, leading to particles dispersing across the simulation domain and the robot losing its shape, as demonstrated in Fig. 3.4.

Each individual is placed in the simulation environment while it is exposed to a

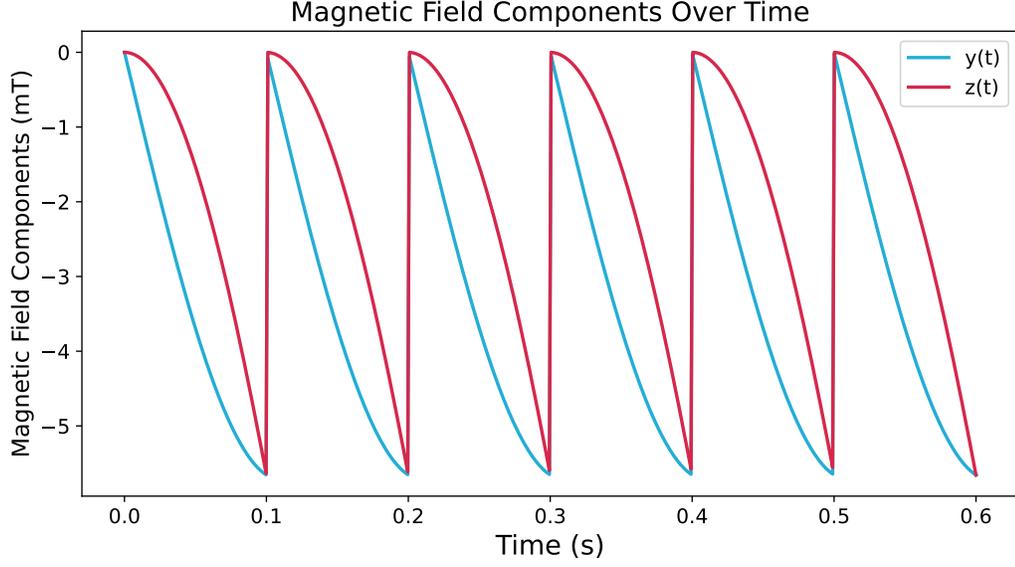


Figure 3.5: Plot of the B_y and B_z components of the magnetic field over six cycles.

rotating magnetic field for a set period of time. The rotation of the magnetic field in the yz -plan causes the walking-like movement of the robot. This rotating magnetic field is defined as follows:

$$\mathbf{B}(t) = \begin{bmatrix} 0 \\ M \cos \theta \\ M \sin \theta \end{bmatrix} \quad (3.4.1)$$

where the magnitude of the field is given by:

$$M = M_{\max} \left(t \bmod \frac{1}{f} \right) \cdot f \quad (3.4.2)$$

The rotation angle is given as:

$$\theta = \pi + \left(t \bmod \frac{1}{f} \right) \cdot f \cdot \frac{\pi}{4} \quad (3.4.3)$$

Here, M_{max} represents the maximum magnitude of the magnetic field, which is equal to 8 mT, and f is the frequency that determines the rotation speed of the magnetic field, and it is set to 10 Hz. This rotating magnetic field that causes the walking movement of the robot is shown in Fig. 3.5.

Chapter 4

Results

The CMA-ES algorithm was executed for five runs using unique random seeds with a reset strategy, which is explained later in this section, and the result was used for further analysis. Due to the computationally heavy simulation environment that has been used, it was not feasible to run more experiments. The best fitness value achieved among these five runs is 3.828 mm, which is the longest distance traveled by the robot in the given time. To make a comparison, the original robot design was simulated under the same conditions, which traveled the distance of 2.630 mm over the same duration. This represents a 45.5% improvement in the horizontal speed of the robot.

Notably, the algorithm begins its search process with randomized initial configurations and is not provided with information on the original design. Despite this, the algorithm not only produces the desired walking behavior but also exceeds the original's performance in terms of speed. This was achieved only by defining a proper fitness function and constraints.

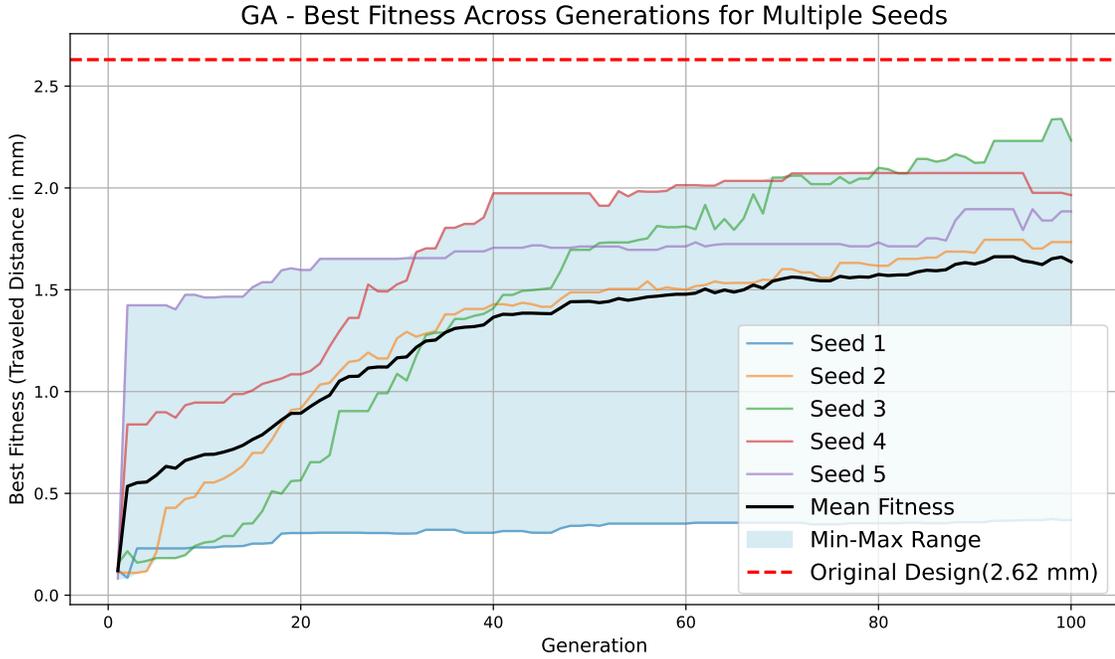


Figure 4.1: The best fitness values over 100 generations for five independent GA runs.

4.0.1 Performance of GA vs. CMA-ES

To analyze the optimization process further, the genetic algorithm was also tested with the same population size and number of generations as CMA-ES. The results indicate that GA is capable of improving fitness over generations. However, it converges before outperforming the original design.

Fig. 4.1 illustrates the best fitness values over 100 generations for five independent runs of GA, with the mean fitness across all runs shown by the black line in the plot. The original human-designed traveled distance, measured over the same simulation period as the evolved individuals, is indicated by a dashed line for reference. The optimization process aims to approach and ideally surpass this line through generations. Based on this figure, all the individual runs start with a low fitness value close

to zero. Seed 1 performs the worst among all other seeds and manages to reach only 0.37 mm. On the other hand, although seed 3 reaches the lowest value in comparison to seeds 2, 4, and 5, its growth rate is considerably higher than the others, as it achieves a maximum of 2.33 mm, which is the closest value to the original design. Furthermore, seed 5 shows the best performance during the first 30 generations, but is overtaken by seed 4 in the following generations, and they both finally converge to approximately a similar point. Lastly, the mean fitness after 100 generations reaches a value of about 1.7 mm, which is mostly due to the disappointing trend of seed 1.

Subsequently, Fig. 4.2 illustrates the best fitness values over 100 generations through five independent runs of CMA-ES, where it reaches an extraordinary achievement by exceeding the original design in 4 out of 5 seeds. Similar to the previous plot, seed 1, by a gradual and slight growth, reaches 0.4 mm, which is the worst result among all the seeds. In contrast, it is evident that using CMA-ES results in a considerable increase in the fitness value beyond 2.62 mm, by approximately the same pattern for the seeds 2, 4, and 5. While seed 4, after 100 generations, reaches to the highest value of about 3.7 mm, it is worth noting that at a specific point between the generations 70 and 80, seed 5 shows a sudden jump, and hits a fitness value of 4.05 mm, which is the highest point among all seeds through all generations. Then, it declines slightly to the second rank trend again. Additionally, the mean fitness reaches about 2.7 mm in the final generation, which is lower than all seeds except for seed 1. Nevertheless, it still remains above the original design value.

By overall observation of the improvement in fitness values over 100 generations in both GA and CMA-ES, it can be concluded that while CMA-ES achieves a fitness value that outperforms the original human-designed robot by 45.5% in 5 runs,

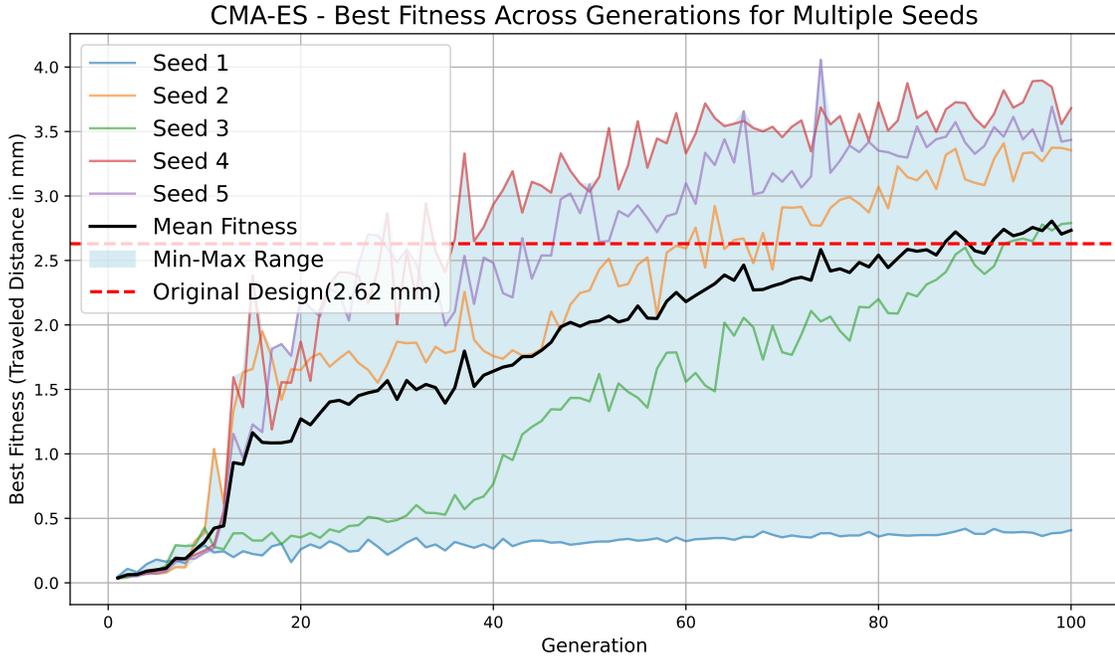


Figure 4.2: The best fitness values over 100 generations for five independent CMA-ES runs.

the highest fitness value that GA managed to reach in the same number of runs is 2.339 mm, which is 11% less than the original design’s performance.

To further illustrate the difference in convergence speeds and overall performance between these two algorithms, we plotted the mean fitness across all CMA-ES and GA runs in Fig. 4.3. It is evident that the convergence of CMA-ES is significantly better; however, in early generations, the GA temporarily outperforms CMA-ES. This indicates that GA cannot compete with CMA-ES over time in solving this problem.

To prevent unsuccessful runs in CMA-ES and improve convergence reliability, a restart strategy was implemented for this algorithm. In each run, if the fitness value does not reach 0.5 mm by generation 20, the algorithm completely restarts. This approach limits the waste of resources on the algorithms that are stuck in local

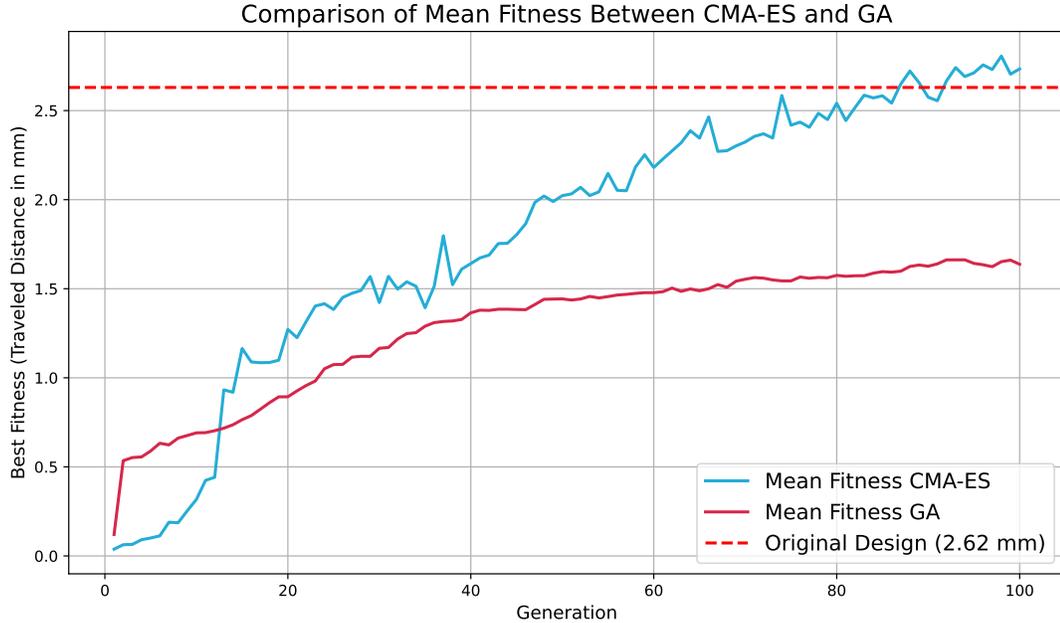


Figure 4.3: Mean of the best fitness achieved over 100 generations for five independent runs of GA and CMA-ES.

optima. To achieve a reliable, optimized robot, the algorithm was executed by this strategy for five independent seeds, and the results from these runs are plotted in Fig. 4.4. According to the result, all seeds are successful in overtaking the original design, and they can achieve the fitness value of more or less 3.5 mm. All seeds at the beginning show a sharp rise in their trend of increasing the fitness value, and then change into a gradual growth. Interestingly, the best fitness value of the best seed in both forms of CMA-ES, with and without the restart strategy, stands around 3.7 mm.

4.0.2 Best-Performing Individuals and Their Behaviors

The best individual of the generation from the two seeds that reached the highest fitness over generations, seed 2 and seed 4, of the algorithm using the restart strategy

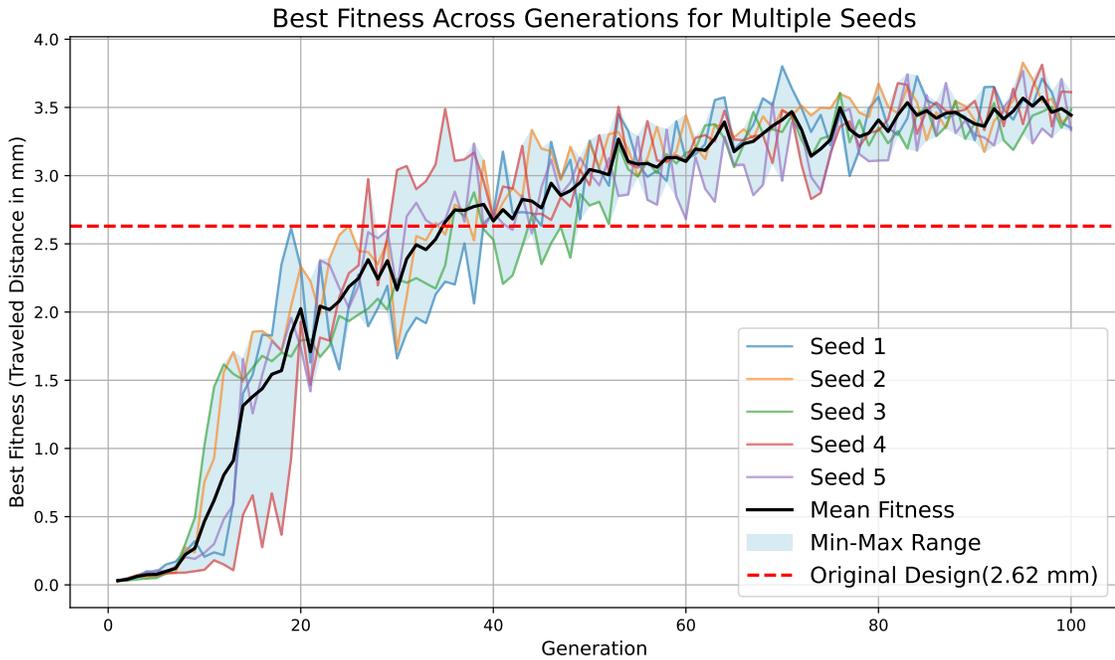


Figure 4.4: The best fitness values over 100 generations for five independent CMA-ES runs with the restart strategy.

were chosen, and their behaviors were then observed in the simulation environment for twice the original simulation time. Both individuals exhibit the same movement pattern as the original design, cyclical bending and straightening movements, which cause a forward motion. The magnetic configuration of these two individuals differs significantly from the original design and from each other. This difference shows the algorithm’s ability to discover innovative solutions by being free from the human basis toward traditional designs.

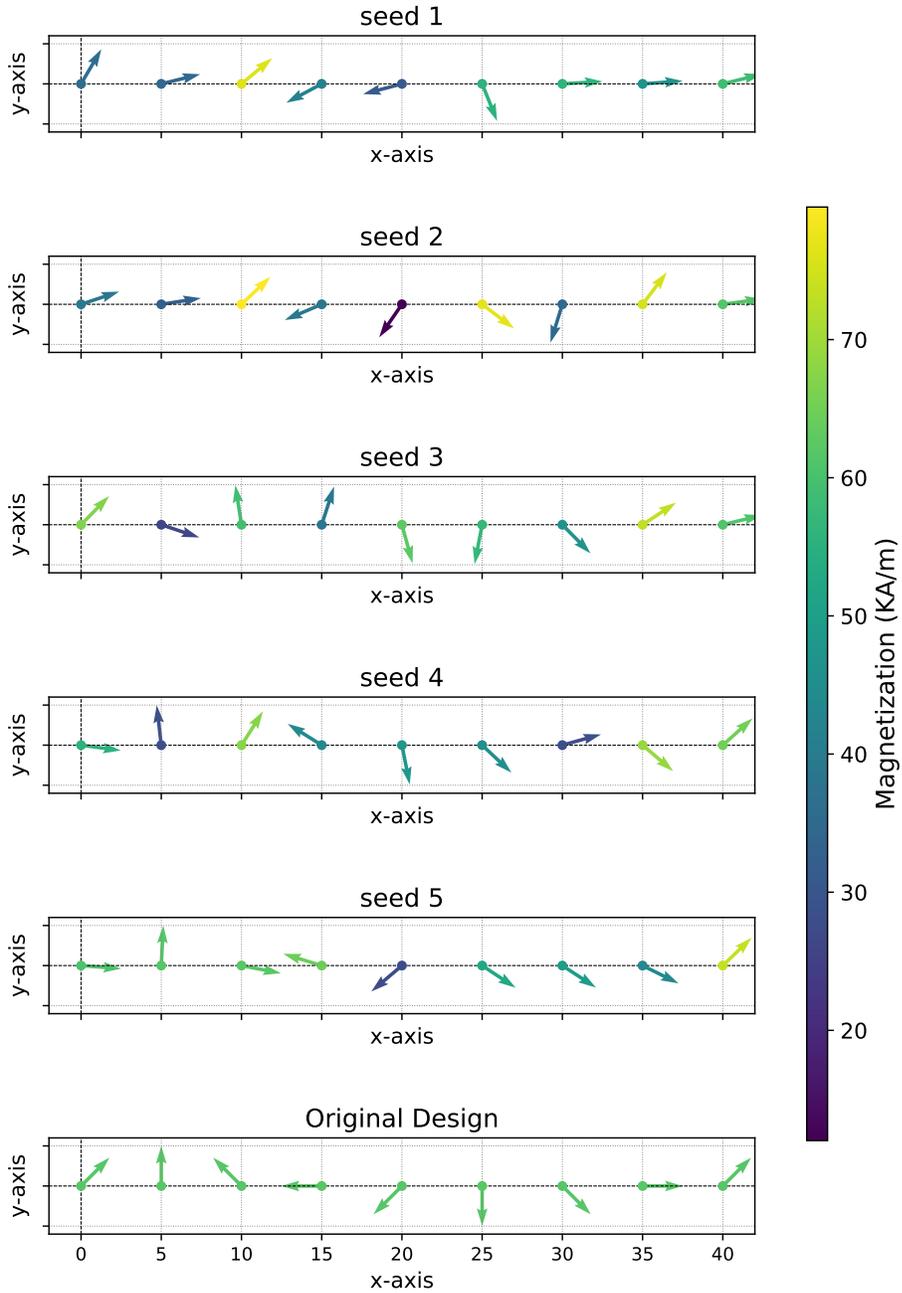


Figure 4.5: This figure illustrates the magnetic configurations of the best individuals from the best generations of each seed. The original robot’s magnetic configuration is also included for reference. The direction of magnetization of each segment is represented by an arrow, while the color of the arrows indicates its magnitude.

Chapter 5

Conclusion

This thesis presented an optimization framework for untethered small-scale magnetic soft robots using two variations of evolutionary algorithms: Genetic Algorithm (GA) and Covariance Matrix Adaptation Evolution Strategy (CMA-ES). By combining these optimization techniques with a Material Point Method (MPM) simulation environment, we successfully designed and optimized the magnetic profile of a soft magnetic robot to achieve high-speed horizontal locomotion. This study highlights the effectiveness of the CMA-ES over the evolutionary algorithm that is commonly used in this field, the genetic algorithm. This study offers a more flexible and efficient approach for designing magnetic soft robots using the desired movement.

Through a comparative study between the GA and CMA-ES, we revealed the advantage of CMA-ES over GA in designing/optimizing the magnetic profile of untethered magnetic soft robots.

Both GA and CMA-ES showed improvement in fitness over generations; however, their convergence rates and final optimization outcomes after the same number of generations varied significantly. Although GA demonstrates an increase in fitness

function, its convergence rate was extremely slow compared to CMA-ES with the same population size. Having a computationally expensive simulation environment as a part of the evaluation function prevents us from dedicating more generations to GA. While CMA-ES reached a fitness value that outperforms the original human-designed configuration by 45.5%, GA only achieved 89% of the original design’s performance, which is relatively low. This highlights the efficiency of CMA-ES in finding high-quality solutions in fewer iterations by having a small population. This efficiency is likely due to the ability of CMA-ES to adapt to the search distribution by using a covariance matrix, while GA relies on fixed mutation and crossover strategies.

Another major contribution of this study is the use of Material Point Method (MPM) based simulation instead of the traditional Finite Element Method (FEM) based approach. While FEM is widely used for simulating and modeling the magnetic soft robot for optimization using evolutionary algorithms, MPM offers better handling of large deformation and self-collisions and computational efficiency for highly flexible structures. The integration of MPM with CMA-ES allowed for more realistic motion modeling of the robots, which makes this study an important step towards more sophisticated designing and optimization of magnetic soft robots using AI.

Nevertheless, this work has several limitations. First, the computational cost of MPM simulations remains high, which limits the number of generations and population sizes that can be practically used. This issue could be addressed by choosing a more efficient simulation environment. The current simulation used in this study is implemented in Python, which, as an interpreted language, is not as fast as compiled languages. Reimplementing the critical parts of the simulation or the entire simulation in a compiled programming language such as C++ could significantly increase

the simulation speed, allowing for larger population sizes and a greater number of generations, which provides a more thorough exploration of the solution space and optimization of robots with a higher number of segments.

Secondly, the current algorithm optimizes both the magnitude and orientation of the magnetic moment for segments. However, manufacturing a robot with segments having different magnetization magnitudes is a challenging task. To facilitate the practical manufacturing of the optimized robot, the algorithm could be modified to optimize only the magnetization direction and represent individuals by a list of angles instead of using the vector components (x and y) while keeping the magnetization magnitude uniform across all segments. Additionally, in practice, automated magnetic material printers tend to have limitations on the range of achievable magnetization magnitudes and offer discrete magnitude levels rather than a continuous range. To support these printers, future versions of the optimization framework should explore selecting magnitudes from a discrete set of available values while still optimizing the directions over a continuous search space. This adjustment would increase the manufacturability of the optimized design using the current fabrication technologies.

The promising results of this research open several directions for future work. Although the integration of the MPM simulation environment and CMA-ES leads to significant improvement in the speed of the magnetic soft robot studied, a more computationally efficient environment can further enhance the adaptability of this framework to more complex soft robot structures with a higher number of segments.

Additionally, this approach can be applied for the optimization of other locomotion types by having varied fitness functions and applying the algorithm on varied kinds of magnetic soft robots. Furthermore, experimental validation of the robot is a crucial

step that evaluates the optimized robot in real-world conditions by manufacturing and testing physical prototypes.

Moreover, this algorithm can be extended to a multi-objective optimization problem, such as balancing the robot locomotion performance with design simplicity, especially when considering manufacturing limitations.

In conclusion, this study is an important step toward the automated design and optimization of magnetic soft robots. By demonstrating the advantages of CMA-ES over the traditional evolutionary method commonly used in this field, the genetic algorithm, and the advantages of using MPM simulation in the evaluation phase of EA, this study contributes to advancing the field of AI-driven robotic optimization. With the advancement of simulation technology, computational techniques, and tools, the future of designing magnetic soft robots can move toward full automation, enabling more innovative and efficient designs.

Bibliography

- [1] Yoonho Kim and Xuanhe Zhao. Magnetic soft materials and robots. *Chemical Reviews*, 122:5317–5364, 3 2022.
- [2] Alistair Bacchetti, Peter Lloyd, Silvia Taccola, Evan Fakhoury, Sandy Cochran, Russell A Harris, Pietro Valdastri, James H Chandler, Angela Faragasso, and Massimo Mastrangeli. Optimization and fabrication of programmable domains for soft magnetic robots: A review. 2022.
- [3] Chunping Ma, Shuai Wu, Qiji Ze, Xiao Kuang, Rundong Zhang, H. Jerry Qi, and Ruike Zhao. Magnetic multimaterial printing for multimodal shape transformation with tunable properties and shiftable mechanical behaviors. *ACS applied materials interfaces*, 13:12639–12648, 3 2021.
- [4] A. Eiben and Jim Smith. *Introduction To Evolutionary Computing*, volume 45. 01 2003.
- [5] Liu Wang, Dongchang Zheng, Pablo Harker, Aman B. Patel, Chuan Fei Guo, and Xuanhe Zhao. Evolutionary design of magnetic soft continuum robots. *Proceedings of the National Academy of Sciences of the United States of America*, 118:e2021922118, 5 2021.

- [6] Shuai Wu, Craig M. Hamel, Qiji Ze, Fengyuan Yang, H. Jerry Qi, and Ruike Zhao. Evolutionary algorithm-guided voxel-encoding printing of functional hard-magnetic soft active materials. *Advanced Intelligent Systems*, 2:2000060, 8 2020.
- [7] S. Zhao, D. Sun, J. Zhang, H. Lu, Y. Wang, R. Xiong, and K. T.V. Grattan. Actuation and biomedical development of micro-/nanorobots – a review. *Materials Today Nano*, 18:100223, 6 2022.
- [8] Jiachen Zhang. Small-scale robots with programmable magnetization profiles. *Untethered Small-Scale Robots for Biomedical Applications*, pages 119–139, 1 2023.
- [9] Igor Paprotny and Sarah Bergbreiter. Small-scale robotics : An introduction. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 8336 LNAI:1–15, 2014.
- [10] Hakan Ceylan, Joshua Giltinan, Kristen Kozielski, and Metin Sitti. Mobile microrobots for bioengineering applications. *Lab on a Chip*, 17:1705–1724, 5 2017.
- [11] Shaojun Jiang, Bo Li, Jun Zhao, Dong Wu, Yiyuan Zhang, Zhipeng Zhao, Yiyuan Zhang, Hao Yu, Kexiang Shao, Cong Zhang, Rui Li, Chao Chen, Zuojun Shen, Jie Hu, Bin Dong, Ling Zhu, Jiawen Li, Liqiu Wang, Jiaru Chu, and Yanlei Hu. Magnetic janus origami robot for cross-scale droplet omni-manipulation. *Nature Communications*, 14, 2023.
- [12] Su Eun Chung, Xiaoguang Dong, and Metin Sitti. Three-dimensional heterogeneous assembly of coded microgels using an untethered mobile microgripper. *Lab on a Chip*, 15:1667–1676, 3 2015.

- [13] Eric Diller and Metin Sitti. Three-dimensional programmable assembly by untethered magnetic robotic micro-grippers. *Advanced Functional Materials*, 24:4397–4404, 7 2014.
- [14] Masaya Hagiwara, Tomohiro Kawahara, Toru Iijima, and Fumihito Arai. High-speed magnetic microrobot actuation in a microfluidic chip by a fine v-groove surface. *IEEE Transactions on Robotics*, 29:363–372, 2013.
- [15] Ziheng Chen, Yibin Wang, Hui Chen, Junhui Law, Huayan Pu, Shaorong Xie, Feng Duan, Yu Sun, Na Liu, and Jiangfan Yu. A magnetic multi-layer soft robot for on-demand targeted adhesion. *Nature Communications 2024 15:1*, 15:1–13, 1 2024.
- [16] Ren Hao Soon, Zhen Yin, Metin Alp Dogan, Nihal Olcay Dogan, Mehmet Efe Tiryaki, Alp Can Karacakol, Asli Aydin, Pouria Esmaeili-Dokht, and Metin Sitti. Pangolin-inspired untethered magnetic robot for on-demand biomedical heating applications. *Nature Communications*, 14, 12 2023.
- [17] Sehyuk Yim and Metin Sitti. Shape-programmable soft capsule robots for semi-implantable drug delivery. *IEEE Transactions on Robotics*, 28:1198–1202, 2012.
- [18] Onaizah Onaizah and Eric Diller. Tetherless mobile micro-surgical scissors using magnetic actuation. *Proceedings - IEEE International Conference on Robotics and Automation*, 2019-May:894–899, 5 2019.
- [19] Islam S.M. Khalil, Dalia Mahdy, Ahmed El Sharkawy, Ramez R. Moustafa, Ahmet Fatih Tabak, Mohamed E. Mitwally, Sarah Hesham, Nabila Hamdi, Anke Klingner, Abdelrahman Mohamed, and Metin Sitti. Mechanical rubbing of blood

- clots using helical robots under ultrasound guidance. *IEEE Robotics and Automation Letters*, 3:1112–1119, 4 2018.
- [20] Arijit Ghosh, Chang Kyu Yoon, Federico Ongaro, Stefano Scheggi, Florin M. Selaru, Sarthak Misra, and David H. Gracias. Stimuli-responsive soft untethered grippers for drug delivery and robotic surgery. *Frontiers in Mechanical Engineering*, 3:267124, 7 2017.
- [21] Zhiguang Wu, Xiankun Lin, Xian Zou, Jianmin Sun, and Qiang He. Biodegradable protein-based rockets for drug transportation and light-triggered release. *ACS Applied Materials and Interfaces*, 7:250–255, 1 2015.
- [22] Mariana Medina-Sánchez, Lukas Schwarz, Anne K. Meyer, Franziska Hebenstreit, and Oliver G. Schmidt. Cellular cargo delivery: Toward assisted fertilization by sperm-carrying micromotors. *Nano Letters*, 16:555–561, 1 2016.
- [23] Daniela Rus and Michael T. Tolley. Design, fabrication and control of soft robots. *Nature 2015 521:7553*, 521:467–475, 5 2015.
- [24] Michele Di Lecce, Onaizah Onaizah, Peter Lloyd, James H. Chandler, and Pietro Valdastrì. Evolutionary inverse material identification: Bespoke characterization of soft materials using a metaheuristic algorithm. *Frontiers in Robotics and AI*, 8:790571, 1 2022.
- [25] N. Hansen and A. Ostermeier. Completely derandomized self-adaptation in evolution strategies. *Evolutionary computation*, 9:159–195, 2001.

- [26] Joshua Davy, Peter Lloyd, James H. Chandler, and Pietro Valdastri. A framework for simulation of magnetic soft robots using the material point method. *IEEE Robotics and Automation Letters*, 8, 2023.
- [27] Jiaqi Miao and Siqi Sun. Design, actuation, and functionalization of untethered soft magnetic robots with life-like motions: A review, 2023.
- [28] Yuanhao Chen, Jiajia Yang, Xuan Zhang, Yiyu Feng, Hao Zeng, Ling Wang, and Wei Feng. Light-driven bimorph soft actuators: design, fabrication, and properties. *Materials Horizons*, 8:728–757, 3 2021.
- [29] Junseong Ahn, Jimin Gu, Jungrak Choi, Chankyu Han, Yongrok Jeong, Jaeho Park, Seokjoo Cho, Yong Suk Oh, Jun Ho Jeong, Morteza Amjadi, and Inkyu Park. A review of recent advances in electrically driven polymer-based flexible actuators: Smart materials, structures, and their applications. *Advanced Materials Technologies*, 7:2200041, 11 2022.
- [30] Weijie Liu, Yongfa Cheng, Nishuang Liu, Yang Yue, Dandan Lei, Tuoyi Su, Meng Zhu, Zhi Zhang, Wei Zeng, Haizhong Guo, and Yihua Gao. Bionic mxene actuator with multiresponsive modes. *Chemical Engineering Journal*, 417:129288, 8 2021.
- [31] Qiang Zhao, John W.C. Dunlop, Xunlin Qiu, Feihe Huang, Zibin Zhang, Jan Heyda, Joachim Dzubiella, Markus Antonietti, and Jiayin Yuan. An instant multi-responsive porous polymer actuator driven by solvent molecule sorption. *Nature Communications* 2014 5:1, 5:1–8, 7 2014.
- [32] Mehdi Eshaghi, Mohsen Ghasemi, and Korosh Khorshidi. Design, manufacturing

- and applications of small-scale magnetic soft robots. *Extreme Mechanics Letters*, 44, 2021.
- [33] Hongman Wang, Zhisen Zhu, He Jin, Rui Wei, Lei Bi, and Wenling Zhang. Magnetic soft robots: Design, actuation, and function. *Journal of Alloys and Compounds*, 922:166219, 11 2022.
- [34] David Quashie, Prateek Benhal, Zhi Chen, Zihan Wang, Xueliang Mu, Xiaoxia Song, Teng Jiang, Yukun Zhong, U. Kei Cheang, and Jamel Ali. Magnetic bio-hybrid micro actuators. *Nanoscale*, 14:4364–4379, 3 2022.
- [35] Nafiseh Ebrahimi, Chenghao Bi, David J. Cappelleri, Gastone Ciuti, Andrew T. Conn, Damien Faivre, Neda Habibi, Alexander Hošovský, Veronica Iacovacci, Islam S.M. Khalil, Veronika Magdanz, Sarthak Misra, Chytra Pawashe, Rasoul Rashidifar, Paul Eduardo David Soto-Rodriguez, Zoltan Fekete, and Amir Jafari. Magnetic actuation methods in bio/soft robotics. *Advanced Functional Materials*, 31:2005137, 3 2021.
- [36] Bradley J. Nelson, Ioannis K. Kaliakatsos, and Jake J. Abbott. Microrobots for minimally invasive medicine. *Annual Review of Biomedical Engineering*, 12:55–85, 8 2010.
- [37] Xinjian Fan, Xiaoguang Dong, Alp C. Karacakol, Hui Xie, and Metin Sitti. Reconfigurable multifunctional ferrofluid droplet robots. *Proceedings of the National Academy of Sciences of the United States of America*, 117, 2020.

- [38] Vignesh Sahadevan, Bivas Panigrahi, and Chia-Yuan Chen. micromachines microfluidic applications of artificial cilia: Recent progress, demonstration, and future perspectives. 2022.
- [39] Jiandong Man, Junjie Zhang, Guangyuan Chen, Ning Xue, and Jiamin Chen. A tactile and airflow motion sensor based on flexible double-layer magnetic cilia. *Microsystems and Nanoengineering*, 9, 2023.
- [40] Wilfried Andrä and Hannes Nowak. *Magnetism in Medicine: A Handbook*. 2007.
- [41] Zhengxin Yang and Li Zhang. Magnetic actuation systems for miniature robots: A review. *Advanced Intelligent Systems*, 2:2000082, 9 2020.
- [42] Jake J. Abbott, Eric Diller, and Andrew J. Petruska. Magnetic methods in robotics. *Annual Review of Control, Robotics, and Autonomous Systems*, 3:57–90, 5 2020.
- [43] Tianqi Xu, Jiachen Zhang, Mohammad Salehizadeh, Onaizah Onaizah, and Eric Diller. Millimeter-scale flexible robots with programmable three-dimensional magnetization and motions. *Science Robotics*, 4, 4 2019.
- [44] Wenqi Hu, Guo Zhan Lum, Massimo Mastrangeli, and Metin Sitti. Small-scale soft-bodied robot with multimodal locomotion. *Nature*, 554, 2018.
- [45] Yoonho Kim, Hyunwoo Yuk, Ruike Zhao, Shawn A. Chester, and Xuanhe Zhao. Printing ferromagnetic domains for untethered fast-transforming soft materials. *Nature*, 558:274–279, 6 2018.
- [46] Guo Zhan Lum, Zhou Ye, Xiaoguang Dong, Hamid Marvi, Onder Erin, Wenqi Hu, and Metin Sitti. Shape-programmable magnetic soft matter. *Proceedings of*

- the National Academy of Sciences of the United States of America*, 113:E6007–E6015, 10 2016.
- [47] Yann Lecun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature* 2015 521:7553, 521:436–444, 5 2015.
- [48] Peter Lloyd, Ali Kafash Hoshidar, Tomas Da Veiga, Aleks Attanasio, Nils Marahrens, James Henry Chandler, and Pietro Valdastri. A learnt approach for the design of magnetically actuated shape forming soft tentacle robots. *IEEE Robotics and Automation Letters*, 5:3937–3944, 7 2020.
- [49] Darrell Whitley. A genetic algorithm tutorial. *Statistics and Computing*, 4:65–85, 6 1994.
- [50] Thomas Bäck. Evolutionary algorithms. *ACM SIGBIO Newsletter*, 12(2):26–31, 1992.
- [51] Pradnya A. Vikhar. Evolutionary algorithms: A critical review and its future prospects. *Proceedings - International Conference on Global Trends in Signal Processing, Information Computing and Communication, ICGTSPICC 2016*, pages 261–265, 6 2017.
- [52] Andrew N. Sloss and Steven Gustafson. 2019 evolutionary algorithms review. pages 307–344, 2020.
- [53] John H Holand. Adaptation in natural and artificial systems. *Ann Arbor: The University of Michigan Press*, 32, 1975.
- [54] Kenneth Alan De Jong. *An analysis of the behavior of a class of genetic adaptive systems*. University of Michigan, 1975.

- [55] Thomas Bäck and Hans-Paul Schwefel. An overview of evolutionary algorithms for parameter optimization. *Evolutionary Computation*, 1:1–23, 3 1993.
- [56] David E Goldberg et al. *Real-coded genetic algorithms, virtual alphabets and blocking*. Citeseer, 1990.
- [57] Francisco Herrera, Manuel Lozano, and Jose L. Verdegay. Tackling real-coded genetic algorithms: Operators and tools for behavioural analysis. *Artificial intelligence review*, 12:265–319, 1998.
- [58] Thomas Bartz-Beielstein, Jürgen Branke, Jörn Mehnen, and Olaf Mersmann. Evolutionary algorithms. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 4:178–195, 2014.
- [59] Stefan Kern, Sibylle D. Müller, Nikolaus Hansen, Dirk Büche, Jiri Ocenasek, and Petros Koumoutsakos. Learning probability distributions in continuous evolutionary algorithms - a comparative review. *Natural Computing*, 3:77–112, 2004.
- [60] Jiachen Zhang and Eric Diller. Millimeter-scale magnetic swimmers using elastomeric undulations. In *IEEE International Conference on Intelligent Robots and Systems*, volume 2015-December, 2015.
- [61] Larry J. Eshelman and J. David Schaffer. Real-coded genetic algorithms and interval-schemata. *Foundations of Genetic Algorithms*, pages 187–202, 1992.