ON MATHEMATICAL OPTIMIZATION, SOFTWARE AND ARTIFICIAL INTELLIGENCE AS FORCE MULTIPLIERS IN MODERN MILITARY APPLICATIONS

Amelia Bucur and Adrian Nicolae Branga
Lucian Blaga University of Sibiu, Department of Mathematics and Informatics, Faculty of Sciences, Sibiu, Romania,

Abstract
Mathematical optimization plays a key role in supporting decision-making processes in the modern military domain, where operational complexity, resource constraints, and information uncertainty require rigorous and efficient solutions. This article highlights how numerical optimization methods, implemented through specialized software and complemented by artificial intelligence techniques, act as force multipliers in contemporary military applications. Optimization can be applied in logistics, mission planning, resource allocation, and command and control systems, highlighting the advantages of hybrid approaches that combine mathematical rigor with the adaptability of machine learning algorithms. In addition, the challenges associated with implementing these technologies are highlighted, including issues related to scalability, interpretability, and human control over AI-assisted decisions. The results suggest that the responsible integration of mathematical optimization, software, and artificial intelligence significantly contributes to increasing operational efficiency and decision-making superiority in modern military environments.
Keywords: optimization, military strategies, artificial intelligence, software

1. Introduction

Mathematical optimization—including linear and nonlinear programming, multi-objective optimization, robust optimization, and meta-heuristic algorithms—has become a central tool for planning and conducting modern military operations. At the strategic level, optimization models allow decision makers to allocate limited resources (forces, ammunition, logistics, flight time, etc.) in a way that maximizes the probability of success and minimizes operational risks and costs. In logistics, routing and scheduling algorithms underlie the planning of transportation, reloading, and redistribution of raw materials in a theater of operations, reducing delivery times and convoy vulnerability. Such applications are frequently studied in Operations Research, Optimization Techniques, and Defense Decision Theory.[1]

At the tactical and operational levels, optimization helps generate and evaluate alternatives through multi-objective methods: the commander can compare solutions that trade-off efficiency, risk of loss, resource consumption, and civilian impact. Interactive optimization methods facilitate human involvement in the decision-making loop, allowing the selection of acceptable solutions from the Pareto front, rather than imposing a single "optimal solution" that may be inappropriate for tactical or ethical reasons. [2]

Cybersecurity and command and control also benefit from optimization: robust architectures and adaptive algorithms can be designed to maintain communication and decision functionality under conditions of deliberate degradation or loss of information. Recent work emphasizes the need to rethink command and control systems from a resilience and modularity perspective, and optimization tools contribute to capability sizing and redundancy allocation. [3]

In parallel, more and more applications are emerging that combine optimization with artificial intelligence: predictive optimization for maintenance, automated mission planning for autonomous platforms, and optimization algorithms inspired by military tactics (meta-heuristics sometimes called after military concepts). However, recent literature draws attention to the limitations — incorrect data, models that oversimplify adversary behavior, and the risk of automating ethically sensitive decisions — and recommends maintaining robust human oversight and explicit ethical constraints in mathematical formulations. [4]

In recent decades, artificial intelligence (AI) has become an essential tool in the fields of exact, technical, military sciences, etc., profoundly influencing the way in which mathematical models are formulated, analyzed, and validated. If in the past mathematical modeling relied exclusively on researchers' intuition and traditional analytical methods, today AI offers a completely new set of techniques that allow the exploration of extremely complex systems, impossible to treat by conventional means. For this reason, the use of AI in mathematical modeling represents a major direction of development in both research and industry, as well as in any other scientific field.

A first advantage of AI is the ability to manage very large amounts of data. Machine learning algorithms can identify subtle relationships or hidden patterns in data sets that would be difficult or impossible to observe using classical mathematical methods. For example, neural networks can approximate functions much more complex than those that can be described analytically, being used in physics, biology, chemistry or economics to formulate predictive models. This ability transforms AI into a powerful ally of mathematicians, not replacing their work, but expanding the potential for analysis.

Another essential aspect is the flexibility of AI. While traditional mathematical models are rigid and require significant simplifications to become tractable, AI-based models allow for the approach of nonlinear, stochastic or chaotic phenomena. For example, in military strategies, fluid dynamics or the study of climate systems, etc., AI can generate hybrid models that combine physical laws with data learning, considerably increasing the accuracy of simulations. Thus, AI becomes a complementary tool to numerical methods, contributing to obtaining more robust solutions that are better adapted to reality.

AI has also brought benefits to mathematical optimization. Techniques such as genetic algorithms, particle swarm optimization or deep neural networks can find optimal solutions in very large search spaces, where classical methods are either inefficient or fail. These approaches are used in military logistics, in the design of engineering structures, in resource management, in industry, etc., where optimization plays a crucial role.

However, the use of AI in mathematical modeling also brings challenges. A critical aspect is interpretability. Black-box models, such as many neural networks, often provide accurate results but are difficult to explain. In mathematics, understanding the internal mechanisms of a model is as important as the result itself, and the lack of transparency can limit applicability in some areas, especially in those where rigorous validation is essential. Therefore, a major research effort is dedicated to developing explainable AI (XAI) methods, which make models more transparent and easier to analyze.

Another challenge is data dependency. AI models are only as good as the data used to train them. In situations where data is insufficient, incomplete, or biased, AI can lead to erroneous models, in contrast to traditional mathematical models that are based on fundamental principles. Therefore, the use of AI requires special attention to the quality and representativeness of data.

Optimization problems occupy a central place in science, engineering, economics, and computer science, aiming to identify the best solution from a large set of possibilities, under well-defined conditions. From optimizing transportation routes and allocating resources to training artificial intelligence models and designing new materials, these problems quickly become extremely complex as their size increases. In this context, quantum computers promise a paradigm shift in the numerical solution of optimization problems, overcoming the limitations of classical computing.

Classical computers solve optimization problems using deterministic or heuristic algorithms, such as gradient methods, genetic algorithms, or simulated annealing. Although efficient for many practical applications, these methods can become inefficient or even impossible to use for problems with a very large number of variables or with a highly irregular objective function landscape characterized by numerous local minima. Computational complexity increases exponentially, and the time required to obtain an acceptable solution becomes prohibitive.

Quantum computing is based on fundamental principles of quantum mechanics, such as superposition, interference, and entanglement. Unlike classical bits, which can only have the values 0 or 1, quantum bits (qubits) can exist in a superposition of these states. This property allows for the simultaneous exploration of a large number of possible solutions, offering a significant potential advantage in combinatorial optimization problems.

One of the most studied frameworks for quantum optimization is the Quantum Approximate Optimization Algorithm (QAOA). It combines ideas from quantum computing and classical optimization, using a parameterized quantum circuit whose results are iteratively improved using a classical algorithm. QAOA is particularly promising for solving Max-Cut problems, constraint satisfaction, or resource planning, where the solution space is discrete and very large.

Another relevant example is quantum annealing, a technique inspired by classical simulated annealing, but which exploits quantum tunneling to avoid getting stuck in local minima. Devices developed by companies such as D-Wave use this principle to address optimization problems formulated in the form of Ising models or QUBO (Quadratic Unconstrained Binary Optimization). Although these systems are not universal quantum computers, they already demonstrate the potential of quantum approaches in real applications.

However, the use of quantum computers in numerical optimization faces significant challenges. Current technology is in the era of noisy intermediate-scale quantum (NISQ) devices, characterized by a limited number of qubits and high errors. Also, not all optimization problems automatically benefit from a quantum advantage, and identifying cases where it is real and significant remains an active topic of research.

It can be said that quantum computers currently offer a revolutionary perspective on the numerical solution of optimization problems, promising substantial speedups for certain classes of problems. Although humanity is still at the beginning of this journey, rapid advances in hardware, algorithms, and the integration of quantum and classical methods suggest that quantum optimization will play an important role in the future of scientific and industrial computing. As the technology matures, its impact could redefine the limits of what is computationally possible. [6,7]

2. Literature Review

Mathematical optimization is a fundamental pillar of decision analysis in the military field, being used extensively in operational planning, logistics, resource allocation, command and control, autonomous systems, etc. Contemporary literature highlights an increasingly pronounced convergence between classical optimization software (based on mathematical programming) and artificial intelligence (AI) techniques, especially machine learning (ML) and reward learning (RL), to address the complexity, uncertainty and dynamics of modern military environments [1].

In terms of activities with students in the field of mathematical optimization application, through careful guidance, the teacher can act as a mediator of knowledge, adapting support to the student's developmental level and encouraging a gradual transition to autonomous learning. Regarding autonomous learning, the student establishes appropriate learning strategies and self-assesses his learning outcomes with minimal external guidance from the teacher. By autonomous learning, in addition to the classical one, we mean student-centered learning, by shifting the focus of education from teaching to learning. Autonomous learning would allow the student to personalize their learning program based on personal strengths and interests and to monitor their own achievements, according to curricular requirements. “Modern society needs specialists capable of acting quickly, creatively and independently. The main means of training these qualities is the autonomous cognitive activity” of students (adapted from https://edict.ro/invatarea-autonoma-un-imperativ-al-educatiei/).

Deterministic optimization software, such as linear programming (LP) and nonlinear programming (NLP) solvers, remain widely used in critical military applications, including logistics planning, force deployment, and mission scheduling. Recent studies show that advances in exact algorithms and software infrastructure (e.g., parallelization and cloud integration) allow solving strategic-scale problems previously considered intractable [8]. In the military context, these tools offer advantages through guarantees of optimality and decision traceability, essential in environments with strict legal and operational constraints. However, real-world military environments are characterized by deep uncertainty, incomplete information, and adaptive adversaries, limiting the applicability of deterministic methods alone. In response, recent literature explores the integration of AI into optimization processes, giving rise to the paradigms of learning-augmented optimization and prescriptive analytics. Bertsimas and Kallus [9] point out that ML can be used to estimate unknown parameters, generate probabilistic scenarios, and reduce the dimensionality of complex defense decision problems.

Reward learning has attracted significant attention in military applications such as adaptive mission planning, autonomous system coordination, and sequential decision making in conflict. Recent surveys indicate that RL is capable of producing efficient policies in large combinatorial problems, including routing, target assignment, and ISR (Intelligence, Surveillance, and Reconnaissance) resource management, although the lack of optimality guarantees and interpretation difficulties remain major challenges.[10]

An active research direction in the military is hybrid AI–OR (Operations Research) optimization, in which AI assists specific steps of classical algorithms, such as branch selection in branch-and-bound or dynamic parameter tuning of meta-heuristics. The work of Lodi and Zarpellon (2017)[11] demonstrates that such approaches can significantly reduce computational times without compromising mathematical rigor, a critical aspect for time-sensitive military applications.

Recent literature also highlights the ethical and operational implications of using AI in military optimization. Issues such as decision transparency, human control, and robustness to erroneous data are frequently cited as limitations of purely AI-based approaches [12]. Consequently, the emerging consensus is that military systems should adopt hybrid frameworks, in which rigorous optimization software is complemented, but not replaced, by machine learning mechanisms.

The present article is organized as follows. The first section contains an introduction to mathematical modeling of optimization problems useful in decision theory in the field of military strategies, optimization that can be solved with specific software for solving mathematical problems or even with the consultation of AI. The second section includes an essay on what has already been published in the field. Section 3 contains examples of optimization problems solved with Maple and MATLAB software. In section 4 we present results and discussions, and at the end of the article some conclusions are included.

3.Methods

3.1.Method of using the MATLAB environment in optimization

The MATLAB environment provides a comprehensive suite of tools for solving optimization problems, both for simple functions and for complex systems with many variables and constraints. Thanks to its specialized libraries (such as Optimization Toolbox), MATLAB is an ideal environment for formulating, analyzing, and numerically solving optimization problems.

A simple category of problems is the minimization or maximization of real functions of one variable. MATLAB provides the function **fminbnd**, which allows finding a minimum in a given interval.

Example 3.1.1. *Determining the minimum of the function $f(x) = x^2-4x+5$ on the interval [0,5].*

MATLAB code:

```
f = @(x) x.^2 - 4*x + 5;
[x_min, f_min] = fminbnd(f, 0, 5)
```

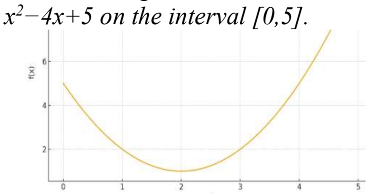

Fig.3.1.1. *Graphical representation*

The function is convex, and MATLAB returns the minimum at the point $x = 2$, with the value $f(2) = 1$. Example 3.1.1 illustrates the efficiency of numerical methods for simple functions, confirming results that can also be obtained analytically.

For functions with several variables, MATLAB uses algorithms such as the quasi-Newton method (the **fminunc** function) or optimization with constraints (**fmincon**).

Example 3.1.2. *Determining the minimum of the nonlinear function $f(x,y) = x^2+y^2+xy-4x-3y+7$.*

MATLAB code: *f = @(v) v(1)^2 + v(2)^2 + v(1)*v(2) - 4*v(1) -*

```
3*v(2) + 7;
    v0 = [0,0];        % punct de start
    [v_min, f_min] = fminunc(f, v0)
```

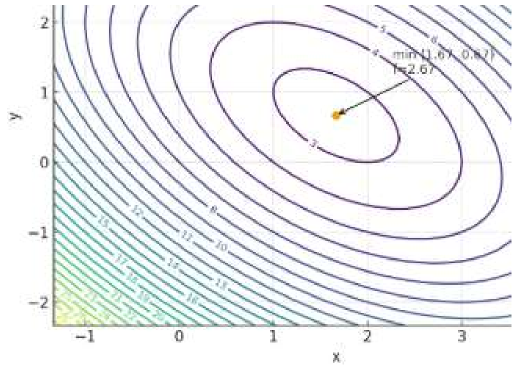

Fig.3.1.2. *The minimum of the function*

The algorithm starts from the initial point and converges to the global minimum, due to the convex nature of the function. The result shows the usefulness of MATLAB in solving multidimensional problems, especially when analytical solutions are difficult to obtain.

Most real-world applications involve equality and inequality constraints. MATLAB provides the **fmincon** function, which is very powerful in conditional optimization.

Example 3.1.3. *Determining the minimum of the function* $f(x,y) = (x-1)^2 + (y-2)^2$ *with the constraint*: $x + y \geq 3$.

Cod MATLAB:

```
f = @(v) (v(1)-1)^2 + (v(2)-2)^2;
v0 = [0,0];
A = [-1 -1];      % scriem x + y >= 3 ca -x - y <= -3
b = -3;
[v_min, f_min] = fmincon(f, v0, A, b)
```

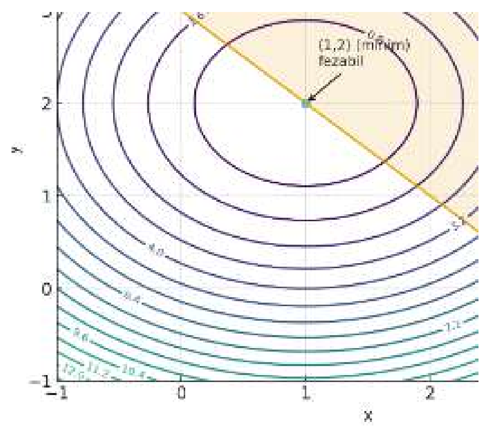

Fig.3.1.3. *The minimum of the function*

The result shows the point on the line $x + y = 3$ closest to the point (1,2), thus demonstrating the ability of MATLAB to efficiently solve problems with linear constraints.

For linear optimization, MATLAB uses the **linprog** function.

Example 3.1.4. *Solving the constrained linear programming problem*: $\min(-3x - 4y), x + 2y \leq 8, 3x + y \leq 9, x, y \geq 0$.

MATLAB code:

```
f = [-3; -4];
A = [1 2; 3 1];
b = [8; 9];
lb = [0; 0];
[x_opt, f_opt] = linprog(f, A, b, [], [], lb)
```

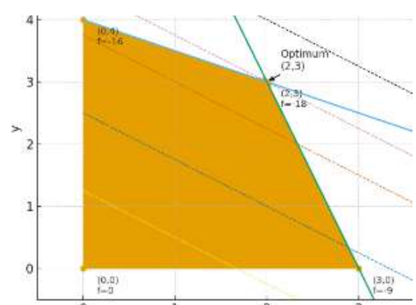

Fig.3.1.4. *The minimum of the function*

This problem is classic in economics and logistics, and MATLAB provides an instant, numerically efficient solution.

Also, for functions with multiple local minima, MATLAB offers the tools from the *Global Optimization Toolbox*, such as: ga (genetic algorithms), simulannealbnd (simulated cooling), patternsearch.

Example 3.1.5. *Determining the global minimum of the Rastrigin function*: $f(x,y) = 20 + x^2 + y^2 - 10[cos(2\pi x) + cos(2\pi y)]$. The Rastrigin function is a standard, non-convex test function used in global optimization, known for its numerous local minima, which make it difficult for gradient-based algorithms, having a unique global minimum at the origin *(0, 0, ...)* where the value of the function is 0, and a generalized mathematical formulation involving terms x_i^2 and $\cos(12x_i)$. For 2 variables, it has the well-known graphical representation in Fig. 3.1.5.

MATLAB code:

```
f = @(v) 20 + v(1)^2 + v(2)^2 - 10*(cos(2*pi*v(1)) + cos(2*pi*v(2)));
opts = optimoptions('ga','Display','off');
[v_min, f_min] = ga(f, 2, [], [], [], [], [-5 -5], [5 5], [], opts)
```

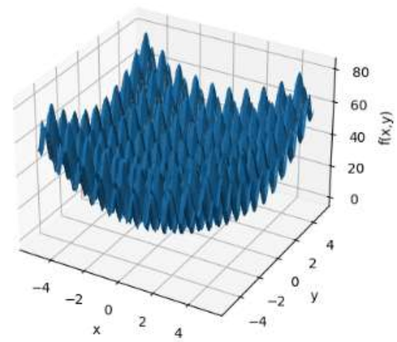

Fig.3.1.5. *Minima of the function in example 3.1.5*

This example highlights the applicability of MATLAB in solving complex problems encountered in artificial intelligence, robotics, or industrial optimization.

3.2. The method of using Maple software in optimization

Example 3.2.1. *Solving the linear optimization problem*: $\min(4x_1 + 3x_2 + 5x_3)\,, x_1 - 2x_2 + 3x_3 \geq 1, x_1 + x_2 - x_3 \geq 1,\ x_j \geq 0, j = 1,2,3.$

```
> with(simplex) :
> minimize( 4·x1 + 3·x2 + 5·x3, {x1 − 2·x2 + 3·x3 ≥ −1, x1 + x2 − x3 ≥ 1}, NONNEGATIVE );
```

$$\left\{x1 = \frac{1}{3}, x2 = \frac{2}{3}, x3 = 0\right\}$$

Fig.3.2.1. *Introducing Example 3.2.1 into Maple and solving the problem*

Example 3.2.2. *Solving the nonlinear optimization problem*: $\min\left[\frac{1}{2}(10x_1^2 + 5x_2^2 + 8x_3^2 - 2x1x_2 + 16x_1x_3 + 4x_2x_3) + 3x_1 + 5x_2 - 4x_3\right], 9x_1 + 4x_2 + 10x_3 = 4, x_1 \geq 0, x_2 \geq 0, x_3 \geq 0.$

```
with(Optimization) :
NLPSolve( 1/2 (10x1^2 + 5·x2^2 + 8x3^2 − 2x1·x2 + 16·x1·x3 + 4·x2·x3) + 3x1 + 5x2 − 4x3, {9x1 + 4x2 + 10x3 = 12},
    assume = nonnegative)
                [0.960000000000000852, [x1 = 0., x2 = 0., x3 = 1.20000000000000]]
```

Fig.3.2.2. *Introducing Example 3.2.2 into Maple and solving the problem*

Example 3.2.3. Graphical highlighting of optima for nonlinear optimization problem: $mi\,n(max)\cos\big(\pi(x + y)\big), x^2 + y^2 \geq 1, x, y \in [-3,3]$.

Maple procedure:

```
restart:
f := (x,y) -> piecewise(x^2+y^2 >= 1, cos(Pi*(x+y)), undefined):
plot3d(
  f(x,y),
  x = -3..3,
  y = -3..3,
  axes = boxed,
  labels = ["x","y","f(x,y)"],
  shading = zhue,
  title = "f(x,y)=cos(Pi*(x+y)), x^2+y^2 >= 1"
);
```

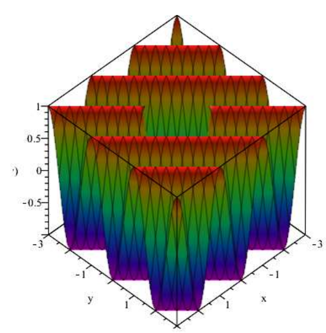

Fig.3.2.3. *Entering Example 3.2.3 into Maple and 3D graphical representation of the problem*

Observations. Only conditional extrema appear on the boundary, not local ones. The number of local optimum points is 0. The global maximum is $f_{max} = 1$, and is obtained for $x + y = 2k$, k being an integer (due to the cosine function). And the global minimum is $f_{min} = -1$, and is obtained for $x + y = 2k+1$, k – an integer (due to the cosine). The general solution for the optimum points satisfying the conditions $x + y = k$ and $xy = (k^2 - 1)/2$, where k is an integer. The values of k can be specified by setting the initial conditions of the problem, $x, y \in [-3,3]$. [13]

4. Results and Discussion

Software for quantum physics, quantum engineering and quantum chemistry, in harmony with mathematical modeling and AI-assisted simulation, will bring new discoveries, such as the discovery of new molecules, new drugs, etc., with optimal error management.

The MATLAB environment is an extremely powerful tool for numerically solving optimization problems, offering: functions dedicated to both simple and complex optimization; efficient algorithms for unconstrained or conditional problems; methods for global optimization; high flexibility in defining functions and constraints; an ideal interactive environment for experimentation, visualization and analysis.

Due to these features, MATLAB is also a standard in numerical optimization, both in the academic environment (for students) and in industrial applications (for engineers) or scientific (for researchers).

Maple is also another example of software useful in numerically solving optimization problems because it provides powerful tools for symbolic and numerical computation, allowing the efficient formulation, analysis, and solution of complex mathematical models.

Through its dedicated functions, Maple facilitates the implementation of optimization algorithms, visualization of results, and verification of obtained solutions.

6. Conclusions

In conclusion, mathematical optimization offers a powerful set of tools for improving efficiency, resilience, and decision-making transparency in the military. Responsible application requires transparency, rigorous model validation, test scenarios, and clear human control mechanisms so that operational advantages do not compromise ethical considerations or civilian security.

The specialized literature indicates that applications of optimization software and AI in the military are evolving towards integrated, adaptive, and robust systems capable of supporting complex decisions in contested environments. Future research focuses on explainability, validation, and the integration of ethical constraints directly into optimization models, which are essential aspects for the responsible use of these technologies in a military context.

Artificial intelligence and the use of software specific to solving mathematical problems represent a revolution in the field of mathematical modeling, providing powerful tools for studying complex phenomena and generating effective solutions. Although it does not replace traditional mathematical modeling, AI and the use of software specific to solving mathematical problems complement and amplify it, opening new perspectives in research and industry. The future of mathematical modeling will undoubtedly be a hybrid one, in which classical methods, software and AI collaborate to provide a deeper and more accurate understanding of the world around us.

7. References

[1] Bengio, Y., Lodi, A., & Prouvost, A. (2021). Machine learning for combinatorial optimization: A methodological tour d'horizon. *European Journal of Operational Research, 290*(2), 405–421.

[2] Bertsimas, D., & Kallus, N. (2022). From predictive to prescriptive analytics. *Management Science, 68*(2), 1025–1044.

[3] Poti, B. & Reale, E. (2007). Changing allocation models for public research funding: an empirical exploration based on project funding data. *Science and Public Policy, 34 (6*), 417-430.

[4] Pathan, R.R., Yatheesh, K.C., Pati, S., Reddy, K.K., Jothimani. S. & Chinchorkar, S. (2024). Mathematical Models in Artificial Intelligence: Optimizing Algorithms for Big Data Analysis in IT Systems. *Journal of Information Systems Engineering and Management, 10(228)*, 804-815.

[6] Simon, D.R. (1997). On the Power of Quantum Computation. *SIAM Journal on Computing*, *26(5)*, 1474–1483.

[7] Watrous, J. (2006). *Lecture Notes on Quantum Computing*, University of Waterloo.

[8] Bixby, R. E., Fenelon, M., Gu, Z., Rothberg, E., & Wunderling, R. (2023). Advances in mixed-integer programming solvers. *INFORMS Journal on Computing, 35*(1), 1–18.

[9] Bertsimas, D., & Kallus, N. (2022). From predictive to prescriptive analytics. *Management Science, 68*(2), 1025–1044.

[10] Mazyavkina, N., Sviridov, S., Ivanov, S., & Burnaev, E. (2021). Reinforcement learning for combinatorial optimization: A survey. *Computers & Operations Research, 134*, 105400.

[11] Lodi, A., & Zarpellon, G. (2017). On learning and branching. *INFORMS Journal on Optimization, 25(2)*, 207-236.

[12] Gasse, M., Chételat, D., Ferroni, N., Charlin, L., & Lodi, A. (2022). Exact combinatorial optimization with graph neural networks. *Nature Machine Intelligence, 4*, 135–144.

[13] Bucur, A. & Branga, A. N. (2024). *Theory and Applications of Numerical Approximation Techniques*. UK: Cambridge Scholars Publishing.