OPTIMIZATION OF MACHINING PROCESSES USING METAHEURISTIC ALGORITHMS - A COMPARATIVE ANALYSIS

THESIS SUBMITTED BY

DEVENDRA GAJANAN PENDOKHARE

DOCTOR OF PHILOSOPHY (ENGINEERING)

DEPARTMENT OF PRODUCTION ENGINEERING
FACULTY COUNCIL OF ENGINEERING & TECHNOLOGY
JADAVPUR UNIVERSITY
KOLKATA, INDIA
2025

JADAVPUR UNIVERSITY
KOLKATA-700032, INDIA

Index No. 211/22/E
Registration No. 1012216001

TITLE OF THE PHD (ENGG.) THESIS:

OPTIMIZATION OF MACHINING PROCESSES USING METAHEURISTIC ALGORITHMS - A COMPARATIVE ANALYSIS

NAME, DESIGNATION & INSTITUTION OF SUPERVISOR:

DR. SHANKAR CHAKRABORTY
Professor, Production Engineering Department
Jadavpur University
Kolkata-700032, India

LIST OF PUBLICATIONS

Title of the Thesis : **OPTIMIZATION OF MACHINING PROCESSES USING METAHEURISTIC ALGORITHMS - A COMPARATIVE ANALYSIS**

Name, Designation and Institute of the Supervisor: : **Dr. Shankar Chakraborty**
Professor
Department of Production Engineering
Jadavpur University, Kolkata, India

List of Publications:

(a) In International Journals

Sl. No.	Title of the paper	Name of the Journal
1	Optimizing plasma arc cutting processes using physics-based metaheuristic algorithms: a comparative analysis	International Journal on Interactive Design and Manufacturing, 2024; 1-35 DOI: 10.1007/s12008-024-02136-y
2	A comparative analysis on parametric optimization of abrasive water jet machining processes using foraging behavior-based metaheuristic algorithms	OPSEARCH, 2025; 1-42, DOI: 10.1007/s12597-025-00916-y
3	A comparative analysis of preying behavior-based metaheuristic algorithms for optimization of laser beam drilling processes	Decision Analytics Journal, 2024; 10:100412
4	A review on multi-objective optimization techniques of wire electrical discharge machining	Archives of Computational Methods in Engineering, 2024; 1-43, DOI: 10.1007/s11831-024-10195-3
5	Human-inspired metaheuristics-based optimization of turning of AISI 6061-T6 aluminium	Journal of The Institution of Engineers (India): Series C, 2025; 1-28, DOI: 10.1007/s40032-025-01185-w

(b) List of Papers Presented/Accepted in International Conferences

Sl. No.	Title of the paper	Name of the Conference
1	Comparative study of multi-response parametric optimization of EDM processes using preying behaviour metaheuristic algorithms	9th International & 30th All India Manufacturing Technology, Design and Research Conference (AIMTDR-2023), 8-10 December 2023, IIT (BHU), Varanasi
2	Parametric optimization of conventional drilling processes using human-based metaheuristic algorithms: A comparative analysis	International Conference on Design and Manufacturing Technologies (ICDMT-2024), 8-10 November 2024, Punjab Engineering College, Chandigarh
3	Multi-response parametric optimization of WEDM processes using foraging behaviour metaheuristic algorithms: A comparative analysis	13th International Conference on Precision, Meso, Micro and Nano Engineering (COPEN-2024), 13-15 December 2024, NIT, Calicut

(c) List of Patents: Nil

JADAVPUR UNIVERSITY

FACULTY OF ENGINEERING & TECHNOLOGY

DEPARTMENT OF PRODUCTION ENGINEERING

CERTIFICATE FROM THE SUPERVISOR

This is to certify that the thesis entitled "**OPTIMIZATION OF MACHINING PROCESSES USING METAHEURISTIC ALGORITHMS - A COMPARATIVE ANALYSIS**" submitted by **Mr. Devendra Gajanan Pendokhare**, who got his name registered on 20.12.2022 for the award of Ph.D. (Engg.) degree of Jadavpur University is absolutely based upon his own work under the supervision of **Prof. Shankar Chakraborty** and that neither his thesis nor any part of the thesis has been submitted for any degree/diploma or any other academic award anywhere before.

.....Shankar Chakraborty.....

21/07/23

(Dr. Shankar Chakraborty)

Thesis Advisor

Department of Production Engineering

JADAVPUR UNIVERSITY

Kolkata-700032

Dr. Shankar Chakraborty
Professor
Department of Production Engg.
Jadavpur University
Kolkata – 700 032

PROFORMA – 1

"Statement of Originality"

I, **Devendra Gajanan Pendokhare,** registered on 20.12.2022 do hereby declare that this thesis entitled "**OPTIMIZATION OF MACHINING PROCESSES USING METAHEURISTIC ALGORITHMS - A COMPARATIVE ANALYSIS**" contains literature survey and original research work done by the undersigned candidate as part of Doctoral studies.

All information in this thesis have been obtained and presented in accordance with existing academic rules and ethical conduct. I declare that, as required by these rules and conduct, I have fully cited and referred all materials and results that are not original to this work.

I also declare that I have checked this thesis as per the "Policy on Anti Plagiarism, Jadavpur University, 2019", and the level of similarity as checked by iThenticate software is **7%**.

.....

Signature of the Candidate

Date: 21/07/2025

Certified by Supervisor

.....

(Dr. Shankar Chakraborty)

Thesis Advisor

Department of Production Engineering

JADAVPUR UNIVERSITY

Kolkata-700032

21/07/25

Dr. Shankar Chakraborty
Professor
Department of Production Engg.
Jadavpur University
Kolkata – 700 032

ACKNOWLEDGEMENT

The insight of this thesis is certainly due to the contribution of several outstanding persons. The author wishes to express his deep sense of gratitude towards the thesis advisor Dr. Shankar Chakraborty, Professor, Department of Production Engineering, Jadavpur University, Kolkata for his constant guidance, helpful suggestions, continuous association, support, encouragement and valuable advice at every aspect from the sprouting stage to the development stage of this research work. Without his constructive and timely advice, this thesis would have not been progressed as smoothly as it did towards its completion. The author indeed owes to him for his patience and valuable time which he has spent for the present research work.

The author gratefully acknowledges the cooperation and encouragement received from Dr. B. Bhattacharyya, QIP Coordinator and former HOD, Dr. Bijan Sarkar, Professor and HOD, Dr. Biswanath Doloi, Professor, Dr. D. Banerjee, Professor, Dr. Biplab Ranjan Sarkar, Professor and Dr. Ajoy Datta, Professor, and all other faculty members of Production Engineering Department, Jadavpur University during the course of the research. The author would like to take this opportunity to thank Dr. Kanak Kalita, Associate Professor, Department of Mechanical Engineering, Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology, Avadi, India who supported all the time during the developmental and experimentation stages.

The author gratefully acknowledges the Head, Department of Production Engineering, Jadavpur University, for providing the laboratory facilities. Thanks are also extended to the FET office and Research section for their cordial assistance and administrative support.

Thanks are extended to the All India Council for Technical Education, New Delhi for QIP (Poly) scheme. The author also extends heartiest thanks to Government of Maharashtra for deputing him to the PhD research work for three years. The author extends heartiest thanks to Dr. Vinod Mohitkar, Director and Shri Pramod Naik, Joint Director, Directrorate of Technical Education, Mumbai. Also, the author heartfully thanks Dr. D. R. Mahajan, Principal, Government Polytechnic, Thane and Dr. N. M. Mhala, Principal, Government Polytechnic, Mumbai, for their constant support to carry out the research work under QIP (Poly) scheme. The author sincerely thanks all the colleagues and staff members of Department of Mechanical Engineering, Government Polytechnic, Thane.

Appreciation is also due to all the fellow colleagues, Mr. Sudeep Santra, Dr. Naresh Besekar, Dr. Himadri Panda, Dr. Santosh Kumar, Dr. Mohit Pandey, Mr. Vikram Titarmare, Mr. Sevakram Kumbhare, Mr. Deep Saha, Mr. Manoj Gajbhiye, Mr. Saikat Chatterjee, Mrs. Samriddhya Ray Chowdhury, Mr. Suresh Kadam, Miss. Shruti Dandge, Mr. Somnath Nandi, Mr. Souvik Karmakar, Mr. Arkporvo Roy and Mr. Sayan Aacharya for their constant cooperation, useful assistance and support during this research work.

The author is profoundly grateful to his parents for their steadfast belief in education and unwavering support for his academic pursuits. A special note of thanks to my daughter, whose warm

wishes and constant encouragement served as a source of inspiration. The author also extends his deepest gratitude to all his family members for their persistent motivation throughout this research work. Without their understanding, none of this would have been possible. It would have been very much difficult to complete this work without the support of my wife Poonam, who sacrificed her quality time during this tenure, being a primary source of strength and inspiration for me.

Finally, the author more sincerely acknowledges all those who contributed, directly or indirectly, to the success of this research work. With heartfelt gratitude, I bow my head to the Almighty for granting me the courage, strength and patience, necessary to complete this endeavor successfully.

(Devendra Gajanan Pendokhare)

VITA

The author, Mr. Devendra Gajanan Pendokhare, son of Mrs. Lata and Mr. Gajanan Maroti Pendokhare was born on 12th April 1988 in a town, Anjanagaon Surji in the district of Amravati, Maharashtra. The author passed S.S.C. examination from Sitabai Sangai High School, Anjangaon Surji, District-Amravati under the State Board of Secondary Education in the year 2003, followed by an ITI (Industrial Training Institute) certification in Electronics Mechanic from Government ITI, Anjangaon Surji under the Board of Directorate of Vocational Education and Training, Mumbai in the year of 2005. He then pursued a Diploma in Mechanical Engineering from Government Polytechnic, Amravati (an Autonomous Institute of Government of Maharashtra) with first class in the year 2008. The author earned his B.E. degree in Mechanical Engineering in the year 2011 from Government College of Engineering, Chandrapur under Nagpur University with first class, followed by M.Tech. in Mechanical Engineering, specialization Manufacturing Systems Engineering from Saraswati College of Engineering, Kharghar, New Mumbai under Mumbai University in 2015 with first class distinction having a CGPA of 7.72.

The author is working as a Lecturer in Mechanical Engineering Department, Government Polytechnic, Thane from January 2012. The author has joined as a Research Scholar in June 2022 in Production Engineering Department, Jadavpur University, Kolkata, under All India Council of Technical Education's scheme of the Quality Improvement Programme (QIP Poly). The author has been working in the area of optimization with a PhD thesis title "Optimization of machining processes using metaheuristic algorithms - A comparative analysis". In December 2022; he got his name registered for Ph.D. (Engg.) in the Department of Production Engineering. He has been pursuing his research activities in the said department for completion of his Ph.D. (Engg.) work.

The author has been engaged in research activities since 2022 onwards. He has published research papers in several international journals and presented his works in the international conferences.

PREFACE

The machining and manufacturing industry has undergone significant transformations with the integration of optimization techniques to enhance efficiency, reduce costs and improve product quality. The growing demand for precision, efficiency and sustainability in machining processes has led to extensive research in both conventional machining and non-traditional machining (NTM) techniques. As modern engineering components have become increasingly complex, industries, such as aerospace, automotive, electronics and medical etc. seek innovative solutions to enhance their machining capabilities.

Traditional optimization methods, such as gradient-based techniques and trial-and-error approaches, often struggle to find out globally optimal solutions. These methods can become trapped into local optima and fail to handle complex, multi-objective optimization problems, involving trade-offs between material removal rate (MRR), surface roughness (SR), tool wear rate (TWR) and specific cutting energy consumption (SCEC). Consequently, metaheuristic algorithms have emerged as powerful tools for optimizing machining parameters, improving productivity, product quality and environmental sustainability while offering robust and adaptable solutions for both conventional machining and NTM processes.

Metaheuristic algorithms, inspired by natural phenomena, such as evolution, swarm intelligence and biological processes, can efficiently navigate through complex search spaces to achieve near-global optimal solutions. Their flexibility, adaptability and robustness make them well-suited for solving machining optimization problems.

This research work presents a comprehensive investigation into parametric optimization across five distinct traditional machining and NTM processes using nature-inspired metaheuristic algorithms. By integrating advanced computational techniques, this research aims to optimize machining parameters for improved efficiency, reduced energy consumption (EC), minimal TWR, enhanced SR and lower environmental impact. It systematically examines five categories of metaheuristic algorithms, including human-inspired algorithms, mating behavior-based algorithms, preying behavior-based algorithms, foraging behavior-based algorithms and physics-based algorithms.

A key contribution of this research is its emphasis on attaining sustainability in manufacturing, by optimizing different machining parameters. The research promotes green machining, reduces carbon footprints, enhances occupational safety and health standards, supporting adoption of environmental friendly and resource-efficient manufacturing practices. This structured framework bridges the gap between theoretical optimization and practical machining applications, guiding selection and application of the most suitable metaheuristic algorithms for different machining processes.

A robust methodological framework is established to evaluate the effectiveness of various metaheuristic algorithms in optimizing machining parameters. The research process involves the following key steps:

i) Selection of the machining processes: Five machining processes, both traditional machining (turning and milling) and NTM techniques, like laser beam machining (LBM), abrasive water jet machining (AWJM) and plasma arc cutting (PAC) are chosen for optimization to represent different aspects of modern manufacturing.

ii) Choice of the metaheuristic algorithms: This study evaluates effectiveness of different nature-inspired metaheuristic algorithms in optimizing machining parameters. The comparative performance of these algorithms is assessed based on solution accuracy, variability, computational time and convergence speed. Five categories of nature-inspired algorithms are picked out, with each category comprising five specific algorithms. These include:

   a) Human-inspired algorithms: Teaching learning-based optimization (TLBO), search and rescue optimization (SAR), teamwork optimization algorithm (TOA), human conception optimizer (HCO) and queuing search algorithm (QSA).
   b) Mating behavior-based algorithms: Firefly mating algorithm (FMA), naked mole-rat algorithm (NMR), butterfly mating optimization algorithm (BOA), barnacle mating optimizer algorithm (BMOA) and American zebra optimization algorithm (AZOA).
   c) Preying behavior-based algorithms: Sailfish optimizer (SFO), Harris Hawks optimizer (HHO), Aquila optimizer (AO), bat optimizer (BA) and grey wolf optimizer (GWO).
   d) Foraging behavior-based algorithms: Dragonfly optimizer (DOA), African vultures optimizer (AVOA), grasshopper optimization algorithm (GOA), fruit fly optimizer (FOA) and bird swarm optimizer (BSA).
   e) Physics-based algorithms: Archimedes optimization algorithm (AOA), atom search optimization (ASO), nuclear reaction optimization (NRO), electromagnetic field optimization (EFO) and gravitational search algorithm (GSA).

iii) Optimization of the machining parameters: The selected algorithms are applied to optimize various critical machining parameters, including cutting speed (CS), feed rate (FR), depth of cut (DOC), TWR, SCEC and MRR for enhanced machining performance.

iv) Performance assessment: These algorithms are evaluated based on some key performance metrics, such as solution accuracy, variability, computational time and convergence speed. The results highlight superior optimization capabilities of the specific algorithms in different machining processes, providing valuable insights for selecting the most suitable optimization techniques in industrial applications. Multi-objective optimization problems are solved using Pareto optimal fronts to identify the best trade-off solutions.

v) Statistical validation: To ensure reliability of the derived findings, advanced statistical validation techniques, including non-parametric tests, are employed to confirm significance of the results, ensuring robustness of the solutions.

   a) Friedman’s mean rank test, and
   b) Wilcoxon rank sum test.

Through extensive experimentation and statistical validation, this research identifies the most effective metaheuristic algorithms for different machining scenarios. The results demonstrate that the specific algorithms outperform others in achieving the optimal parameter settings, ultimately improving machining efficiency and sustainability.

Furthermore, this study validates the superior performance of the top five algorithms across the considered categories based on an end milling process. This reinforces the effectiveness and applicability of metaheuristic optimization in real-world manufacturing. The findings provide valuable insights into selecting the most appropriate optimization techniques for various machining applications.

Turning operations, one of the most widely used machining processes in the manufacturing industries, are essential for removal of excess material from the exterior surface of cylindrical workpieces. In this process, achieving an optimal combination of various input parameters is essential for improving product quality, extending tool life and achieving higher machining efficiency. However, optimization of a turning process is challenging due to multiple interacting variables influencing the machining responses. Effective optimization must also consider key resources, such as energy, tools, production time and costs, as they significantly impact environmental sustainability and overall profitability. Additionally, machining parameters, such as CS, FR and DOC must be precisely controlled to maximize MRR, and minimize SR, power consumption (Pc) and flank wear, while ensuring prolonged tool life and energy efficiency.

To address these challenges, this study explores the application of five human-inspired metaheuristic algorithms, i.e. TLBO, SAR, TOA, HCO and QSA using past experimental data to optimize the turning operation on AISI 6061-T6 aluminum. These algorithms leverage human-based problem solving strategies to refine the machining parameters effectively. Their performance is evaluated based on solution accuracy, computational effort and overall efficiency. A Pareto optimal front is developed to identify the optimal parametric intermix for the multi-objective optimization problem, balancing trade-offs between various performance measures. Among the considered algorithms, TLBO emerges as the most effective one, excelling in optimizing MRR, SR and SCEC. Statistical validation using quality metrics, including spacing (SP) and hypervolume (HV), along with critical analysis of performance using Friedman's mean rank test and Wilcoxon rank sum test results, corroborate TLBO's superiority over other techniques. Its ability to achieve consistent, high-quality results highlights its potential for industrial applications where precision and efficiency are of paramount importance.

With a growing emphasis on green and sustainable manufacturing, conventional machining processes, such as turning, drilling and milling are increasingly being replaced by their environment-friendly alternatives. These green processes not only improve workplace safety and health standards, but also contribute to environmental sustainability by reducing EC and carbon footprints. Considering these advancements, industries are actively seeking innovative techniques to have transition towards sustainable machining without compromising productivity. Aligning with this sustainability-driven

approach, this research focuses on optimizing green machining processes, particularly dry milling, while considering minimization of Pc as the primary objective. Five mating behavior-based metaheuristic algorithms, like FMA, NMR, BOA, BMOA and AZOA are employed to optimize the said green machining process. These algorithms mimic the mating behaviors of biological species to enhance the search procedure for optimal solutions in complex optimization problems.

For dry milling of stainless steel 304 material, FMA outperforms its competitors in attaining the optimal machining parameters, significantly improving power factor, SCEC and SR. Its superior exploration-exploitation balance enables more effective convergence towards the optimal settings, reducing variability in the machining outcomes. The superiority of FMA is further validated through two statistical metrics, including SP and HV, along with comparative tests, such as Friedman's mean rank test and Wilcoxon rank sum test. Additionally, comparative assessments against the established metaheuristic algorithms, such as genetic algorithm, particle swarm optimization, artificial bee colony, ant colony optimization and TLBO further confirm FMA's effectiveness in optimizing the said green machining process. These findings demonstrate that mating behavior-based algorithms offer a promising avenue for sustainable manufacturing, enhancing machining efficiency while minimizing environmental impact.

Another significant focus of this research work is the optimization of a laser beam drilling (LBD) process, a crucial technique in the manufacturing of micro-holes for applications in diverse industries, like automotive, aviation, electronics and healthcare. The study investigates applicability of five preying behavior-based metaheuristic algorithms, i.e. SFO, HHO, AO, BA and GWO, in optimizing CO_2 laser-based micro-drilling on polycarbonate material. Optimization targets include laser power, exposure time and focal plane position with the goal of refining hole diameter, hole depth and heat affected zone (HAZ).

The comparative analysis highlights the superior performance of GWO over other preying behavior-based metaheuristics in terms of solution quality, convergence speed and computational efficiency. GWO's simple principle, fast convergence speed and high search accuracy contribute to its superior performance, particularly in achieving better hole quality with minimal HAZ. These findings are further validated through two statistical metrics, including SP and HV, along with statistical evaluations, including Friedman's mean rank and Wilcoxon rank sum tests, which confirm GWO's dominance in LBD process optimization. Given the precision-driven nature of laser drilling, the ability to accurately control its varying parameters using metaheuristic algorithms opens new possibilities for manufacturing applications in micro-engineering and high-performance materials.

Due to hybridization of the material removal mechanisms of abrasive jet and water jet machining processes, AWJM appears as an efficient NTM process providing higher productivity, superior surface quality and excellent dimensional accuracy across diverse work materials irrespective of their mechanical and thermal properties. Due to the complexity of input parameters and conflicting machining responses, optimizing an AWJM is a challenging task requiring sophisticated optimization approaches.

An AWJM process is optimized here using five foraging behavior-based metaheuristic algorithms, i.e. DOA, AVOA, GOA, FOA and BSA. These algorithms simulate foraging strategies of biological species to explore and exploit potential solutions effectively.

Among them, AVOA demonstrates maximum efficiency, outperforming other algorithms in terms of solution accuracy, variability and computational effort. This algorithm's strong exploration capabilities allow for more precise adjustments of key parameters, such as water jet machining, stand-off distance (SOD) and traverse speed, leading to superior machining outcomes. The performance analysis of the considered algorithms is assessed using two quality matrics (SP and HV) and validated through Friedman's mean rank and Wilcoxon rank sum tests. The ability of an AWJM process to machine hard-to-cut materials, coupled with effective optimization, makes it a valuable technique in industries requiring higher precision and minimal thermal damage of the machined componants.

Similarly, PAC has become a flexible and effective method for precisely cutting complex profiles on various difficult-to-machine materials, including superalloys and composites. It offers many benefits, such as superior dimensional accuracy, higher productivity and capability to cut thicker materials. However, optimizing this process is challenging due to involvement of many variables, intricate cutting mechanism, and interactions between process parameters and responses. In this research work, five newly developed physics-based metaheuristic algorithms, i.e. AOA, ASO, NRO, EFO and GSA are proposed for optimizing a PAC process. These algorithms harness the principles of physics, such as atomic interactions and gravitational force, to refine process parameters efficiently.

The results indicate that EFO algorithm outperforms its competitors in achieving the optimal process parameter settings, including CS, gas pressure, arc current and SOD to optimize MRR, kerf taper and HAZ. It effectively balances computing effort, convergence time and solution quality while delivering high-quality cuts with minimal material wastage and lower EC. Statistical assessments, including quality metrics, non-parametric ranking-based evaluations and comparative analysis against some popular state-of-the-art optimization methods, further validate EFO's effectiveness in optimizing the said PAC process.

Finally, investigations are performed on application of five top-performing metaheuristic algorithms, i.e. TLBO, FMA, GWO, EFO and AVOA for parametric optimization of the end milling operation on hybrid Al7075/SiC/Gr composite to find out the best-performing algorithm. Each algorithm is selected based on its suitability for different machining processes, including human-inspired TLBO algorithm for the turning process, mating behavior-based FMA technique for the green machining (particularly for the dry milling process), preying behavior-based GWO algorithm for the LBD process, physics-based EFO algorithm for the PAC and foraging behavior-based AVOA for the AWJM process. Comparative analyses are conducted using statistical non-parametric tests, namely Wilcoxon rank sum test and Friedman's mean rank test, to evaluate the algorithm performance. Additionally, Pareto optimal fronts are developed to obtain unique parametric settings and optimal multi-objective solutions for the said end milling process. Finding out the optimal combination of the end milling parameters and

achieving the desired response values are made easier with the aid of parallel plots. Additionally, the related values of SP and HV metrics are analyzed to identify the best-performing algorithm.

Among the considered algorithms, GWO consistently outperforms others in both single- and multi-objective optimization of the said end milling process. GWO attains the optimal values of MRR, SR and microhardness with the best possible parametric combination of spindle speed, FR, DOC and percentage of SiC/Gr. GWO surpasses its competitors in convergence speed and computing time to find out the near global optimal solution, significantly reducing the runtime compared to AVOA, FMA, EFO and TLBO techniques. The study further highlights that higher SS and DOC, along with moderate FR and SiC/Gr percentage values, result in the optimal performance across multiple machining responses. Overall, GWO proves to be the most efficient algorithm, making it the preferred choice in optimizing the said end milling operation on the hybrid composite.

Despite the promising results obtained through the extensive computational analysis conducted in this research work, a notable limitation exists with respect to absence of real-time confirmatory experiments. Since all the optimization results are based on the past experimental datasets, there is no direct validation of the derived solutions in real-world machining environments. Future research should focus on real-time experimental validation, integrating emerging optimization techniques, and hybridizing metaheuristic algorithms with machine learning models to enhance adaptability and performance in machining processes optimization.

While this research work provides a solid foundation for optimizing machining processes using metaheuristic algorithms, several opportunities for future work still remain. Real-time implementation and validation through physical experiments are crucial for verifying the practical applicability of the proposed solutions. Integration of machine learning techniques can further refine the predictive capabilities, leading to improved machining efficiency. Extending this research work to include additional machining techniques, such as additive manufacturing and hybrid machining, can broaden the scope of optimization. Furthermore, development of adaptive algorithms that dynamically adjust the machining parameters based on real-time sensor feedback can improve process control and efficiency. By using emerging optimization techniques and hybridizing metaheuristic algorithms with machine learning models, future studies can further improve their effectiveness.

This research work makes a significant contribution to modern manufacturing technologies by providing a robust framework for optimizing diverse machining processes using metaheuristic algorithms. The findings should establish a solid foundation for further exploration of these algorithms in diverse industrial applications, ultimately promoting sustainable and efficient machining practices. They would offer valuable insights for the researchers, engineers and industry professionals in selecting the most suitable optimization techniques for improving machining efficiency, sustainability and cost-effectiveness, ultimately promoting more advanced and sustainable manufacturing strategies.

Dedicated to my family and teachers ….
My pillar of strength and source of inspiration for their faith in education, endless support, love and encouragement

TABLE OF CONTENTS

Title		Page No.
TITLE SHEET		i
LIST OF PUBLICATIONS		iii
CERTIFICATE FROM THE SUPERVISOR		iv
STATEMENT OF ORIGINALITY		v
ACKNOWLEDGEMENT		vi
VITA		viii
PREFACE		ix
TABLE OF CONTENTS		xvi
ABBREVIATIONS		xix
1	**INTRODUCTION**	1
	1.1 An introduction to machining processes	1
	1.2 A brief overview of the optimization methods	8
	1.3 Need for optimization of the machining processes	12
	1.4 Objectives and scope of the present research work	14
2	**LITERATURE REVIEW**	17
	2.1 Optimization of the turning processes	17
	2.2 Optimization of the green machining processes	22
	2.3 Optimization of the laser beam machining processes	25
	2.4 Optimization of the abrasive water jet machining processes	29
	2.5 Optimization of the plasma arc machining processes	34
3	**AN OVERVIEW OF METAHEURISTIC ALGORITHMS**	37
	3.1 Human-inspired metaheuristics	37
	3.1.1 Teaching learning-based optimization algorithm	38
	3.1.2 Search and rescue optimization	39
	3.1.3 Teamwork optimization algorithm	42
	3.1.4 Human conception optimizer	43
	3.1.5 Queuing search algorithm	46
	3.2 Mating behavior-based metaheuristics	48
	3.2.1 Firefly mating algorithm	49
	3.2.2 Naked mole-rat algorithm	52
	3.2.3 Butterfly mating optimization algorithm	54
	3.2.4 Barnacle mating optimization algorithm	56
	3.2.5 American zebra optimization algorithm	58

3.3 Preying behavior-based metaheuristics 60
3.3.1 Sailfish optimizer 61
3.3.2 Harris Hawks optimizer 63
3.3.3 Aquila optimizer 66
3.3.4 Bat algorithm 68
3.3.5 Grey wolf optimizer 69
3.4 Foraging behavior-based metaheuristics 71
3.4.1 Dragonfly optimization algorithm 72
3.4.2 African vultures optimization algorithm 74
3.4.3 Grasshopper optimization algorithm 77
3.4.4 Fruit-fly optimizer 79
3.4.5 Bird swarm algorithm 80
3.5 Physics-based metaheuristics 83
3.5.1 Archimedes optimization algorithm 84
3.5.2 Atom search optimization 87
3.5.3 Nuclear reaction optimization 89
3.5.4 Electromagnetic field optimization 93
3.5.5 Gravitational search algorithm 95
4 OPTIMIZATION OF A TURNING PROCESS USING HUMAN-INSPIRED METAHEURISTIC ALGORITHMS 98
4.1 Single-objective optimization of the turning process 103
4.2 Multi-objective optimization of the turning process 106
4.3 Performance analysis 109
4.4 Results and discussion 110
5 OPTIMIZING A GREEN MACHINING PROCESS BASED ON MATING BEHAVIOR-BASED METAHEURISTICS 114
5.1 Single-objective optimization of the dry milling process 116
5.2 Multi-objective optimization of the dry milling process 119
5.3 Performance analysis 122
5.4 Results and discussion 123
6 OPTIMIZATION OF A LASER BEAM MACHINING PROCESS USING PREYING BEHAVIOR-BASED METAHEURISTICS ALGORITHMS 125
6.1 Single-objective optimization of the LBD process 127
6.2 Multi-objective optimization of the LBD process 129
6.3 Performance analysis 132
6.4 Results and discussion 133

7 OPTIMIZING AN ABRASIVE WATER JET MACHINING PROCESS BASED ON FORAGING BEHAVIOR-BASED METAHEURISTICS 135
7.1 Single-objective optimization of the AWJM process 136
7.2 Multi-objective optimization of the AWJM process 139
7.3 Performance analysis 142
7.4 Results and discussion 143
8 OPTIMIZATION OF A PLASMA ARC CUTTING PROCESS EMPLOYING PHYSICS-BASED METAHEURISTIC ALGORITHMS 145
8.1 Single-objective optimization of the PAC process 146
8.2 Multi-objective optimization of the PAC process 150
8.3 Performance analysis 152
8.4 Results and discussion 154
9 OPTIMIZING AN END MILLING PROCESS USING THE BEST-PERFORMING METAHEURISTIC ALGORITHMS 156
9.1 Single-objective optimization of the end milling process 157
9.2 Multi-objective optimization of the end milling process 160
9.3 Performance analysis 163
9.4 Results and discussion 164
10 CONCLUSIONS 166
REFERENCES 169

ABBREVIATIONS

ABC	Artificial bee colony	AC	Arc current
ACO	Ant colony optimization	AFR	Abrasive flow rate
AGTO	Artificial gorilla troops optimizer	AHA	Artificial hummingbird algorithm
AI	Artificial intelligence	AIS	Artificial immune system algorithm
ALO	Ant lion optimizer	ANFIS	Adaptive neuro-fuzzy inference system
ANN	Artificial neural network	ANOVA	Analysis of Variance
AO	Aquila optimizer	AOA	Archimedes optimization algorithm
APS	Abrasive particle size	APSO	Accelerated PSO
ASO	Atom search optimization	AVOA	African vultures optimizer
AWJM	Abrasive water jet machining	AZOA	American zebra optimization algorithm
BA	Bat optimizer	BBD	Box-Behnken design
BBO	Biogeography-based optimizer	BFRP	Banana fiber-reinforced polymer
BHA	Blackhole algorithm	BMOA	Barnacle mating optimizer algorithm
BO	Bonobo optimizer	BOA	Butterfly mating optimization algorithm
BSA	Bird swarm optimizer	CA	Cutting angle
CCD	Central composite design	Cent	Hole circularity at entry
Cext	Hole circularity at exit	CF	Cutting force
CI	Chatter indicator	CIA	Cohort intelligence algorithm
CNC	Computer numerical control	CS	Cutting speed
CSA	Cuckoo search algorithm	CT	Cutting time
DD	Drill diameter	DE	Differential evolution
DFA	Desirability function approach	DFR	Dross formation rate
DLL	Delamination length	Dm	Milling depth
DOA	Dragonfly optimizer	DOC	Depth of cut
EA	Evolutionary algorithm	EC	Energy consumption
EDM	Electrical discharge machining	EFO	Electromagnetic field optimization
ET	Exposure time	FA	Fireworks algorithm
FFA	Firefly algorithm	FFD	Full factorial design
FMA	Firefly mating algorithm	FOA	Fruit fly optimizer algorithm
Fp	Pulse frequency	FPP	Focal plane position
FR	Feed rate	FRP	Fibre-reinforced plastic
GA	Genetic algorithm	GEP	Gene expression programming
GFRP	Glass fiber reinforced polymer	GJO	Golden jackal optimisation
GOA	Grasshopper optimization algorithm	GRA	Grey relational analysis
GRASP	Greedy randomized adaptive search procedure	GSA	Gravitational search algorithm
GTO	Giant trevally optimizer	GWO	Grey wolf optimizer
GZOA	Gazelle optimisation algorithm	HAZ	Heat affected zone
HC	Hole circularity	HCO	Human conception optimizer
Hd	Hole diameter	HFMOEA	Hybrid fuzzy multi-objective evolutionary algorithm
Hh	Hole depth	HHO	Harris Hawks optimizer
HSA	Harmony search algorithm	HSMO	Hybrid spider monkey optimizer
HSTLBO	Harmony search and teaching-learning-based optimization	HTS	Heat-transfer search

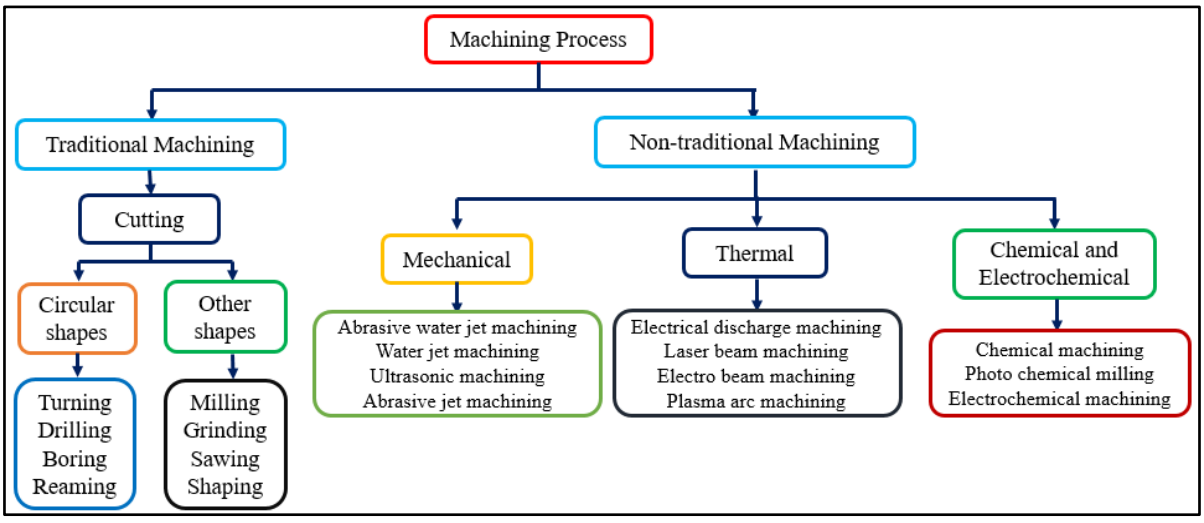
HV	Hypervolume	ILS	Iterated Local Search
ISSA	Improved salp swarm algorithm	JA	Jaya algorithm
KA	Kerf angle	KD	Kerf deviation
KHA	Krill herd algorithm	KT	Kerf taper
KW	Kerf width	LBD	Laser beam drilling
LBM	Laser beam machining	LC	Lamp current
LMBPA	Levenberg Marquardt back-propagation algorithm	Mc	Machining (cutting) cost
MFA	Mayfly algorithm	MFO	Moth flame optimizer
MH	Microhardness	ML	Machine learning
MOORA	Multi-objective optimization based on the ratio analysis	MQL	Minimum quantity lubrication
MRR	Material removal rate	MWCNT	Multi-walled carbon nanotubes
ND	Nozzle orifice diameter	NFRP	Natural-fiber-reinforced polymer
NMR	Naked mole-rat algorithm	NP	Nondeterministic polynomial
Nr	Nose radius	NRO	Nuclear reaction optimization
NS	Nozzle speed	NSGA-II	Non-dominated sorting genetic algorithm II
NTM	Non-traditional machining	OA	Orthogonal array
PAC	Plasma arc cutting	Pc	Power consumption
PF	Power factor	Pg	Gas pressure
Pl	Laser power	PO	Political optimizer
PSA	Pareto search algorithm	PSO	Particle swarm optimization
QSA	Queuing search algorithm	RLBO	Reinforcement learning-based optimization
RLT	Recast layer thickness	RSM	Response surface methodology
RSO	Rat swarm optimization	RVFL	Random vector functional link
RVM	Relevance vector machine	SA	Simulated annealing
SAR	Search and rescue optimization	SCEC	Specific cutting energy consumption
SD	Standard deviation	SFO	Sailfish optimizer
SOD	Stand-off distance	SP	Spacing
SR	Surface roughness	SS	Spindle speed
St	Traverse speed	SVM	Support vector machine
TA	Taper angle	Tc	Cutting temperature
TH	Torch height	TLBO	Teaching-learning-based optimization
TOA	Teamwork optimization algorithm	TOPSIS	Technique for order preference by similarity to ideal solution
TPLPSO	Teaching and peer-learning particle swarm optimization	TS	Tabu search
Vb	Flank wear	Vg	Arc gap voltage
VNS	Variable neighborhood search	VO	Verse optimizer
WCA	Water cycle algorithm	WEDM	Wire electrical discharge machining
WJP	Water jet pressure	Wm	Milling width
WOA	Whale optimization algorithm	Wp	Pulse width
WSMO	Weighted sum multi-objective optimization		

1. INTRODUCTION

1.1 An introduction to machining processes

With the increasing geometrical complexity and miniaturization of modern engineering components, machining operations have become more challenging, particularly in advanced industries, such as aerospace, automotive, marine, chemical processing, oil and gas, and medical sectors [1]. The demand for high precision manufacturing has led to extensive research in both conventional machining and NTM techniques. While conventional machining methods remain widely used, they often struggle to process difficult-to-cut materials, including medium and high carbon steels, superalloys and composites, due to their unique properties, such as high hardness, toughness, enhanced fatigue strength, corrosion and chemical resistance, and ability to withstand elevated temperatures. However, machining those materials using traditional methods presents significant challenges, involving excessive heat generation, increased cutting forces and rapid tool wear, ultimately leading to reduced efficiency and higher production cost [2, 3, 4].

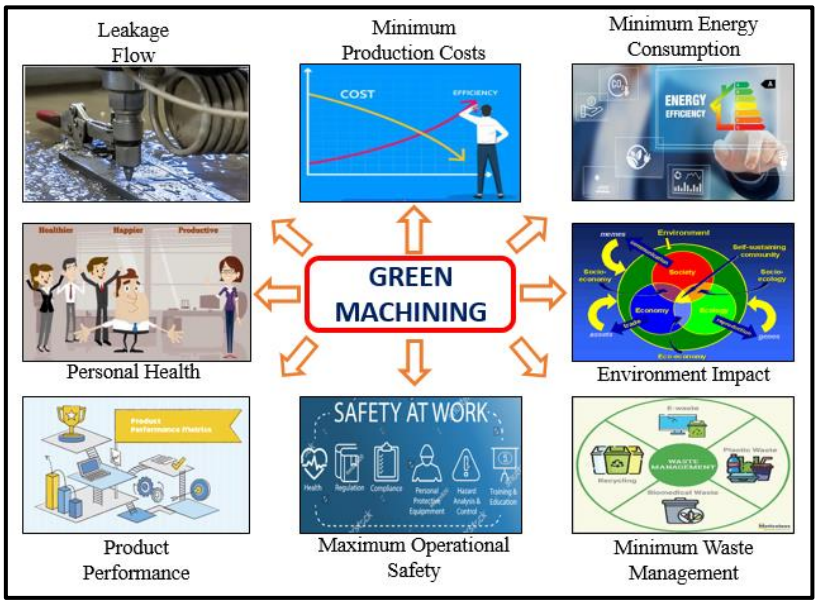
To address these limitations, several NTM techniques have emerged out as efficient alternatives. The NTM processes, such as LBM, PAC, AWJM, EDM and WEDM have proven to be highly effective for machining hard and tough materials while ensuring superior precision and surface quality [5, 6]. Unlike the conventional techniques, which rely on direct tool-workpiece contact, NTM methods employ various energy forms, such as mechanical, thermal, electrical and chemical, to facilitate material removal at the atomic level, minimizing tool wear and process inefficiencies [7]. Furthermore, hybrid machining approaches, which combine the advantages of multiple machining methods, have gained prominence. Examples include electrochemical discharge machining, which integrates electrochemical and EDM principles, and electrochemical grinding, a fusion of electrochemical and mechanical grinding processes [8, 9]. The detailed classification of the machining processes is shown in Figure 1.1.



Figures 1.1 Classification of machining processes

Traditional machining operations, like turning, shaping and drilling remain essential for material removal, achieving desired shapes, and ensuring accuracy, tolerance and surface finish [10]. However,

selection of an appropriate machining process depends on multiple factors, including workpiece geometry, operational costs, and mechanical and tribological properties of the work material. As industries strive to improve sustainability, there has been a growing shift towards eco-friendly machining solutions. This transition aligns with the philosophy of 'Go Green-Think Green-Act Green,' focusing on energy efficiency, carbon footprint reduction and sustainable manufacturing practices [11]. Figure 1.2 demonstrates the benefits of implementing green machining processes in a typical manufacturing environment [12].



Figures 1.2 Benefits of green machining processes

Since machining parameters directly influence the output responses, their optimal selection is critical for achieving a balance between efficiency, quality and environmental sustainability [13]. Conventional machining techniques significantly depend on metal-working fluids for lubrication, chip removal, friction reduction, dimensional accuracy and overall process efficiency [14]. However, these fluids contribute to nearly half of the total production cost, and release hazardous fumes, posing health risks to the operators and environmental concerns [15].

To mitigate these challenges, the green machining techniques, such as dry cutting and MQL, have gained significant attention. Additionally, eco-friendly alternatives to conventional metal-working fluids, including cryogenic cooling, high-pressure cooling and biodegradable lubricants, offer promising solutions to reduce environmental impact while maintaining higher machining efficiency. The integration of real-time feedback systems, sensor technologies and adaptive control mechanisms further enhances machining performance, making it possible to optimize the process parameters dynamically and improve overall efficiency [16]. The working principles of some of the machining processes related to this research work are discussed here-in-under:

Turning process

Turning is a fundamental operation widely utilized in most of the manufacturing industries for removal of excess material from the exterior surface of a cylindrical workpiece. It involves rotating the

workpiece while a cutting tool moves radially, axially or both, to achieve the desired shape geometry. This process can be executed either in a single cut or multiple cuts on a lathe, which supplies the necessary power for rotation of the workpiece at a specific speed while feeding the cutting tool at a defined rate and DOC. This process has become extremely popular in automotive, aerospace and other industries mainly due to its versatility and cost-effectiveness. It has been experienced that for any of the machining processes, choice of the appropriate input parameters is extremely crucial for achieving high quality products at low cost [17]. Like other machining processes, in the turning operation, several input parameters, like SS or CS, DOC, FR, CA, Nr, type of the work and tool materials, type of the cutting environment etc. need to be carefully considered as they have significant contributions in attaining higher MRR, and lower CF, Pc, SR, CT, SCEC, Vb etc. [18]. Poor parameter selection may lead to undesirable effects, such as temperature rise, tool breakage and work material fatigue [19, 20]. Figure 1.3 illustrates the design of a typical turning process.

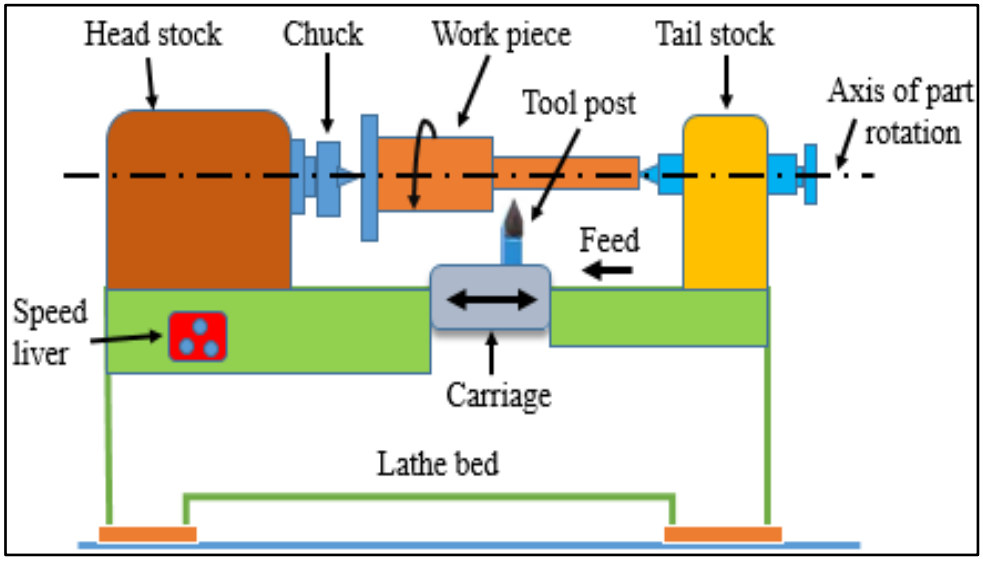


Figure 1.3 Working principle of a turning operation

Milling process

Milling is a versatile and widely used machining technique that removes material using a rotating multi-point cutting tool. It is widely used in modern manufacturing industries, particularly with advancements in CNC technology, which contributes to automation, high productivity and superior-quality production. The milling operations can be categorized based on the cutting tool's movement and material removal strategy, such as face milling, end milling, form milling etc. Figure 1.4 demonstrates the schematic of a typical milling process.

End milling is a critical machining operation in which a rotating cutter with cutting edges on its periphery and end removes material from the workpiece. It is commonly used for machining slots, keyways, pockets and contours. The process allows for high precision and generation of complex geometries. As the cutting action occurs in both radial and axial directions, it has become an ideal processes for achieving detailed and smooth finishes on different components.

Milling can be performed under various conditions, such as dry milling, MQL and flood cooling, each influencing tool performance, surface quality and environmental impact. The dry milling, in particular, operates with minimal or no lubrication, helping to reduce fluid-related cost and mitigate environmental concerns. Numerous researchers have investigated the effects of different dry milling parameters, such as SS, FR, tool geometry and cutting environment on the achievable machining performance with respect to MRR, SR, tool wear, CF, Pc etc. [21]. Among these factors, surface quality is often considered one of the most critical quality characteristics, as it directly influences key properties of the machined components, including wear resistance, fatigue strength and corrosion resistance [20]. Similar to turning, optimizing the milling parameters is also essential. Improper selection may lead to reduced machining performance, excessive tool wear, temperature rise and material fatigue, ultimately compromising both productivity and product quality [21].

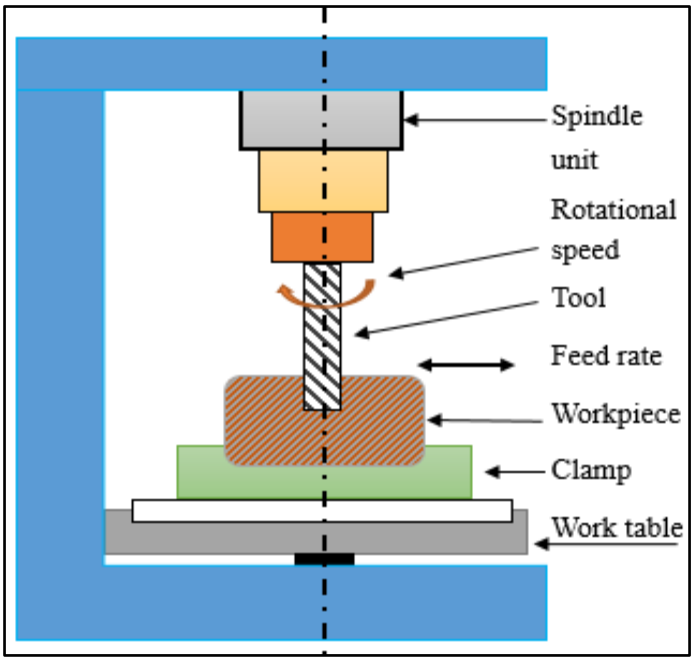


Figure 1.4 Working principle of a milling process

LBD process

Increasing demand for miniaturized components as micro-holes with diameters ranging from a few microns to hundreds of microns used in automobile, aviation, electronics and medical industries makes LBD a viable option in today's manufacturing environment. LBD has emerged out as a viable option to generate micro-holes and complicated shape geometries in almost all types of difficult-to-cut materials, including metals, non-metals, composites and ceramics. In this process, a narrow laser beam having high intensity and very small pulse duration is focused to ablate material from the workpiece surface generating tiny holes with higher aspect ratio (>5:1), minimum HAZ and almost zero taper geometry. Thus, the material removal mechanism of the LBD process is absorption of laser energy through a series of pulses at the same spot causing localized melting and vaporization. During the LBD operation, a highly pressurized assist gas, i.e. air, oxygen or nitrogen is utilized to blow away the debris

from the machining zone. Although it has several advantageous features, like no tool wear, higher precision and flexibility, excellent surface finish, higher repeatability, no residual stress and burr formation, no chatter and vibration, but it still suffers from other disadvantages, like low MRR, higher initial cost, requirement of skilled manpower etc. [22, 23]. Micro-holes generated by this process have found immense applications in electronics (panel display boards, printed circuit boards, micro-electro-mechanical systems, solar cells etc.), medical industry for generation of micro-channels, packaging, manufacturing of ignition target balls, micro-heat exchangers, cooling holes in aero-engine turbine blades etc.

Among various laser sources, pulsed Nd:YAG laser and CO_2 laser, which are solid and gas lasers respectively, have become quite popular for the above-mentioned industrial applications. These laser sources facilitate micro-drilling of thick materials due to higher incident peak power during the pulsating mode. It has been experimented that Nd:YAG laser performs better than CO_2 laser with respect to higher production rate, and generation of geometrical features having excellent dimensional accuracy and tolerance due to its higher energy density and lower average beam power. Usually, the quality characteristics of micro-holes during the LBD process are expressed with respect to SR, TA, HC, Hd at both entry and exit, HAZ, RLT, MRR etc. which are often influenced by different input parameters of the LBD process, like Pl, LC, Fp, Wp, type and pressure of the assist gas etc. [24]. Various researchers have experimentally studied the effects of the LBD parameters on the quality characteristics of micro-holes, and have recommended to operate the LBD process while setting its different input parameters at their optimal levels. Complex material removal mechanism, stochastic behavior of the process, and nonlinear relationship between the input and output parameters make optimization of an LBD process more challenging and complicated. Figure 1.5 depicts the working principle of a typical LBD process.

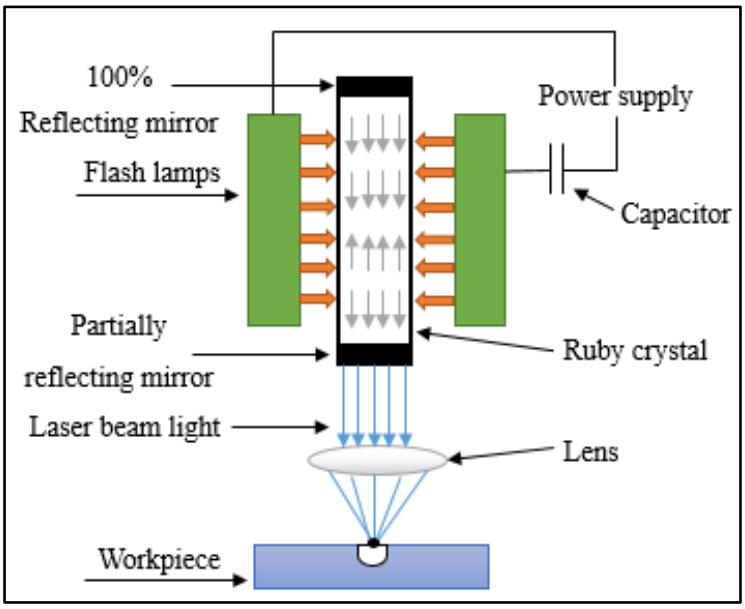


Figure 1.5 Working principle of an LBD process

AWJM process

Due to hybridization of the material removal mechanisms of abrasive jet and water jet machining processes, AWJM appears as an efficient NTM process providing higher productivity, superior surface quality and excellent dimensional accuracy for almost all types of work materials irrespective of their mechanical and thermal properties. AWJM process was developed during early 1980s as an innovative cutting technique, meeting the requirements of machining almost all types of conductive and non-conductive materials, like medium and high carbon steels, aluminium, titanium and their alloys, ceramics, composites, marbles, rocks etc. In this process, different abrasives, like aluminium oxide, boron carbide, silicon carbide, garnet etc. are proportionally mixed with water so that they along with water come out from a nozzle at extremely high speed to remove material from the workpiece due to plastic deformation. Thus, material removal is facilitated by converting the potential energy into high velocity kinetic energy. The working principle of an AWJM process is schematically shown in Figure 1.6. This process is specially suitable for machining thicker materials (thickness varies between 0.8 and 25 mm or more) while maintaining the preferred level of dimensional accuracy and surface quality [25, 26]. It also has other advantages, like higher flexibility, no thermal stress, higher CS, no HAZ and burr formation, and environmental friendliness [27, 28]. Higher initial cost, requirement of skilled manpower, low MRR and corrosive machining environment are its major demerits.

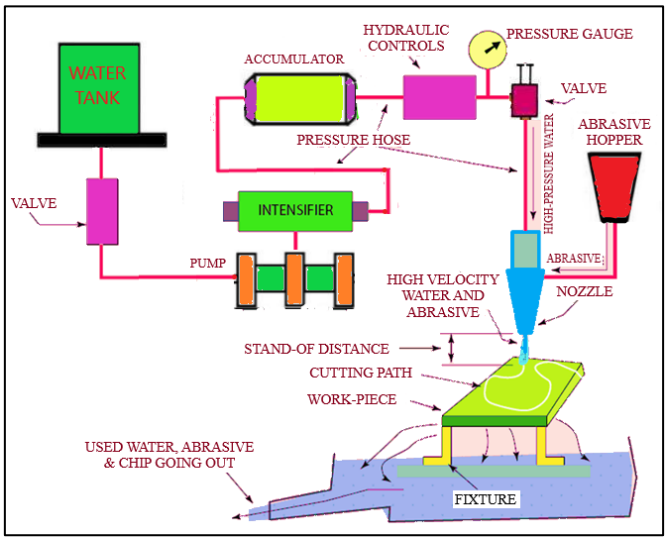


Figure 1.6 A typical AWJM process

PAC process

PAC, introduced in the 1950s, has become an advanced technique to address limitations of the traditional machining processes, and has evolved as a highly efficient process for cutting complex profiles in most of the hard-to-machine materials due to its numerous advantages, like superior dimensional accuracy, higher productivity, ability to cut thicker materials etc., while achieving better cut quality, particularly with respect to reduced HAZ, improved surface quality and controlled kerf characteristics, which remain a challenging task due to numerous process variables involved [29, 30].

The working principle of a PAC process involves an arc formed between the electrode and the workpiece, constricted by a fine-bore copper nozzle, which significantly increases temperature and velocity of the plasma emanating from the nozzle [31]. It employs nitrogen, argon and compressed air to generate a plasma stream with a water-cooled nozzle to effectively manage the intense heat while cutting complex profiles on various advanced conductive materials, like aluminium, titanium, magnesium, copper, stainless steel etc. [32]. This plasma can reach temperatures exceeding 20,000 °C and velocities approaching the speed of sound. During cutting, the plasma gas flow is intensified, allowing the deeply penetrating plasma jet to cut through the material, with molten material being expelled by the efflux plasma. This process significantly reduces tool wear, and efficiently cuts intricate shapes without direct contact between the tool and the workpiece [33, 34]. The schematic of a typical PAC process is depicted in Figure 1.7. During the cutting operation, PAC usually involves in generation of smoke, noise and evaporated molten metal, which pollute the environment, especially when cutting non-ferrous metals with high current. Therefore, to reduce its detrimental effects on the environment and operators' health, it becomes imperative to operate it with an ideal combination of the input parameters [29, 35].

The PAC has been the prime focus of numerous studies aimed at optimizing its various process parameters, such as SOD, Pg, CS, AC, TH etc. to improve its corresponding responses, like SR, KW, HAZ, KT, MRR, MH etc. Despite considerable advancements, the involvement of a few parameters, like FR and edge roughness has not been investigated thoroughly [33, 36]. It has been observed that these PAC parameters would significantly influence attainment of the desired values of most of the responses. For example, lower CS and SOD may lead to increased MRR while a rise in Pg may decrease the value of MRR with a linearly decreasing SR. On the other hand, higher AC may result in higher MRR along with lower SR, while higher CS may be responsible for an enlarged kerf surface quality, reducing dimension of HAZ of the machined components. A minimized MH is observed at lower values of AC and SOD [35, 37, 38].

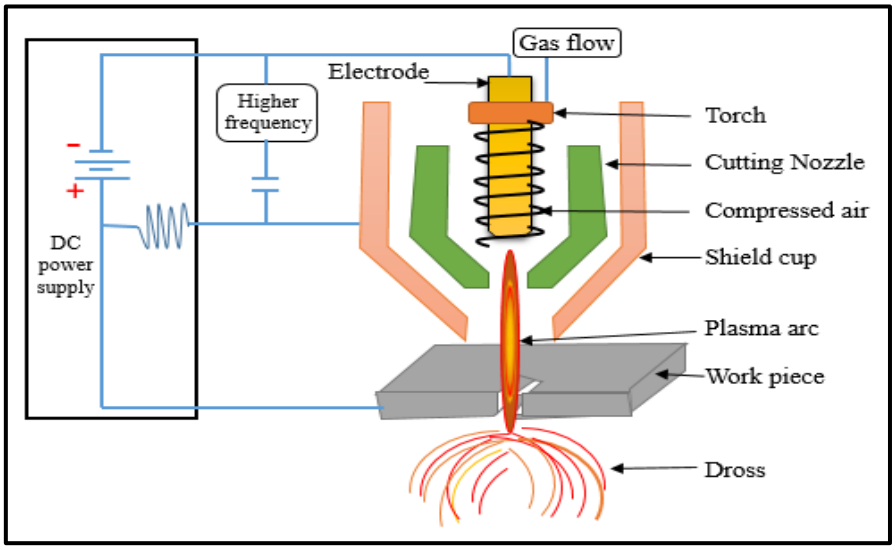


Figure 1.7 Working principle of a PAC process

1.2 A brief overview of the optimization methods

Optimization is the process of finding out the best possible solution to a problem by maximizing or minimizing an objective function while satisfying the given constraints. It plays a crucial role in various fields, including engineering, manufacturing, supply chain management, finance, AI and operations research. Selection of an appropriate optimization method depends on the problem complexity, computational resources and desired objectives. Numerous techniques have been developed, each with distinct behaviors suited for specific problems. Optimization techniques are broadly classified into deterministic (traditional) and stochastic methods. Deterministic methods, such as Newton's method, linear programming and gradient-based approaches, guarantee optimal solutions but require well-defined mathematical models. These methods often provide only one solution per iteration and can be computationally intensive [39, 40]. In contrast, stochastic methods overcome these limitations by offering better solutions for complex multimodal problems. Metaheuristic algorithms, such as GA, PSO and SA rely on probabilistic strategies to efficiently explore vast search spaces. These techniques are particularly useful for solving complex, nonlinear and multi-objective optimization problems where traditional methods fail.

Modern advancements have led to hybrid optimization techniques that combine multiple approaches to enhance accuracy, convergence speed and computational efficiency. Real-world applications of optimization range from supply chain management and financial portfolio optimization to ML hyperparameter tuning and resource allocation in large-scale systems. Despite their advantages, challenges, such as computational cost, local optima trapping and parameter tuning require careful consideration for their successful implementation [41-43].

Many optimization problems are classified as NP-hard, meaning they cannot be solved in polynomial time unless NP = P. To address this, researchers have developed various workarounds and approximations, categorized as heuristics and metaheuristics, to find acceptable solutions with minimal time complexity. Heuristics are more problem-specific, whereas, metaheuristics provide a broader algorithmic framework that can be applied to almost any optimization problem [44, 45].

Environmental decision making problems often require selecting the most suitable option among multiple alternatives. Environmental models play a key role in this process by assessing the utility of different decision alternatives, such as infrastructure interventions, management strategies and policy options. However, due to complexity and large number of alternatives, identifying the best option can be challenging. By integrating simulation models with modern metaheuristic optimization techniques, the decision makers can identify solutions that maximize resource efficiency, achieve the best environmental outcomes within a given budget, or optimize trade-offs between various competing objectives. This enables more effective decision making, ensuring that the selected options would provide the best value while minimizing the negative environmental impacts [42]. Overall, optimization remains a critical tool across various domains, driving efficiency, accuracy and better decision making. As research continues to evolve, improvements in hybrid methods, computational efficiency and

adaptability to real-world constraints would further expand the scope and applicability of the optimization techniques.

Metaheuristic algorithms play a crucial role in optimizing machining and manufacturing processes by determining the optimal combination of different parameters, such as CS, FR, SOD, Pg, Pl, St, DOC etc. to achieve several objectives, like minimal tool wear rate, SCEC, HAZ and Pc, maximum MRR and MH, and improved surface quality. These algorithms effectively balance exploration (diversification) and exploitation (intensification) to navigate complex optimization problems, avoiding local optima and improving solution quality. Advancements in evolutionary computation have further enhanced their applicability in multi-objective and real-time decision making scenarios, although challenges, such as computational cost, parameter tuning and convergence speed remain active areas of research [46-49].

Exploration involves a global search across the solution space to identify promising regions, while exploitation refines solutions within these regions to find out better alternatives. Metaheuristics are widely used due to their simplicity, ease of implementation and adaptability across diverse problem domains without requiring modifications to their algorithmic structure [50]. A metaheuristic begins with a randomly generated solution and iteratively refines it to move closer to an optimal outcome. Since a single algorithm can be employed for multiple agents, and can represent various behaviors through parameterization and neighboring strategies, their implementation is relatively straightforward [51, 52].

The term ‘metaheuristic,’ introduced by Glover in 1986, originates from the Greek words ‘meta’ (beyond) and ‘heuristic’ (to discover). These high-level, problem-independent strategies utilize heuristics to efficiently find out near-optimal solutions to complex problems, particularly NP-hard problems, where exact algorithms demand impractical computation times or resources. These algorithms have been successfully applied to solving intractable optimization problems, offering efficient solutions even for very large problem sizes within a reasonable timeframe [41]. Unlike traditional optimization methods that rely on gradient-based approaches, metaheuristics do not require derivative information, making them applicable to a wide range of problems with minimal modifications. Beyond optimization, they are also used in environmental management, where they help identify decision alternatives that balance trade-offs between multiple desired outcomes. Their adaptability and efficiency make them indispensable for solving complex problems across various domains [46, 48].

Metaheuristics have been widely applied in machining processes optimization, where the input-output relationships are nonlinear. Many of these algorithms are ‘nature-inspired’ (also called ‘metaphor-based’), as they involve heuristics that imitate processes in natural systems. Their ability to handle real-time optimization challenges, deliver high quality solutions within shorter computation times, and function as approximate and non-deterministic methods, distinguish them from precise optimization approaches in terms of reliability and efficiency [53, 54]. The key benefits of metaheuristic algorithms include ease of implementation, conceptual simplicity, independence of problem type and

derivative information, reduced time complexity, no gradient mechanism, faster convergence speed and accuracy, and effectiveness in solving nonlinear, non-convex, NP-hard high-dimensional problems [54, 55].

A metaheuristic starts with a random solution and works to refine it iteratively until it approaches the optimal solution. It is easy to develop a metaheuristic since a single algorithm may be used for all the agents, and can represent different behaviors through nearby strategies and parameterization [56, 57]. Figure 1.8 presents a taxonomy of the metaheuristic algorithms, which are high-level strategies used to find approximate solutions to complex optimization problems. These metaheuristics are broadly categorized into two groups, i.e. nature-inspired and non-nature-inspired approaches.

i) Nature-inspired metaheuristics: These algorithms are inspired by natural processes, such as biological evolution, animal behavior, physical phenomena or human decision making strategies. They are further divided into five subcategories.
    a) Evolutionary-based: Simulate the principles of natural selection and evolution, where the solutions evolve over successive generations through selection, crossover and mutation. Examples of these algorithm include GA, DE, FOA, BOA etc.
    b) Swarm-based algorithms: Inspired by the collective behavior of social animals, such as birds, ants and bees, these algorithms use decentralized and cooperative approaches to explore the solution space. PSO, ACO, ABC, GWO, ALO etc. are the typical examples of swarm-based metaheuristic algorithms.
    c) Physics-based approaches: Simulate physical principles, such as motion, forces and interactions between particles. They include AOA, ASO, EFO, GSA, NRO etc.
    d) Human-inspired algorithms: Inspired by human decision making and learning, and cognitive processes. TLBO, QSA, HCO, TOA, SAR etc. come under this category.
    e) Others: Include algorithms inspired by various natural processes, like foraging, mating and preying behaviors, such as HHO, DOA, AO, BSA, SFO etc.

ii) Non-nature-inspired metaheuristics: Unlike nature-inspired approaches, these algorithms rely on mathematical principles, probability and AI rather than natural phenomena. They are further divided into three subcategories.
    a) Local search-based approaches: Focus on exploring neighboring solutions to refine the optimization results. These metaheuristics include TS, ILS, HSA etc.
    b) Mathematical and probabilistic algorithms: Employ probabilistic, statistical and mathematical formulations to guide the search process. Examples of these algorithm include SA, AIS, GRASP, VNS etc.
    c) Hybrid and ML-based approaches: Integrate multiple optimization techniques with ML approaches to enhance adaptability and improve solution quality, including RLBO and

hyper-heuristics, which dynamically select or generate heuristics based on the problem characteristics.

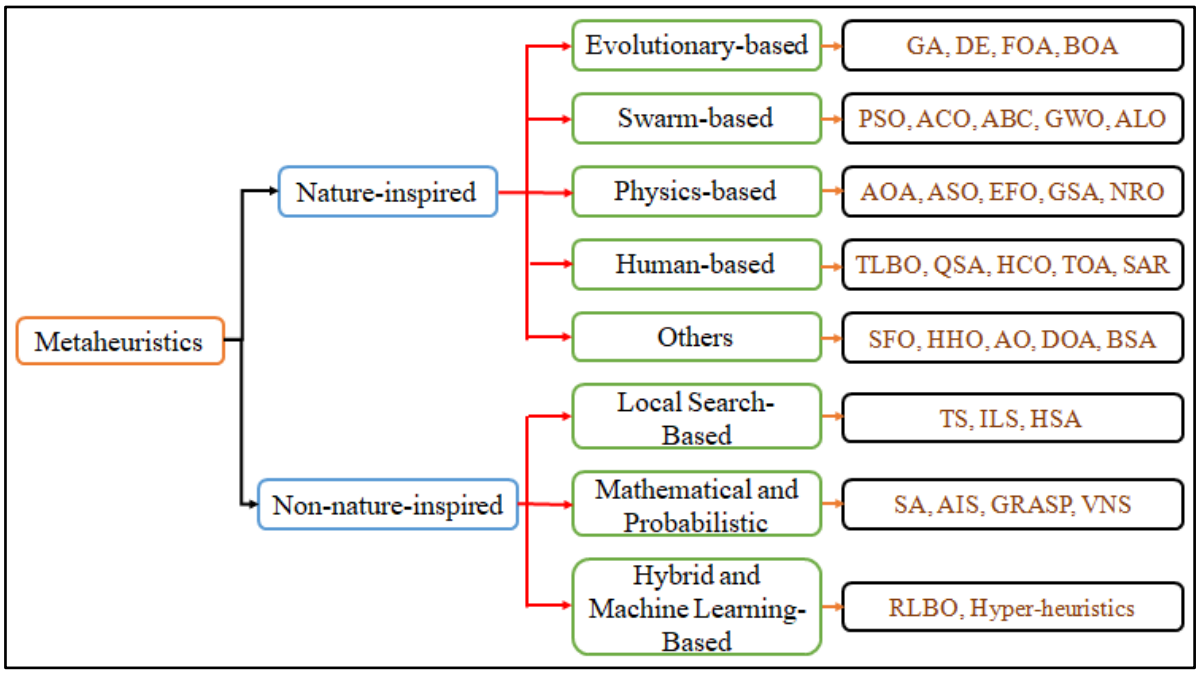


Figure 1.8 Classification of the metaheuristic algorithms

Figure 1.8 is structured hierarchically, illustrating relationships between metaheuristic categories and specific algorithms. Broadly, nature-inspired metaheuristics can be categorized into population-based and single solution-based algorithms, with EAs being widely used due to their optimization efficiency [42, 46]. Classical metaheuristic algorithms developed before 2000 include GA, PSO, ACO, genetic programming, DE, SA, TS, GRASP, AIS, ILS etc. While these methods have achieved significant success, the past two decades have introduced new evolutionary approaches inspired by biological and behavioral processes, improving solutions for many of the previously unsolved benchmark problems [41, 49].

Beyond engineering, metaheuristics have applications in diverse fields, such as finance, logistics, healthcare and computer science. In finance, they assist in portfolio optimization, risk management and algorithmic trading. In logistics, they improve vehicle routing, inventory management and facility location planning. In healthcare, they optimize patient scheduling, resource allocation and treatment planning. In computer science, metaheuristics contribute to ML, computer vision and natural language processing [46, 47].

Despite their advantages, metaheuristics have certain limitations. Due to their stochastic nature, they do not guarantee finding out the global optimum. They can also be computationally expensive, specially for large-scale problems, and often require careful parameter tuning for their optimal performance. However, as computational power continues to advance, hybrid and adaptive metaheuristic approaches are expected to further enhance their efficiency and applicability. A key challenge in designing efficient metaheuristics is maintaining a proper balance between exploration and

exploitation. Various classification schemes have been proposed based on how these two processes are implemented and the metaphors underlying different search strategies [47]. A metaheuristic is thus an algorithmic framework that can be applied to a wide range of optimization problems with minimal modifications to suit a specific problem. The abstract of the working mechanism of a metaheuristic is shown as below [47].

```
Algorithmic framework of a metaheuristic
Create one or more initial solutions
    While (stopping criterion not satisfied) do
        If exploit then
            Create new solution by exploitation step;
        Else
            Create new solutions by exploration step;
        End
        Update best-found solution;
    End
    Return best-found solution;
```

The key characteristics of the metaheuristics include the following:

a) They are not designed for a specific problem but can be adapted to different optimization challenges.

b) Provide approximate solutions rather than exact ones.

c) Explore the search space to identify solutions that are 'good enough.'

d) Operate at an abstraction level that allows generalization across different domains.

e) They are often easily parallelizable, enhancing computational efficiency.

f) Range from basic local search techniques to advanced learning-based methods.

g) Incorporate strategies to avoid premature convergence, improving solution diversity.

h) Leverage heuristics as domain-specific knowledge within an overarching search strategy.

i) Modern metaheuristics utilize guidance memory to retain and apply search experience, enhancing performance.

By combining flexibility, adaptability and efficiency, metaheuristics therefore continue to be instrumental in solving complex optimization problems across diverse applications [47].

1.3 Need for optimization of the machining processes

Optimization plays a crucial role in achieving superior cut quality, minimizing material waste and enhancing overall productivity in the machining processes. Several key process parameters, including CS, FR, DOC, SOD, SS, Pg, ND etc. significantly influence the machining performance. These parameters, commonly used across various machining processes, must be carefully controlled to achieve the optimal results. Selection of the appropriate parameter settings at different operating levels directly impacts the key machining responses, such as MRR, SR, SCEC etc. Table 1.1 presents a comprehensive list of various process parameters and corresponding responses for the machining processes considered in this research work. By optimizing these parameters, the manufacturers can

enhance surface finish, maintain dimensional accuracy and reduce defects, such as dross formation. Proper parameter selection also minimizes tool deflection and vibration, improving process stability and reliability. Additionally, effective optimization controls heat generation during machining, preventing thermal damage to the workpiece and extending tool life. As a result, precision, efficiency and cost-effectiveness are significantly improved in modern manufacturing operations [58, 59].

Table 1.1 Machining parameters and responses

Machining process	Process parameters	Responses
Turning	CS, FR, DOC, CA, SS, Nr, lubrication condition	SCEC, SR, MRR, Mc, Vb, CF, Pc
Milling	CS, FR, DOC, SS, Wm, Nr	SR, PF, EC, MRR, SCEC, Vb, Pc
LBM	Pl, ET, FPP, Pg, AC, CS, Fp	MRR, SR, HAZ, TA, Cent, Cext, Hd, Hh, KT, KW
AWJM	WJP, SOD, St, AFR, APS, ND	MRR, KA, SR, DLL, KW, Hd, HC
PAC	SOD, Pg, CS, AC, TH, Vg	MRR, SR, KT, HAZ, KW, MH, DFR

Machining process optimization enhances productivity by reducing cycle times, refining tool paths and improving throughput. It also lowers operational costs by minimizing tool wear, reducing energy consumption, decreasing material waste and lowering the environmental impact. Furthermore, by optimizing machining parameters, the manufacturers can ensure consistent quality in mass production, leading to reduced rework and material wastage. Additionally, optimizing machining conditions extends tool life, reducing frequency of tool replacements and maintenance costs. Environmental-friendly machining strategies, such as MQL and dry machining, can be integrated into optimization techniques to further reduce ecological footprints [60, 61].

The complexity of machining processes arises from multiple conflicting objectives, such as minimizing production time and cost while maximizing surface finish and tool life. These processes involve highly constrained, nonlinear relationships between the variables, often with mixed integer-discrete-continuous parameters. Traditional optimization methods, such as linear and nonlinear programming, struggle with these complexities. The stochastic nature of material removal mechanisms further complicates the optimization process, requiring advanced mathematical tools. Additionally, variations in material properties, machine tool stiffness and cutting tool geometry introduce uncertainties making process optimization even more challenging.

To address these challenges, metaheuristic algorithms are widely used. These techniques efficiently explore search spaces, avoid local optima and determine the optimal machining parameters without extensive trial-and-error experimentation. Hybrid optimization approaches, such as integrating ANN with EA, are gaining popularity for their ability to model complex relationships and improve optimization accuracy. The ML-based models are also being utilized to predict the optimal machining

conditions based on historical data, further enhancing process efficiency. Such advanced approaches help manufacturers enhance efficiency, reduce operational costs and improve product quality, ensuring competitiveness in the modern manufacturing industries [46].

Beyond machining, production process optimization focuses on improving the entire manufacturing workflow by analyzing the flow rates, machine layout and labor utilization. By balancing operating workloads and machines at their optimal speeds and eliminating bottlenecks, the manufacturers can minimize downtime, enhance machine utilization and improve operational efficiency. Addressing root causes of unplanned downtime, such as machine breakdowns, operator errors and quality issues, further ensures productivity and reliability. Predictive maintenance, enabled by the internet of things and data analytics, plays a key role in modern production optimization, helping manufacturers detect equipment failures before they occur and avoid costly downtimes [62, 63].

Optimizing production processes leads to several advantages, including reduced variation in product manufacturing, improved accuracy to specifications, increased reliability and durability, and fewer defects. Automation and real-time monitoring systems contribute significantly to optimization by reducing human errors, improving process control and enabling quick adjustments when deviations occur. Additionally, process optimization enhances operational visibility by streamlining workflows, automating tasks and improving cross-department communication. This empowers the managers to track progress, identify bottlenecks and make data-driven decisions [64].

By leveraging advanced optimization techniques, the manufacturers can improve efficiency, reduce costs and enhance product quality, achieving a sustainable and competitive edge in the industry. Future advancements in AI, digital twin technology and cyber-physical system would further revolutionize process optimization, making manufacturing more adaptive, intelligent and efficient. As industries continue to move towards smart manufacturing and Industry 4.0, optimization would remain a key driver of innovation and success.

1.4 Objectives and scope of the present research work

Machining processes, such as turning, milling, LBM, AWJM, PAC etc. play crucial roles in modern manufacturing. Efficiency, accuracy and surface quality of the machined components in these processes are largely governed by various input parameters, such as CS, FR, DOC, Pg, tool geometry, machining environment etc. Improper selection of those parameters can result in undesirable outcomes, including increased tool wear, poor surface finish and inefficient material removal.

To overcome these challenges and achieve optimal machining performance, metaheuristic algorithms have been extensively employed for parametric optimization. These algorithms are highly effective in dealing with complex, multi-objective optimization problems where traditional optimization techniques often prove inadequate. Considering these requirements, the objectives of the present research work are outlined as follows:

a) to compare the performance of various metaheuristic algorithms in predicting the optimal results for single and multi-objective optimization problems in both traditional machining and NTM processes,
b) to assess consistency and robustness of the adopted metaheuristic algorithms in deriving the optimal parametric solutions for different machining operations,
c) to implement five human-inspired metaheuristic algorithms for optimizing the process parameters in a turning operation,
d) to integrate five mating behavior-based algorithms for optimizing the process parameters of a green machining operation, to promote sustainability and energy efficiency,
e) to employ five preying behavior-based algorithms for identifying the most suitable parameter settings in a CO_2-based micro-LBD process,
f) to apply five foraging behavior-based algorithms for optimizing the cutting conditions in an AWJM process,
g) to integrate five physics-based metaheuristic algorithms for parametric optimization of a PAC process,
h) in order to identify the most robust approach for future machining applications, to validate the optimization performance of the best five algorithms from five distinct categories of metaheuristics in an end milling operation, and
i) to strengthen the optimization results by conducting the performance analysis using two quality metrics (SP and HV), and applying two statistical non-parametric tests (Friedman's mean rank test and Wilcoxon rank sum test). Additionally, the corresponding Pareto optimal fronts are also developed to derive the unique parametric settings and optimal multi-objective solutions for the considered machining processes.

The application of metaheuristic algorithms in machining process optimization enhances machining performance while also contributing to sustainable and energy-efficient manufacturing. By reducing tool wear, optimizing material usage and minimizing waste, these algorithms support environmental-friendly manufacturing practices. Future research may explore integrating them with intelligent manufacturing systems, enabling automated process optimization and real-time decision making in industrial machining environments.

In most of the machining processes, equal weights are typically assigned to the considered responses to simplify the computational steps. However, the effects of varying these weights on the optimal parametric settings are analyzed using Pareto optimal fronts. While five distinct categories of the nature-inspired metaheuristics are used and their relative performance are compared, future studies may concentrate on comparing their performance against other techniques, like energy valley optimizer, flow direction algorithm, equilibrium optimizer, multi-verse optimizer, predator-prey optimizer, bacterial foraging optimization algorithm, elephant swarm water search algorithm, Manta Ray foraging algorithm, battle royale optimization, war strategy optimization, coronavirus herd immunity

optimization etc. Effects of varying values of different algorithm-specific parameters and weights assigned to the responses on the optimization performance of the considered algorithms may also be explored.

It may be worthwhile to evaluate their effectiveness in addressing many-objective optimization problems. In future research work, more emphasis may be put on exploring the applicability of various ML algorithms in modeling the relationships between different machining parameters and cut quality characteristics, eventually leading to higher precision and process efficiency. Application of these algorithms may further be exploited by incorporating parameter tuning techniques, hybridizing with other metaheuristics and leveraging advanced optimization tools.

2. LITERATURE REVIEW

Machining process optimization is essential for enhancing productivity, improving surface quality, reducing tool wear and minimizing production costs. Various machining processes, including turning, milling, LBM, AWJM and PAC, have been optimized using traditional and advanced computational techniques. This literature review presents key research contributions in these domains.

2.1 Optimization of the turning processes

The previous researchers have already employed several techniques, including RSM and Taguchi methodology to optimize and fine-tune the operational settings of different turning processes. However, there has been limited research on metaheuristic-based optimization of turning processes to reduce energy consumption, while maintaining the environmental protection laws. It is always essential to set the input parameters of any turning process at their optimal levels due to their association with the responses and complex cutting behavior. To resolve the parametric optimization challenges, the researchers have turned to various mathematical tools, with metaheuristic algorithms playing a key role. Table 2.1 summarizes their applications, together with more details, such as experimental plan, workpiece material, turning parameters and response(s). Notably, DE [65], MFA [66, 67], PSO [67-73], CIA [70], PO [71], GA [73-83], PSA [83], GWO [83-85], VO [85], ALO [85], SA [73, 85, 86], WOA [87], AHA [88], NSGA-II [89-94], TLBO [95], SAR [95], TOA [95], HCO [95] and QSA [95] have become extremely popular among the researchers. Additionally, most of the turning operations, specially those involving difficult-to-machine materials, have utilized different experimental design plans, like Taguchi's OA, BBD, FFD, CCD etc. However, only few research works had compared the performance of those algorithms relating to computational effort, speed of convergence and solution quality. Although many algorithms have effectively been applied for optimization of the turning processes, little effort has been put forward to identify the most effective one among those developed based on a similar principle.

The literature review underscores the critical role of optimizing various turning parameters to improve machining performance with minimum energy requirement. Abderazek et al. [65] optimized the turning of EN-GJL-250 cast iron using DE, considering CS, FR and DOC as the input parameters, and Pc, SR and CF as the responses. The study highlighted DE's effectiveness in multi-objective optimization of the said process. Arunkumar and Devendiran [66] investigated and simultaneously optimized MQL-turning parameters for sustainable machining of Nimonic 75 using weighted MFA, where the weights had been determined through grey relational coefficient. Compared to the experimental results, the optimized computation of the corresponding grey relational coefficients had led to significant reduction in the output responses, demonstrating effectiveness of the grey relational coefficient-MFA approach. In a subsequent study, Arunkumar and Devendiran [67] validated that MFA would be more efficacious than PSO in optimizing the turning operation on Nimonic 80A under dry and nano-MQL conditions, and picked out the related process parameter settings as FR = 0.15 mm/rev, CS = 76 m/min and DOC = 0.75 mm, resulting in achievement of minimum values of SR, Vb, CF and

cutting acceleration. Doan et al. [68] applied an improved quantum-behaved PSO algorithm in combination with two multi-criteria decision making method tools, i.e. TOPSIS and method based on the removal effects of criteria (MEREC), for optimization of turning operation on titanium alloy. The study treated CS, FR, DOC and CA as the input variables, and SR, comprehensive energy and turning noise as the outputs. In another study, Doan et al. [69] employed entropy method, and vibration and communication PSO algorithm to compute the weights, and determined the optimal factor levels, focusing on SR and SCEC as the key responses. The optimization results achieved significant improvements in both machining quality and energy efficiency.

Shastri et al. [70] adopted PSO and CIA techniques to optimize the machining of titanium alloy, treating CS, FR and CA as the three key parameters. The study validated the effectiveness of multi-CIA, demonstrating superior improvement in the machining responses compared to PSO, leading to enhanced machining quality. Elsheikh et al. [71] developed an improved RVFL model optimized using PO algorithm to predict the turning responses. The RVFL-PO model outperformed the standalone RVFL and RVFL-PSO, achieving the highest accuracy, with maximum coefficient of determination recorded as 0.961, compared to 0.844 for RVFL-PSO and 0.768 for RVFL. Xie et al. [72] proposed a novel SR prediction model based on energy consumption, considering SCEC as one of the main responses. PSO algorithm had been hybridized with SVM and RVM to optimize its different tuning parameters. The performance of SVM-PSO had been compared with RVM-PSO, and based on the experimental results, it was concluded that SVM-PSO would outperform RVM-PSO with respect to solution accuracy, making it an effective tool in predicting the considered responses. Jeet et al. [73] applied a super-hybrid optimization technique, i.e. Taguchi coupled weighted aggregated sum product assessment (WASPAS) with GA, SA and PSO for optimizing the turning of AISI 4340 steel to determine the ideal cutting conditions. The input parameters included CS, FR and DOC, while responses were SR, chip thickness and Tc. Zohra et al. [74] optimized the turning of AISI 4140 steel using GA. The study focused on SCEC as the main response, with CS, FR and DOC as the turning parameters, highlighting GA's capability in achieving lower SCEC.

After conducting turning experiments on high speed steel, Noor et al. [75] developed the corresponding nonlinear equations and optimized the said process using GA technique. It was revealed that an ideal combination of the distinct turning parameters as CS = 40 m/min, FR = 0.2 mm/rev and DOC = 0.5 mm would lead to maximum MRR, and minimum SCEC and Pc values concurrently. The optimal results had finally been contrasted against those derived using DFA. Nguyen et al. [76] employed GA to optimize the machining of SS 40XC, achieving the ideal combination of different process parameters. The study resulted in improvements of 8.91% in SCEC, 20.00% in SR and 14.75% in Mc, demonstrating its economic benefits. Similarly, Zebala et al. [77] optimized turning of titanium alloy using GA technique, considering CS, FR and DOC as input variables, and CF and Pc as the responses, achieving significant enhancement in machining efficiency. Based on GA technique, Vukelic et al. [78] solved the optimization problem of minimizing Pc, Vb, SR and CT, while conducting dry

turning experiments on Inconel 601, with CS, FR and DOC as the process parameters, reaffirming GA's effectiveness in optimizing machining of superalloy. Usha and Rao [79, 80] optimized machining of AISI 1045 and AISI 1040 steels using GA, incorporating MQL flow rate, nanoparticle volume concentration, CS, FR and DOC as input parameters, and CF, SR and Tc as the responses. Their results indicated notable reductions in both CF and SR.

In a subsequent study, Laouissi et al. [81] examined the influences of three turning variables, i.e. SS, FR and DOC on Pc, MRR, SR and CF during turning of EN-GJL-250 cast iron, and later optimized the process using GA. Serra et al. [82] developed first-order empirical models using multi-linear regression model to establish the relationships between the machining parameters and responses. The efficiency of the derived models was validated using ANOVA. Those models were used for multi-objective optimization with weighting factors and GA. An industrial example later confirmed the proposed methodology's effectiveness by comparing the optimized values against the experimental results. Abbas et al. [83] performed the turning operation on AISI 1045 steel, and determined the ideal parametric intermix as CS = 156.5 m/min, FR = 0.050 mm/rev and DOC = 0.57 mm. It was observed that GWO would perform better than PSA and GA in optimizing the process under consideration, while providing the corresponding response values as MRR = 4460.25 mm^3/min, SR = 0.719 μm, CF = 161 N and Tc = 463.5°C. Fountas et al. [84] developed full quadratic regression models to correlate different machining conditions with the machinability characteristics during longitudinal turning of CuZn39Pb3 brass alloy. The authors successfully optimized the considered responses using an advanced artificial GWO algorithm, achieving optimal turning parameter values.

Nouioua et al. [85] decided the optimal values of SR, Pc and CF during turning of X210Cr12 steel using four different metaheuristics, i.e. SA, GWO, ALO and VO, and noticed that all the considered algorithms would present almost the same parametric mixtures of FR, CS, DOC and Nr. Nguyen [86] developed an ANFIS-SA model and a RSM-DFA approach to optimize the rotary turning process on 9XC. The ANFIS-SA model determined the optimal parameter settings as CS = 200 m/min, FR = 0.4 mm/rev, DOC = 0.15 mm and CA = 18°. Under that condition, MRR had been increased by 33.16%, while improvements in SCEC and SR were 50.29% and 19.77%, respectively. The results demonstrated that ANFIS-SA model would provide more reliable and superior optimization outcomes as compared to RSM-DFA approach. Tanvir et al. [87] applied a hybrid WOA technique to optimize three turning parameters, i.e. CS, FR and DOC, achieving effective multi-response optimization results. The optimal parametric combination was determined as CS = 42.64 m/min, FR = 0.14 mm/rev and DOC = 0.32 mm. The study focused on optimizing multiple responses, including SR, CF, Pc, Tc, MRR and heat rate, demonstrating effectiveness of the hybrid WOA in machining process optimization. To unravel the parametric optimization problem during turning of polyoxymethylene POM-C material, Hakmi et al. [88] optimized the said process by integrating an ANN with a novel multi-objective AHA technique, proposing a fast and efficient computational tool. The ANN-AHA technique successfully optimized the cutting conditions, demonstrating its effectiveness for multi-objective optimization.

Oussama et al. [89] conducted an experimental study on the turning process of AISI 316L stainless steel. The ANOVA was employed to assess the influence of each machining parameter on the outputs. Based on the NSGA-II optimization results and TOPSIS analysis, the optimal machining conditions were identified as CS = 160 m/min, FR = 0.08 mm/rev, DOC = 0.5 mm and the lubricating condition as MWCNT-reinforced nanofluid-assisted MQL method. Those conditions were recommended as the effective solutions for achieving lower SR, while minimizing both feed force and Tc.

Chihaoui et al. [90] analyzed SR, Pc and MRR during machining of grey cast iron EN-GJL-250. The findings indicated that both GRA and NSGA-II/TOPSIS could provide the most effective solutions for achieving high quality machining with lower Pc. Benkhelifa et al. [91] optimized dry turning of AISI 316L using ANN-NSGA-II approach, which had been proved highly effective for industrial applications compared to DFA. The adopted method accurately predicted SR, Vb and Tc, leading to optimized cutting conditions and improved production control. Using NSGA-II technique, Jia et al. [92] optimized the turning operation of 45# steel to minimize SCEC, Pc and SR. The Pareto optimal solution set was obtained and experiments on a CNC lathe resulted in 38.3% reduction in SCEC and 47.0% decrease in SR, confirming effectiveness of the proposed approach. Risco-Alfonso et al. [93] developed a mathematical model to optimize CF and FR using NSGA-II during dry turning and MQL machining of AISI 316L steel. The study effectively identified the most relevant solutions for concurrently minimizing CF, Vb and SR. Bagaber and Yusoff [94] compared the multi-objective optimization solutions applying both RSM and NSGA-II, followed by experimental confirmation tests. The second-order NSGA-II results showed over 70% improvement in the response values against RSM technique. The proposed model proved to be effective in optimizing machining energy, cost and environmental impact, making it suitable for sustainable machining. Pendokhare and Chakraborty [95] optimized the turning operation on AISI 6061-T6 aluminum using five human-inspired metaheuristic algorithms, i.e. TLBO, SAR, TOA, HCO and QSA, by analyzing the past experimental data. Their comparative performance had been evaluated based on solution accuracy, computational effort and overall efficiency. A Pareto optimal front was developed to identify the optimal parametric intermix for the multi-objective optimization problem, making trade-offs among various performance measures. Among the considered algorithms, TLBO had been emerged as the most effective one, excelling in optimizing MRR, SR and SCEC. Its superiority was further validated through statistical quality metrics, including SP and HV, along with critical analysis of performance using Friedman's mean rank and Wilcoxon rank sum tests.

Most of the prior research works, as outlined in Table 2.1, have focused on optimizing turning operations using different discrete metaheuristic algorithms. In some of the cases, the authors have also compared consistency of the derived solutions against other randomly chosen techniques developed on different governing principles.

Table 2.1 Applications of different metaheuristics for optimization of turning processes

Author(s)	Work material	Turning parameters	Response(s)	Design plan	Optimization method(s)
Abderazek et al. [65]	EN-GJL-250 cast iron	CS, FR, DOC	Pc, SR, CF	L_{36} OA	DE
Arunkumar and Devendiran [66]	Nimonic 75	CS, FR, CA, nozzle distance	SR, CF, residual stress	L_{27} OA	MFA
Arunkumar and Devendiran [67]	Nimonic 80A	CS, FR, DOC	SR, CF, Vb, cutting acceleration	L_{27} OA	PSO, MFA
Doan et al. [68]	Titanium alloy	CS, FR, DOC, CA	SR, comprehensive energy, turning noise	BBD	PSO
Doan et al. [69]	SKD11 steel	CS, FR, DOC, CA	SR, SCEC	BBD	PSO
Shastri et al. [70]	Titanium alloy	CS, FR, CA	CF, SR, Vb	BBD	PSO, CIA
Elsheikh et al. [71]	AISI 4340	CS, FR, DOC	CF, SR, tool wear rate	L_{16} OA	PO, PSO
Xie et al. [72]	C45E4 steel	SS, FR, DOC	MRR, SR, SCEC	FFD	PSO
Jeet et al. [73]	AISI 4340	CS, FR, DOC	SR, chip thickness, Tc	L_9 OA	GA, SA, PSO
Zohra et al. [74]	AISI 4140	CS, FR, DOC	SCEC	L_{16} OA	GA
Noor et al. [75]	High speed steel	CS, FR, DOC	SCEC, Pc, MRR	FFD	GA
Nguyen et al. [76]	SS 40XC	CS, FR, DOC, CA	SCEC, SR, Mc	L_{27} OA	GA
Zebala et al. [77]	Titanium alloy	CS, FR, DOC	CF, Pc	L_{16} OA	GA
Vukelic et al. [78]	Inconel 601	CS, FR, DOC	Pc, Vb, SR, CT	FFD	GA
Usha and Rao [79]	AISI 1045	CS, FR, DOC, flow rate of MQL, volume concentration of nano-particles	CF	CCD	GA
Usha and Rao [80]	AISI 1040	CS, FR, DOC, flow rate of MQL, volume concentration of nano-particles	CF, SR, Tc	CCD	GA
Laouissi et al. [81]	EN-GJL-250 cast iron	CS, FR, DOC	Pc, SR, CF, MRR	L_{27} OA	GA
Serra et al. [82]	AISI 52100	CS, FR, DOC	SR, MRR, CF, CT, Pc	FFD	GA
Abbas et al. [83]	AISI 1045	CS, FR, DOC	MRR, SR, CF, Tc	FFD	GWO, GA, PSA
Fountas et al. [84]	Brass alloy	SS, FR, DOC	Pc, SR, CF	L_{18} OA	GWO
Nouioua et al. [85]	X210Cr12 steel	CS, FR, DOC, Nr	SR, Pc, CF	BBD	SA, ALO, GWO, VO
Nguyen [86]	Steel 9XC	CS, FR, DOC, CA	SCEC, SR, MRR	L_{27} OA	SA
Tanvir et al. [87]	SS 304	CS, FR, DOC	SR, CF, Pc, Tc, MRR, heat rate	CCD	WOA
Hakmi et al. [88]	Polyoxymethylene POM-C	CS, FR, DOC, Nr, CA	SR, CF, Pc, MRR	L_{16} OA	AHA
Oussama et al. [89]	SS AISI 316L	FR, CS, DOC, lubrication condition	SR, feed force, Tc	L_{27} OA	NSGA-II
Chihaoui et al. [90]	EN-GJL-250 gray cast iron	CS, FR, DOC	SR, Pc, MRR	L_{27} OA	NSGA-II
Benkhelifa et al. [91]	AISI 316L	CS, FR, DOC	SR, Vb, Tc	L_{27} OA	NSGA-II
Jia et al. [92]	45# steel	SS, FR, DOC	SCEC, Pc, SR	L_9 OA	NSGA-II
Risco-Alfonso et al. [93]	AISI 316L	CS, FR, lubrication condition	CF, Vb, SR	FFD	NSGA-II
Bagaber and Yusoff [94]	Stainless steel	CS, FR, DOC	SCEC, SR, Mc	CCD	NSGA-II
Pendokhare and Chakraborty [95]	AISI 6061-T6 aluminium	CS, FR, DOC	SCEC, SR, MRR	CCD	TLBO, SAR, TOA, HCO, QSA

2.2 Optimization of the green machining processes

The technique of dry milling involves removing material from the workpiece with no or minimum liquid lubrication. Numerous studies have already been conducted to investigate the effects of various dry milling parameters on the responses. Selection of the optimal combinations of those input parameters based on different mathematical techniques has piqued much interest among the past researchers leading to improved machining performance. Table 2.2 enlists applications of various mathematical techniques (mainly metaheuristic algorithms) considered for parametric optimization of green machining processes along with other relevant information, like experimental design plan employed, type of the work material, input parameters and responses. It can be noticed from Table 2.2 that GA [96-101], PSO [96, 103-109, 113], FOA [98], NSGA-II [102], LMBPA [105], SA [110], TS [111], GZOA [112], DOA [113], AGTO [113], ACO [113], GWO [113] and HHO [113, 114] have already become much popular among the research community. It is also revealed that most of the green machining experiments have been performed based on Taguchi's OA, CCD, BBD, FFD and D-optimal design plans on a variety of hard-to-cut work materials. But, there have been a very limited number of publications comparing optimization performance of the adopted algorithms with respect to computational effort, convergence speed and solution accuracy. Most of the algorithms have been discretely employed for parametric optimization of the green machining processes and there has been no attempt put forward to identify the best performing algorithm among its peers developed based on the same principle.

Based on a BBD plan, Zahoor et al. [96] investigated the influences of different process parameters on SR while performing milling operation on Inconel 718 superalloy, and later optimized the said process using GA and PSO. It was interestingly unveiled that both the optimizers had identified the same process parameter settings, resulting in achievement of the minimum SR value. Mumtaz et al. [97] investigated the optimization problem of machining parameters during MQL-assisted milling of AISI 1045 steel material, employing RSM and GA to analyze and optimize SR and Pc as the response variables. Moreira et al. [98] conducted the milling experiments on AISI 4340 work material, and employed both GA and FOA to optimize the said process. It was noticed that FOA would provide better results with respect to maximum MRR, and minimum SCEC and Pc. Deng et al. [99] optimized the milling process parameters for steel 16Mn using a quantum GA, simultaneously improving SCEC, SR, tool life and EC. Alrashdan et al. [100] employed regression analysis to model SR and EC during end milling of AISI D_2 steel. The optimization was performed using GA technique, considering SS, FR and DOC as the process parameters. Kadirgama et al. [101] optimized CNC end milling of AA6061-T6 to minimize SR using both RSM and radian basis function network. The derived results showed that FR had been the most influential factor, with radian basis function network providing more accurate SR predictions than RSM. The process was further optimized using GA.

Considering CS, FR, Dm and Wm as the input parameters, and EC and SR as the responses, Jia et al. [102] optimized the CNC milling operation on steel 45 material using NSGA-II technique, leading

to minimum values of both EC and SR. Mishra and Singh [103] applied PSO to ANN-based prediction models for CNC milling of Al-6061 T6, identifying the optimal parameter ranges to simultaneously enhance MRR and minimize CI. Furthermore, multi-objective PSO-optimized ANN model showed higher sensitive to input optimization. Li et al. [104] applied PSO technique to the milling of AISI 1045 material, demonstrating its effectiveness in resolving trade-offs between several conflicting objectives. The results highlighted the advantages of sustainable optimization methods, enabling simultaneous improvement in multiple responses, such as EC and CT. Jang et al. [105] optimized the milling process on SM45C structural steel using an ANN model with LMBPA and PSO to minimize SCEC and SR. The optimized cutting conditions resulted in a model prediction error of less than 1% compared to the experimental results. Yu et al. [106] conducted a multi-objective optimization study on hard milling of AISI H13 steel under MQL with graphite nanoparticles. The RSM and Kriging models were utilized to develop the corresponding mathematical regression models capturing relationships between the machining parameters and responses based on the experimental data. The optimization was performed using a multi-objective PSO algorithm combined with the Pareto approach.

Sen et al. [107] adopted a hybrid optimization approach combining GEP with PSO. A confirmation test showed a minimal average discrepancy of less than 3% between the experimental and GEP-PSO predicted results, demonstrating the robustness of the adopted approach. In [108], the authors investigated end milling of Inconel 690 using PSO algorithm, optimizing CS, FR and DOC. The study aimed to minimize SR, CF and CT, demonstrating effectiveness of PSO algorithm in improving the machining performance. Song et al. [109] optimized a milling process using PSO, considering SS, FR, Wm and Dm as the key parameters, affecting both SCEC and MRR. It was noticed that higher SS would improve surface finish, while higher FR would enhance MRR but might affect quality. The PSO algorithm effectively optimized the machining conditions with higher efficiency. On the other hand, Shi et al. [110] proved the efficacy of SA technique in optimizing the milling operation of Al 7050-T7451 alloy. Li et al. [111] used TS algorithm for CNC milling of steel 45, optimizing SS, FR, DOC and cutting width to enhance SCEC and MRR. Zhang et al. [112] demonstrated that GZOA had been highly effective in optimizing CNC milling operation of steel 45, exhibiting a superior convergence rate. The results indicated that the optimized machining process would achieve 13.37% reduction in SCEC and 6.26% decrease in SR. Mishra et al. [113] optimized the milling operation of S50C carbon steel using GWO, HHO, DOA, AGTO, ACO and PSO. The performance of those algorithms was evaluated based on solution accuracy, convergence speed and computation time. The results indicated that both GWO and HHO would demonstrate satisfactory performance across the considered metrics. Song et al. [114] optimized SR and MRR during milling of 30CrMnSiNiA steel by integrating HHO algorithm with TOPSIS method. A gaussian process regression-based prediction model was developed to predict the milling process responses, and its effectiveness was validated through comparative analysis with SVM and ANN models using experimental results.

Table 2.2 reveals that most of the previous studies have attempted to employ various metaheuristic algorithms to maximize performance of the green machining processes with minimum energy consumption and use of no or minimum liquid lubricant consumption. They have periodically compared accuracy of the derived optimal solutions against other algorithms developed based on different principles. No clear justification for the consideration of two or more metaheuristics in the optimization of green machining processes has been offered in any of the research investigations. They have been just chosen randomly.

Table 2.2 Literature review on applications of metaheuristics in optimizing green machining processes

Author(s)	Work material	Machining process	Parameters	Response(s)	Design plan	Optimization method(s)
Zahoor et al. [96]	Inconel 718	Milling	CS, FR, DOC	SR	BBD	GA, PSO
Mumtaz et al. [97]	AISI 1045	End milling	FR, SS, DOC	SR, Pc	BBD	GA
Moreira et al. [98]	BS EN24T alloy	CNC milling	CS, FR, SS	SCEC, MRR, Pc	L_{24} OA	GA, FOA
Deng et al. [99]	Steel 16 Mn	Milling	SS, CS, Dm, Wm	SR, SCEC, tool life, EC	L_{25} OA	GA
Alrashdan et al. [100]	AISI D_2 steel	End milling	CS, FR, DOC	SR, EC	FFD	GA
Kadirgama et al. [101]	AA6061-T6	CNC end milling	CS, FR, axial depth, radial depth	SR	BBD	GA
Jia et al. [102]	Steel 45	CNC milling	CS, FR, Dm, Wm	EC, SR	L_{16} OA	NSGA-II
Mishra and Singh [103]	Al-6061 T6	CNC milling	CS, FR, DOC	MRR, CI	CCD	PSO
Li et al. [104]	AISI 1045	Milling	SS, FR, Dm, Wm	SCEC, CT	L_{27} OA	PSO
Jang et al. [105]	SM45C structural steel	Milling	CS, FR, DOC, lubricant flow rate	SR, SCEC	FFD	PSO, LMBPA
Yu et al. [106]	AISI H13 steel	Hard milling	CS, FR, DOC, hardness of workpiece	SCEC, SR, MRR, CT	BBD	PSO
Sen et al. [107]	Inconel 690	End milling	CS, FR, DOC	EC, Mc, SR, Vb, carbon emission	L_{27} OA	PSO
Sen et al. [108]	Inconel 690	End milling	CS, FR, DOC	SR, CF, CT	L_{27} OA	PSO
Song et al. [109]	Al 6061	Milling	SS, FR, Wm, Dm	SCEC, MRR	L_{27} OA	PSO
Shi et al. [110]	Al-7050-T7451	Milling	CS, DOC, FR	Power	L_{27} OA	SA
Li et al. [111]	Steel 45	CNC milling	SS, FR, DOC, Wm	SCEC, MRR	L_9 OA	TS
Zhang et al. [112]	Steel 45	CNC milling	SS, FR	SCEC, SR	L_{25} OA	GZOA
Mishra et al. [113]	S50C carbon steel	CNC milling	SS, FR, DOC, Flowmeter rate	Carbon emission	L_9 OA	GWO, HHO, DOA, AGTO, ACO, PSO
Song et al. [114]	30CrMnSiNiA steel	Dry milling	SS, FR, DOC, Wm	SR, MRR	FFD	HHO

2.3 Optimization of the laser beam machining processes

Acknowledging the immense importance and suitability of LBD processes for generation of micro-holes in different hard-to-cut engineering materials in many of the technologically advanced industries, the past researchers attempted to employ numerous metaheuristic algorithms for deriving the optimal intermixes of different input parameters of those processes, as shown in Table 2.3. It can be noted from Table 2.3 that GA [115-130], GJO [130, 131], NSGA-II [132, 133], PSO [118, 134-137], APSO [138-140], WOA [139-143], CSA [144, 145], FA [145], TLBO [130, 146], FFA [147], AZOA [148], GTO [148], SFO [149], HHO [149], AO [149], BA [149] and GWO [149] have become most popular for optimizing the LBD processes.

Despite their computational complexity, metaheuristic algorithms can identify global or near-global optimal solutions due to their continuous search space, making them effective for optimizing LBD processes; however, their application has primarily focused on specific LBD processes, with limited comparative studies evaluating their optimization performance in terms of accuracy, variability of solutions and computational effort, whereas, extensive research has been conducted on machining process optimization using various nature-inspired and evolutionary metaheuristic algorithms, investigating different materials, input parameters, responses, experimental designs and optimization techniques to enhance machining performance and product quality.

Kumar et al. [115] optimized the machining of Inconel 718 using computer-aided GA-based multi-objective optimization technique. By considering Pg, Wp, Fp and CS as the input parameters, the study aimed to simultaneously optimize RLT at both the entry and exit of the drilled hole. During analysis of an LBM process, Singh and Gangwar [116] considered CS, Pg and Fp as the input parameters. The SR value for AISI 321 steel was determined using regression models, while GA was applied for parametric analysis and simultaneous optimization of SR and KT. Gowd et al. [117] optimized Cent and Cext as the hole geometry parameters, along with KT, using RSM and GA. To enhance the desired output responses during LBD operation, it was noticed that GA-based optimality analysis would yield improved fitness function values. Kalita et al. [118] optimized a laser beam micro-marking operation on gallium nitride using GA and PSO techniques, and observed that both of them would identify the same parametric settings as AC = 19.6 A, Fp = 4000 Hz and CS = 13 mm/s for having maximum value of mark intensity.

Employing a CCD approach for optimizing the machining of titanium alloys, Shrivastava and Pandey [119] optimized KD and KW using a hybrid technique integrating GA and multiple regression analysis. Their findings demonstrated effectiveness of GA technique in reducing machining defects and improving accuracy. Chatterjee et al. [120] optimized the machining process for AISI 316 using GA to determine the ideal parametric configurations. Empirical equations were employed as the objective functions in the optimization process. The results obtained through GA closely aligned with those derived from Taguchi approach, ensuring production of high quality laser drilled holes. Gautam and Mishra [121] adopted an integrated approach combining GRA and GA for single-objective optimization

of various kerf qualities. At the optimal laser parameter settings, KW, KD and KT had been significantly improved by 20.40%, 17.65% and 55.64%, respectively. Furthermore, the GA technique achieved an overall response improvement of 31.23%. Tura et al. [122] applied GA in conjunction with RSM to improve SR during machining of SS 304. A multi-objective GA achieved a minimum SR of 0.93746 µm with the optimal input parameters as CS = 2028.712 mm/m, Pg = 11.389 Bar and FPP = -2.499 mm. A comparative analysis of both the methods indicated that GA had outperformed RSM, demonstrating greater effectiveness in optimizing SR. Zhang et al. [123] optimized the machining of Ni-based superalloys by adjusting parameters, such as Pl, Fp, rotation speed and FR, to minimize processing time and KT. Application of GA effectively fine-tuned those parameters, resulting in reduced machining time and improved accuracy.

Du et al. [124] evaluated the relative importance of various LBM parameters through ANOVA and optimize them utilizing GA. The desired optimal parametric settings as Pl = 6.73 W, overlap rate = 99%, defocus rate = 0 mm, Pg = 0.2 MPa, FR = 0.02 mm and end position = - 0.4 mm would significantly improve Cent, Cext and TA of the holes, resulting in considerable enhancement in dimensional accuracy. Zhao et al. [125] applied GA technique to optimize the LBM process of Ni-based superalloys. Several key parameters, like Pl, Wp, Fp, defocusing amount etc. were considered during the optimization process. By integrating Gaussian support vector regression model with GA, the study achieved 93% reduction in TA and 107% increase in MRR, demonstrating effectiveness of GA in optimizing the machining parameters. A micro-drilling operation on composite material was optimized by Sahoo and Mishra [126] using GA, leading to identification of the optimal parametric intermix of different input parameters as LC = 180 A, Wp = 3 ms, SOD = 3 mm and Pg = 8 kg/cm^2. It was concluded that GA could effectively optimize the machining parameters, successfully validating the optimization results. Mahesh and Kandasamy [127] optimized the machining of GFRP by analyzing different laser drilling parameters, such as Pl, CS, focal length and Hd, to minimize both KD and KT. The authors employed mathematical modelling and optimization through RSM and multi-objective GA technique. Experimental validation had confirmed effectiveness of the adopted optimization tool to achieve significant reductions in KD and KT. Chatterjee et al. [128] predicted the performance metrics for laser drilling of AISI 316 and Ti6Al4V alloys using AI methods, like ANFIS and multi-gene genetic programming. It was inferred that multi-gene genetic programming models would outperform ANFIS models in terms of prediction accuracy. Yadav et al. [129] investigated the machining of Al/SiC metal matrix composite through laser beam percussion drilling process to enhance hole accuracy while minimizing Cent, Cext, TA and spatter size. Experimental findings indicated that reducing Wp and increasing Pg would lead to decrease in KA, improving Cent and Cext. GA-based optimization had significantly enhanced the process parameters, demonstrating their positive impact on drilling efficiency and accuracy.

During Nd:YAG laser drilling of GFRP and Kevlar-29, Sahoo and Mishra [130, 131] considered Wp, LC, Pg and SOD as the input parameters, while KT and OC as the responses variables.

Optimization had been performed using GA, TLBO and GJO techniques. Among them, GJO had demonstrated the fastest convergence. Confirmatory experiments validated its effectiveness, with results closely matching with the predicted outcomes. Nandi and Kuar [132] conducted research on statistical modeling to simultaneously minimize HAZ and KT in micro-drilling operations using an Nd:YAG LBM setup on ceramic work material. The RSM technique was employed for prediction, while multi-objective optimization was performed using NSGA-II. Ding et al. [133] proposed an integrated model combining a generalized regression neural network and NSGA-II with an elite strategy to predict and optimize various quality attributes during fiber laser cutting of stainless steel. The proposed generalized regression neural network-NSGA-II model could effectively predict and optimize the cutting parameters based on the experimental dataset. Shrivastava and Pandey [134] conducted an experimental investigation on pulsed Nd:YAG laser cutting of Inconel 718 material. The proposed hybrid PSO methodology, validated by the experimental results and a second-order regression model, had demonstrated significant improvement, specifically, enhancements of 75% in KD, 12.67% in KW and 33.70% in KT. Likewise, in [135], the output quality characteristics, i.e. KW and KT had been improved by approximately 10% and 57%, respectively. Overall, the optimization process resulted in a total improvement of 46% using PSO. Swain et al. [136] optimized the machining of K60 alumina ceramic using ANOVA, RSM and PSO. The study concluded that PSO had the significant potential for improving efficiency in the manufacturing industries. Pramanik et al. [137] applied AI techniques, along with PSO algorithm, for modeling and optimizing an LBM process. Their study focused on enhancing quality features using empirical findings and existing knowledge, with statistical analysis and experimental results confirming significance of all the input variables during the considered LBM process.

Mishra et al. [138] conducted an extensive study using APSO algorithm, observing significant performance improvements, and further confirmed that APSO could effectively optimize the responses during laser micro-drilling of nano-composites. Further investigations by Misra et al. [139] optimized the micro-drilling operation on polymer matrix nanocomposites using APSO and WOA techniques for achieving minimum taper and HAZ. The predicted response values had been better than those obtained using RSM. In another study, Mishra et al. [140] expanded their analysis using ANOVA and regression-aided APSO and WOA techniques. The MOORA-Taguchi method derived the corresponding equations for multi-objective WOA optimization, with confirmatory tests validating that APSO algorithm could predict better results. In a subsequent study, Mishra et al. [141] applied MOORA-Taguchi method with second-order regression equation development for multi-objective WOA optimization, while ANOVA had validated the model's adequacy. The adopted approach enhanced the multi-response optimization results for improved worker safety, product quality and sustainability. Mishra et al. [142] demonstrated WOA's superiority in reducing both KT and HAZ during polymer matrix composite machining, making it suitable for industrial applications to increase productivity. Finally, in a more comprehensive optimization approach, Mishra et al. [143] integrated GRA with WOA to improve machining

performance with the optimal parameter selection. Confirmation tests proved grey-WOA as a superior optimization approach.

Madić and Radovanović [144] optimized the AISI 304 machining process, by integrating ANN with CSA. The adopted approach could efficiently determine the optimal parameters for minimizing SR in laser cutting. On the other hand, the relative optimization performance of FA and CSA techniques was compared by Goswami and Chakraborty [145] for solving parametric optimization of two LBD processes. While both techniques had yielded similar parametric settings, CSA slightly outperformed FA in terms of average computation speed, convergence time and performance consistency. Pramanik et al. [146] used TLBO algorithm to optimize the LBM parameters for Monel material, applying both the single and multi-objective optimization results to refine KT and HAZ. Their study demonstrated TLBO as a powerful tool for optimizing the LBM parameters. Kar and Goswami [147] employed FFA to optimize the micro-drilling conditions for titanium grade 5 material. The FFA approach significantly improved the corresponding results, achieving minimal HC, KD and HAZ. Tamilarasan and Rajamani [148] investigated machining of AlZnMgCu1.5 alloy using ANOVA to evaluate the process factor's influences, identifying Pg as the most significant parameter. Their study focused on KD showed that GTO could provide faster and more accurate parametric values than AZOA. Confirmation tests had validated the optimal solution obtained through GTO, confirming its effectiveness in process optimization. Pendokhare and Chakraborty [149] investigated the applicability of five preying behavior-based metaheuristic algorithms, i.e. SFO, HHO, AO, BA and GWO, to optimize the CO_2 laser-based micro-drilling operation on polycarbonate material. The comparative analysis highlighted the superior performance of GWO over the other techniques in terms of solution quality, convergence speed and computational efficiency, achieving better hole quality with minimal HAZ. Those findings had been further validated through two statistical metrics, including SP and HV, along with statistical evaluations, like Friedman's mean rank and Wilcoxon rank sum tests, which could confirm GWO's dominance in LBD process optimization. It would be helpful to the process engineers to apply the most appropriate algorithm from a group of alternative metaheuristics resulting in global or near-global optimization of the considered LBD processes.

Table 2.3 A comprehensive literature review of optimization of LBD processes

Author(s)	Design plan	Material	Input parameters	Response(s)	Optimization method(s)
Kumar et al. [115]	L_{27} OA	Inconel 718	Pg, Wp, Fp, CS	RLT	GA
Singh and Gangwar [116]	L_9 OA	AISI 321	Pg, Fp, CS	SR, KT	GA
Gowd et al. [117]	CCD	AISI 304L	LC, Fp, Pg, Wp	Cent, Cext, KT	GA
Kalita et al. [118]	CCD	Gallium nitride	LC, Fp, CS	Mark intensity	GA, PSO
Shrivastava and Pandey [119]	CCD	Titanium alloy	Pg, Wp, Fp, CS	KD, KW	GA
Chatterjee et al. [120]	L_{27} OA	AISI 316	Pl, pulse repetition rate, Wp, Pg	Cent, Cext, KT, spatter area	GA
Gautam and Mishra [121]	BBD	Hybrid composite	LC, Wp, Fp, Pg, CS	KD, KW, KT	GA, GRA
Tura et al. [122]	L_9 OA	AISI 304	CS, Pg, FPP	SR	GA

Table 2.3 Contd.

Zhang et al. [123]	CCD	Ni-based superalloy	Pl, Fp, CS, FR	Processing time, KT	GA
Du et al. [124]	L_{25} OA	DD6 single crystal alloy	Pl, overlap rate, defocus rate, FR, Pg, end position	Cent, Cext, TA	GA
Zhao et al. [125]	FFD	Ni-based superalloy	Pl, Wp, Fp, defocusing amount	MRR, TA	GA
Sahoo and Mishra [126]	BBD	Composite	LC, Wp, SOD, Pg	TA	GA
Mahesh and Kandasamy [127]	BBD	GFRP	Pl, CS, focal length, Hd	KD, KT	GA
Chatterjee et al. [128]	L_{27} OA	AISI 316 and Ti6Al4V	Wp, LC, Fp, Pg	Cent, Cext, KT, HAZ, spatter area	GA
Yadav et al. [129]	CCD	Al/SiC metal matrix composite	Wp, Pg, LC	Cent, Cext, TA, spatter size	GA
Sahoo and Mishra [130]	BBD	GFRP and Kevlar-29	Wp, LC, Pg, SOD	KT, OC	GA, TLBO, GJO
Sahoo and Mishra [131]	BBD	GFRP and Kevlar-29	Wp, LC, Pg, SOD	KT, OC	GJO
Nandi and Kuar [132]	CCD	Ceramics	LC, Fp, Pg, Wp	HAZ, KT	NSGA-II
Ding et al. [133]	L_{27} OA	Stainless steel	Pl, CS, Pg, defocus	SR, KW	NSGA-II
Shrivastava and Pandey [134]	L_{27} OA	Inconel 718	Pg, SOD, CS, Pl	KD, KW, KT	PSO
Shrivastava and Pandey [135]	L_{27} OA	Inconel 718	Pg, SOD, CS, Pl	KW, KT	PSO
Swain et al. [136]	BBD	Ceramics	LC, Fp, CS, number of passes	KD	PSO
Pramanik et al. [137]	CCD	Ti-6Al-4V	Sawing angle, Pl, duty cycle, Fp, CS	KT, HAZ	PSO
Mishra et al. [138]	L_{25} OA	Composite	CS, LC, Pg, Wp	MRR, KT, HAZ	APSO
Mishra et a. [139]	CCD	Composite	Fp, CS, LC, Pg	HAZ, KT	APSO, WOA
Mishra et al. [140]	L_{25} OA	Composite	LC, CS, Pg, Fp, Wp	HAZ, MRR	APSO, WOA
Mishra et al. [141]	L_{25} OA	Composite	CS, LC, Fp, Pg, Wp	KT, MRR, HAZ	WOA
Mishra et al. [142]	BBD	Polymer matrix composite	CS, LC, Fp, Pg	KT, HAZ	WOA
Mishra et al. [143]	BBD	Composite	CS, LC, Fp, Pg	KT, HAZ	WOA, GRA
Madić and Radovanović [144]	L_{27} OA	AISI 304	CS, Pl, Pg, FPP	SR	CSA
Goswami and Chakraborty [145]	BBD	Ceramics	LC, Fp, Pg, job thickness, Pl, wait time, Wp	Cext, KT, MRR	FA, CSA
Pramanik et al. [146]	CCD	Monel	Sawing angle, power setting, duty cycle, Fp, CS	HAZ, KT	TLBO
Kar and Goswami [147]	FFD	Titanium grade 5	CS, Fp, number of passes, Pl	HAZ, KD, HC	FFA
Tamilarasan and Rajamani [148]	BBD	AlZnMgCu 1.5 alloy	Pg, PE, CS, Wp	KD	GTO, AZOA
Pendokhare and Chakraborty [149]	BBD	Polycarbonate	Pl, ET, FPP	Hh, Hd, HAZ	SFO, HHO, AO, BA, GWO

2.4 Optimization of the abrasive water jet machining processes

Based on the survey of literature, it is noticed that the past researchers have employed a wide range of strategies to improve performance of both the traditional machining and NTM processes. Due to the requirement of greater computational effort and chance of getting trapped in local optimal

solutions, the conventional optimization techniques, like exhaustive search method, direct search method, gradient-based method etc. are unable to identify the robust and global parametric combinations of those processes. It has already been mentioned that metaheuristic algorithms are capable of finding out the almost near global optimal solutions for complex and stochastic machining processes while keeping the constraints within their acceptable ranges. Table 2.4 shows applications of different metaheuristics in optimizing AWJM processes along with other relevant information, like work material machined, input parameters, responses and design plans employed. It is noticed from Table 2.4 that the past researchers have optimized AWJM processes using various metaheuristics, like HTS [27], GA [150, 151], MFO [150], NSGA-II [152, 156], PSO [153-156], SA [151, 155-157], CSA [155, 156], BHA [156], BBO [156], FFA [156, 163, 164], TLBO [158, 159, 162, 172], GWO [160-162], KHA [160], WOA [161], HSMO [162], CIA [164], ISSA [165], GSA [166], HSA [161, 174], GOA [167, 168, 175], RSO [169], AIS [170], JA [171, 172, 174], Rao algorithms [172], BO [173], DOA [174, 175], AVOA [175], FOA [175] and BSA [175]. It is also revealed that the earlier researchers have conducted AWJM experiments on various hard-to-cut work materials based on Taguchi's OA, CCD, BBD, FFD and D-optimal design plans, and identified the optimal parametric combinations of the considered processes.

Based on Taguchi's L_9 OA, Ganesan et al. [150] demonstrated that MFO algorithm had been more effective than GA in optimizing AWJM process on onyx composite materials. Zain et al. [151] optimized SR during Al7075-T6 alloy machining with GA and SA, proposing two integrated SA-GA systems. Both the systems could effectively estimate the optimal parameters, maximizing machining performance. Rao et al. [152] analyzed the interactions between different responses and machining variables. Second-order regression equations were developed and optimized using multi-objective NSGA-II technique. Kumar and Babu [153] optimized machining of NFRP composite using PSO, achieving the optimal values of SR, KT and MRR. Confirmation tests had also validated the experimental results obtained through the optimization process.

While optimizing the AWJM operation on soda lime silica glass, Ghosh and Martinsen [154] integrated PSO algorithm with ANN to develop an efficient prediction model and validated its superiority over GRA technique. Treating APS, AFR, WJP and St as the input parameters, and KA and SR as the responses, Gnanavelbabu and Saravanan [155] optimized the AWJM operation on Ti-6Al-4V alloy using PSO, CSA and SA techniques, and noticed that CSA would provide minimum values of both the responses. Sukla and Singh [156] conducted AWJM experiments and endeavored to optimize the said process using PSO, FFA, SA, CSA, BHA and BBO techniques. It was noticed that BBO would outperform the other algorithms with respect to convergence speed and variability of the solutions. Furthermore, a set of non-dominated solutions had been derived employing NSGA-II leading to multi-objective optimization of the AWJM process. Nabavi et al. [157] investigated the machining of AISI M24 material using an interpreted regression-SA approach. Their results demonstrated that the adopted tool could effectively and accurately identify the optimal cutting parameters to achieve the desired SR

values. Joel et al. [158] optimized the AWJM of C360 brass using multi-objective TLBO algorithm to minimize SR, and maximize MRR and hardness. Experimental validation had confirmed accuracy of the optimized parameters, achieving 0.1012 g/s MRR, 5.61 μm SR and 88.86 BHN hardness. Sharma et al. [159] studied AWJM of Ti-6Al-4V, evaluating the influences of different process parameters on SR and MRR through means plots and ANOVA technique. It was noticed that TLBO had demonstrated high efficiency, achieving the optimal results within just 3 seconds.

Applying GWO and KHA techniques, Balaji et al. [160] determined the optimal values of SR, cylindricity and Hd during AWJM operation on AISI 304 material. It was concluded that KHA could produce more accurate solutions, with an error margin of less than 2% compared to the experimental observations. Computation time and values of other considered quality matrices of KHA had also been lower than those of GWO technique. Dharmalingam et al. [161] investigated the effects of AFR, St and DD on entry and exit delamination and SR as the responses, while performing AWJM operation on 3D-printed onyx-glass fiber hybrid composites. The process was later optimized using GWO, HSA and WOA leading to identification of the optimal AWJM parameters achieving simultaneous minimization of delamination factors and SR. Jagadish et al. [162] optimized the AWJM operation on green composites using GWO, TLBO and HSMO techniques, and compared their performance with respect to solution accuracy, convergence speed and computation time. It was concluded that HSMO had marginally better solution accuracy than TLBO and GWO, and TLBO had the fastest convergence speed, followed by GWO algorithm. With respect to computation effort, GWO had the minimum time, followed by TLBO and HSMO. Palanikumar et al. [163] subsequently attempted to fabricate a novel NFRP composite laminate using compression molding. The study optimized the machining parameters through RSM and FFA. The derived results indicated that FFA could provide the lowest SR values, while RSM could result in minimal KW. Gulia and Nurgundkar [164] studied the application of CIA technique for optimizing the AWJM operation on AA631-T6 work material, and proved its superiority over FFA. Tamilarasan and Renugambal [165] first developed the second-order equations using RSM to model the AWJ cutting process with high precision. Their study found that ISSA could achieve a lower standard deviation compared to salp swarm algorithm, indicating that ISSA had been highly effective in optimizing the AWJ cutting parameters, enhancing machinability of green composites. Fuse et al. [27] optimized the machining of Ti-6Al-4V by integrating RSM with HTS algorithm. Their study validated the model's effectiveness in accurately predicting and optimizing the AWJM parameters.

Mokkandi et al. [166] utilized GSA to identify the optimal combination of the AWJM parameters. The corresponding regression models were developed to predict SR and KA for Al-NiTi-nano SiC composites. The results of applying GSA demonstrated that the identified optimal values would effectively enhance the machining performance. Adapa et al. [167] employed GOA to optimize the process variables for machining of lite bamboo reinforced polymer composites. The parametric analysis revealed that SOD and NS had been the most influential factors affecting MRR and SR during the AWJM process. Tamilarasan et al. [168] optimized the AWJ machining of ceramic tiles using RSM

and ANOVA. The authors applied elite opposition-based learning-GOA to minimize SR and KT, achieving only ±2% prediction error, demonstrating its effectiveness for the said AWJM process. Similarly, Tamilarasan et al. [169] optimized machining of titanium alloy using RSO algorithm to minimize KT, confirming the model's adequacy and reliability in determining the global optimal process parameters. Rajyalakshmi et al. [170] studied machining of Monel 400 material using ANOVA, optimizing the corresponding process parameters with AIS algorithm to achieve minimal KT and SR, and confirming its effectiveness in AWJM process optimization. Jagadish and Manjunath [171] conducted parametric analysis, ANOVA and regression analysis to evaluate the AWJM process for BFRP composites. The JA technique had optimized the machining conditions, achieving prediction error of 3.2% for MRR and 7.8% for SR, confirming its reliability for AWJM process optimization. Tripathi et al. [172] developed the related mathematical models using nonlinear regression equation to correlate different process variables and outputs. The authors applied Rao-1, Rao-2 and Rao-3 algorithms to determine the optimal machining conditions, and compared the derived results with JA and TLBO approaches. The study found that Rao algorithms could outperform others with respect to computational efficiency and effort, demonstrating their effectiveness in machining process optimization. Chaouch et al. [173] worked on FRP composite machining, and employed multi-objective BO approach to enhance KT and SR during the AWJM process. The study demonstrated that ANN with multi-objective BO would be an effective tool for selecting the optimal machining parameters. Finally, Vasudevan et al. [174] investigated the performance of Inconel 718 superalloy during the AWJM drilling process and applied a novel DOA approach. Comparative results showed that DOA would outperform HSA and JA with reference to drilling quality and efficiency. Pendokhare et al. [175] optimized an AWJM process using five foraging behavior-based metaheuristic algorithms, i.e. DOA, AVOA, GOA, FOA and BSA. Among them, AVOA had demonstrate maximum efficiency, outperforming other algorithms in terms of solution accuracy, variability and computational effort. It would enabled finer tuning of different critical process parameters, like WJP, SOD and St, leading to superior machining performance. The performance analysis of the considered algorithms had been assessed using two quality matrics (SP and HV) and validated through Friedman's mean rank and Wilcoxon rank sum tests.

Table 2.4 shows that the past researchers have attempted to optimize the performance of AWJM processes using some distinct metaheuristic algorithms, and occasionally compared their solution accuracy against other algorithms having dissimilar characteristics. In any of the research studies, no valid reason has been provided regarding why two or more metaheuristics have been considered for optimization of the AWJM processes. They have simply been chosen arbitrarily.

Table 2.4 Literature survey on applications of metaheuristics in optimizing AWJM processes

Author(s)	Material	Parameters	Response(s)	Design plan	Optimization method(s)
Ganesan et al. [150]	Onyx composite	AFR, DD, St	DLL at entry and exit, SR	L_9 OA	GA, MFO
Zain et al. [151]	Wrought alloy	St, WJP, SOD, AFR, APS	SR	L_{27} OA	GA, SA
Rao et al. [152]	GFRP	CS, FR, SOD	MRR, TW, SR	L_9 OA	NSGA-II
Kumar and Babu [153]	NFRP composite	St, AFR, SOD	SR, KT, MRR	BBD	PSO
Ghosh and Martinsen [154]	Soda lime silica glass	APS, FR, SOD, WJP, abrasive concentration	MRR, top KW, bottom KW, SR	L_9 OA	PSO, GRA
Gnanavelbabu and Saravanan [155]	Ti-6Al-4V	APS, AFR, WJP, St	KA, SR	BBD	PSO, CSA, SA
Sukla and Singh [156]	AA631-T6	St, AFR, SOD	Top KW, KA	L_{25} OA	PSO, FFA, SA, CSA, BHA, BBO, NSGA-II
Nabavi et al. [157]	AISI M24	AFR, WJP, St, SOD	SR	L_9 OA, D-optimal	SA
Joel et al. [158]	C360 brass	Abrasive feed, SOD, St	MRR, SR, hardness	BBD	TLBO
Sharma et al. [159]	Ti-6Al-4V	St, SOD, AFR	SR, MRR	L_9 OA	TLBO
Balaji et al. [160]	AISI 304	Abrasive mixture level, SOD, FR	SR, circularity, cylindricity, Hd	L_{18} OA	GWO, KHA
Dharmalingam et al. [161]	Onyx composite	AFR, DD, St	DLL at entry and exit, SR	L_9 OA	GWO, HSA, WOA
Jagadish et al. [162]	Green composites	WJP, St, SOD	Process time, SR	BBD	GWO, TLBO, HSMO
Palanikumar et al. [163]	NFRP composite	St, AFR, SOD	SR, KW	BBD	FFA
Gulia and Nurgundkar [164]	AA631-T6	WT, ND, SOD, St	SR, KW	L_{18} OA	FFA, CIA
Tamilarasan and Renugambal [165]	Composites	Abrasive concentration, WJP, SOD, AFR	Kerf inclination, DLL at entry and exit	CCD	ISSA
Fuse et al. [27]	Ti-6Al-4V	St, AFR, SOD	MRR, SR, KA	BBD	HTS
Mokkandi et al. [166]	Al-NiTi-nano SiC composites	WJP, SOD, St	SR, KA	L_{27} OA	GSA
Adapa et al. [167]	Lite bamboo-reinforced polymer	SOD, St, WJP	MRR, SR	L_9 OA	GOA
Tamilarasan et al. [168]	Ceramic	WJP, St, AFR, SOD	SR, KT	BBD	GOA
Tamilarasan et al. [169]	Titanium alloy	WJP, St, AFR, SOD, ND	KT	CCD	RSO
Rajyalakshmi et al. [170]	Monel 400	WJP, St, AFR, SOD	SR, KT	L_{16} OA	AIS
Jagadish and Manjunath [171]	BFRP composite	WJP, SOD, NS	MRR, SR	L_{27} OA	JA, DFA
Tripathi et al. [172]	GFRP composite	AFR, CS	SR, MRR, roundness, cylindricity	L_9 OA	JA, TLBO, Rao 1, 2, 3 algorithms
Chaouch et al. [173]	FRP composite	WJP, St, AFR, SOD	KT, SR	FFD	BO
Vasudevan et al. [174]	Inconel 718	WJP, AFR, SOD	Entry diameter, exit diameter, erosion diameter	FFD	DOA, HSA, JA
Pendokhare et al. [175]	Lanthanum phosphate/ Yttria composite	WJP, SOD, St	MRR, KA, SR	CCD	DOA, AVOA, GOA, FOA, BSA

2.5 Optimization of the plasma arc machining processes

The prior research works have already employed several techniques, such as Taguchi method and RSM to optimize and improve PAC settings. However, there have not been much research on parametric optimization of PAC processes utilizing various metaheuristic algorithms. It is imperative to operate any PAC process while setting its different input parameters at their ideal values because of their possible interactions with the responses and complicated cutting mechanism. For solving parametric optimization problems of PAC processes, the past researchers have adopted a variety of mathematical tools, particularly metaheuristic algorithms. Table 2.5 enlists their applications, along with other pertinent information, like experimental design plan, type of work material, input parameters and responses. It can be noticed from Table 2.5 that GWO [32], SA [32], PSO [32, 183], probabilistic multi-objective approach [35], HSA [183], WOA [32, 183, 184], GA [32, 37, 178-182, 185, 186], JA [32, 176, 177, 186], ABC [37, 187], TLBO [185, 186], MFO [188], WCA [189], AOA [190], ASO [190], NRO [190], EFO [190] and GSA [190] have already become much popular among the research community. It is also revealed that Taguchi's OA, CCD, BBD and FFD plans have been the basis of experiments for majority of the PAC studies, carried out on a range of difficult-to-cut materials. However, there have been a very limited number of publications comparing optimization performance of the adopted algorithms in terms of computational effort, convergence speed and accuracy of the solutions. Majority of them have been prudently deployed for parametric optimization of PAC processes and no effort has been put forward to identify the best-performing algorithm among its peers designed based on the same principle.

By utilizing an ANFIS model, Siva Kumar et al. [37] and Patel et al. [186] examined the impacts of different PAC parameters on various process outputs, and validated the derived findings against other widely used optimization techniques. Siva Kumar et al. [37] also conducted PAC experiments on Monel 400 material, and employed two hybrid approaches based on ANFIS-GA and ANFIS-ABC to optimize the said process. It was noticed that ANFIS-ABC would provide better results with respect to minimum SR, KW and MH. Zheng and Yu [35] investigated the influences of four input parameters, i.e. Vg, SOD, CS and plasma offset on SR, MRR and kerf ratio, while performing PAC operation on SS304 alloy. It had involved PAC with two different types of nozzle and optimizing the process using a probabilistic multi-objective approach.

The GA technique has been extensively employed in machining process optimization, as demonstrated by various studies. Masoudi et al. [178] developed predictive models for SS309 material using intelligent systems based on GA and ANN, incorporating a hybrid genetically optimized neural network for simultaneous process optimization. Their findings indicated that the adopted strategy could effectively enhance the output parameters of the said PAC process. Similarly, Hamdy et al. [179] optimized PAC parameters for mild steel using MOORA-GA method. The MOORA method transformed the multiple responses into a single multi-performance characteristic index, which had been modeled and optimized using GA, confirming its effectiveness in enhancing the PAC parameters. Dash

et al. [180] optimized the PAM of EN31 steel, applying GA and fuzzy logic. Their study aimed to minimize SR and maximize MRR, and demonstrated that both GA and fuzzy logic could effectively optimize the PAC process parameters. Additionally, Peko et al. [181] applied GA to optimize cut quality in PAC of EN AW-5083 alloy, achieving maximum MRR, while minimizing KW, bevel angle and SR.

The GA optimizer, when combined with other metaheuristic approaches, has demonstrated improved accuracy and efficiency in machining process optimization. Mangaraj et al. [32] determined the optimal values of MRR, HAZ, SR, KW, chamfer and DFR during PAC operation on AISI 304 work material. The findings from the hybrid RSM-evaluation based on distance from average solution (EDAS) had subsequently been compared against GA, GWO, JA, WOA, PSO and SA, revealing that all the metaheuristics would provide almost similar optimal parametric combinations. On the other hand, the optimization performance of a novel hybrid technique, i.e. ANN-GA was evaluated by Melaku and Bogale [182] while solving the corresponding parametric optimization problem of PAC of AISI 1020 material. An improved ANN based on LMBPA was subsequently developed and trained using the experimental data to achieve maximum prediction accuracy. It was concluded that the process optimization would result in improvements of 1.121, 3.250 and 4.269% for MRR, SR and ND, respectively. Similarly, Patel et al. [185] proved that TLBO would be more effective than GA in optimizing PAC operation on AISI D_2 steel material, and identified the corresponding process parameter settings as CS = 4000 m/min, Pg = 95 psi and TH = 0.5 mm, resulting in achievement of minimum SR values.

The JA has also been effectively applied in machining process optimization. Jamsari et al. [176] conducted PAC experiments on SS304 and S235 work materials, and employed JA to optimize the said process. It was noticed that JA would be able to quickly and accurately identify the optimal parametric combinations for minimizing both SR and KT. Rao et al. [177] optimized PAC operation on AISI 4340 material using JA, leading to maximum MRR and minimum DFR values. Additionally, Patel et al. [186] combined JA with TLBO and GA for AISI D_2 machining, while successfully optimizing SR and MRR. It was concluded that JA had a shorter convergence time, making its execution faster compared to TLBO and GA.

Hybrid metaheuristic approaches have demonstrated enhanced accuracy and efficiency in machining process optimization. Rajamani et al. [183] endeavored to optimize PAC of Monel 400 material using WOA, PSO and HSA, and compared their performance in regard of solution accuracy, convergence speed and computation time. It was concluded that WOA would outperform the other algorithms in successfully deriving the global optimal solutions. Similarly, WOA had also been effectively applied to optimize PAM processes, Karthick et al. [184] optimized Inconel 718 machining using a RSM approach and multi-parameter ANOVA. WOA had been applied for multi-response optimization, improving productivity and cutting quality while controlling MRR, KT and HAZ. Das et al. [187] optimized EN 31 machining employing GRA and ABC techniques, leading to improved response values. Karthick et al. [188] proved the efficacy of MFO technique in optimizing PAC

operation on Inconel 718. It was noticed that MFO would be quite feasible in prediction and optimization of the said PAC process for minimum SR, MH and KD. Tamilarasan et al. [189] optimized the cutting of Ti-6Al-4V alloy to achieve maximum MRR using RSM, incorporating ANOVA, regression analysis and a quadratic model. The WCA had been employed for optimal PAC parameter selection, with confirmation experiments validating the predictions, yielding a relative error of only 2.688%. Pendokhare and Chakraborty [190] proposed the application of five newly developed physics-based metaheuristic algorithms, i.e. AOA, ASO, NRO, EFO and GSA, for optimization of a PAC process. The results indicated that EFO algorithm would outperform its competitors in achieving the optimal process parameter settings, including CS, Pg, AC and SOD to optimize MRR, KT and HAZ. It had effectively balanced the computing effort, convergence time and solution quality while delivering high-quality cuts with minimal material wastage and lower EC. Statistical assessments, including quality metrics, non-parametric ranking-based evaluations and comparative analysis against some popular state-of-the-art optimization methods, had further validated EFO's effectiveness in optimizing the said PAC process.

It is evident from the review of the literature [32, 35, 37, 176-190] that KT, MRR, HAZ, KW, MH and SR have been considered as the most significant quality indicators evaluating performance of the PAC operations. Majority of the earlier researches, as shown in Table 2.5, have attempted to optimize performance of PAC operations while utilizing a variety of metaheuristic algorithms. The accuracy of the derived solutions has occasionally been compared against other randomly selected algorithms proposed using dissimilar guiding principles.

Table 2.5 Literature survey on applications of different metaheuristics in optimizing PAC processes

Author(s)	Material	Parameters	Response(s)	Design plan	Optimization method(s)
Zheng and Yu [35]	SS304	Vg, SOD, CS, plasma offset	SR, MRR, kerf ratio	L_{18} OA	Probabilistic multi-objective approach
Jamsari et al. [176]	SS304, S235	SOD, Pg, CS	SR, KT	FFD	JA
Rao et al. [177]	AISI 4340	CS, AC, TH, Vg	MRR, DFR	CCD	JA
Masoudi et al. [178]	SS309	Pg, CS, AC	SR, KW, HAZ	FFD	GA
Hamdy et al. [179]	Mild steel	CS, AC, SOD	MRR, SR, KW, DFR	L_{27} OA	GA
Dash et al. [180]	EN 31	Pg, AC, TH	MRR, SR	L_{27} OA	GA
Peko et al. [181]	EN AW-5083	CS, AC, cutting height	KW, bevel angle, SR, MRR	FFD	GA
Siva Kumar et al. [37]	Monel 400	SOD, Pg, CS, AC	SR, KW, MH	BBD	GA, ABC
Mangaraj et al. [32]	AISI 304	SOD, Pg, CS, AC	MRR, HAZ, SR, KW, chamfer, DFR	BBD	GA, PSO, JA, WOA, SA, GWO
Melaku and Bogale [182]	AISI 1020	SOD, Pg, CS, AC	MRR, SR, ND	L_{27} OA	GA, LMBPA
Rajamaniet al. [183]	Monel 400	SOD, Pg, CS, AC	MRR, KT, HAZ	BBD	WOA, PSO, HSA
Karthick et al. [184]	Inconel 718	Pg, AC, TH, CS	MRR, KT, HAZ	BBD	WOA
Patel et al. [185]	AISI D2	CS, Pg, TH	SR	L_{16} OA	TLBO, GA
Patel et al. [186]	AISI D2	CS, Pg, TH	MRR, SR	L_{16} OA	TLBO, JA, GA
Das et al. [187]	EN 31	Pg, AC, TH	MRR, SR	CCD	ABC
Karthick et al. [188]	Inconel 718	Pg, AC, TH, CS	SR, MH, KD	BBD	MFO
Tamilarasan et al. [189]	Ti-6Al-4V	Pg, AC, TH, CS	MRR	BBD	WCA
Pendokhare and Chakraborty [190]	Monel 400	SOD, Pg, CS, AC	MRR, KT, HAZ	BBD	AOA, ASO, NRO, EFO, GSA

3. AN OVERVIEW OF METAHEURISTIC ALGORITHMS

Metaheuristic algorithms have been widely deployed for optimizing various machining processes characterized by nonlinear relationships between the input parameters and responses. As problem-independent, high-level algorithmic frameworks, they provide a toolkit of techniques for developing heuristic optimization solutions. These algorithms are designed to obtain a 'good enough' solution within a 'small enough' computation time, balancing solution quality and processing efficiency. Their primary advantage lies in offering a trade-off between exploration (global optima) and exploitation (local optima), ensuring adaptability to real-time optimization challenges while delivering high quality solutions in less computing time [53, 54]. However, according to the 'No-Free-Lunch' theorem, while metaheuristics are highly effective for certain classes of optimization problems, they cannot provide universal solutions for all optimization challenges [191].

Nature-inspired metaheuristics draw inspiration from various natural and social phenomena. Metaheuristic algorithms have become indispensable tools in engineering, AI and operational research, offering robust and efficient solutions for complex optimization problems. This overview categorizes metaheuristic algorithms based on human-inspired, mating behavior-based, preying behavior-based, foraging behavior-based and physics-based approaches. Table 3.1 shows the common parameters used in all the considered algorithms.

Table 3.1 Common parameters used in all the considered algorithms

Parameter(s)	Value(s)
Maximum number of iteration (*IT*)	100
Size of the population (*N*)	50
No. of independent runs	30

3.1 Human-inspired metaheuristics

Humans take immense pleasure in thinking and acting unpredictably in their capacity. Despite significant scientific advancements in the last century, understanding of the human brain still remains incomplete. Human intelligence is defined by the ability to understand, make decisions, reason, learn, resolve problems, innovate, remember and communicate. Incorporating these human cognitive abilities into optimization techniques, also elevates the potential of solving complex and real-world challenges. Unlike other nature-inspired evolutionary processes, like swarm and physics-based algorithms, human-inspired approaches adapt to their environments at much faster rate, resulting in development of several optimization algorithms inspired by human factors [192].

Selection of the optimal settings of several algorithmic parameters is also significant as they greatly influence the search characteristic, convergence behavior and grade of the solutions. Table 3.2 shows of the critical tuning parameters for each of the five algorithms under consideration.

Table 3.2 Tuning parameters for different human-inspired metaheuristics

Algorithm	Parameter	Value
TLBO	-	-
SAR	SE MU	0.05 70D (D denotes dimension of the problem)
TOA	Update factor	[0, 1]
HCO	P_{fit} γ w C_1 and C_2	0.65 Random number (0 to 1) 0.1 Random number (2 to 4)
QSA	-	-

3.1.1 Teaching learning-based optimization algorithm

TLBO is a population-based optimization technique [193] inspired by the teaching and learning processes occurring in a classroom. It has two primary phases, e.g. teacher phase and learner phase. The teacher shares knowledge with the students, attempting to shift the mean performance of the class towards the optimum. The students (learners) engage in interactions with each other to enhance their knowledge by learning from their peers. Simplicity, absence of algorithm-specific parameters, faster convergence, compromised balance between global and local search processes, and broad applicability make it a strong contender in the domain of optimization, particularly for complex, multimodal and constrained problems [193].

Step 1: Initialization

This algorithm randomly generates and initializes a population of learners (*N*), and subjects (courses) are represented by the number of design variables (*V*). Each learner is a potential solution to the current problem.

Step 2: Teacher phase

The teacher (best solution) shares knowledge to improve the entire class's (population) mean performance. The learning process is driven by moving the population towards the teacher.

Let $X_{teacher}$ be the best solution (the teacher) in the population, T_{mean} be the mean of all the learners' solutions in the population, r_i is a randomly generated number between 0 and 1, and T_F is the teaching factor, typically either 1 or 2, representing how much knowledge the teacher shares with the learners.

The update equation for each learner, X_i in the population during the teacher phase is as follows:

$$X_{new} = X_i + r_i \times (X_{teacher} - T_F \times T_{mean}) \tag{3.1}$$

where X_i is the current solution of i^{th} learner, X_{new} is the updated solution after the teacher influences the learner, T_F is determined as $T_F = \text{round}[1 + rand(0, 1)]$, and the mean T_{mean} of the population is calculated as $T_{mean} = \frac{1}{N}\sum_{i=1}^{N} X_i$.

The teaching factor allows the teacher to impart more or less influence over the learners.

Step 3: Learner phase

During the learner phase, students interact with others to improve their knowledge. A learner X_i randomly selects another learner X_j from the population and updates its solution based on whether the chosen learner X_j has a better fitness than the learner X_i.

The update equation depends on the fitness comparison between X_i and X_j, where $i \neq j$.

If $f(X_i) < f(X_j)$ then $X_{new} = X_i + r_i \times (X_i - X_j)$ (3.2)

If $f(X_j) \leq f(X_i)$ then $X_{new} = X_i + r_i \times (X_j - X_i)$ (3.3)

where $f(X_i)$ and $f(X_j)$ are the fitness values of learners X_i and X_j, respectively.

Algorithm 3.1.1: Pseudo-code of TLBO

```
Randomly initialize the population (learners) with solutions (X_i for i = 1 to N) and V
Evaluate each learner's fitness (X_i)
while stopping criterion (e.g. maximum no. of generations) is not fulfilled
   Identify X_teacher in the population
   Update learners based on the teacher's influence using Eq. (3.1)
   if X_new is better than X_i, replace X_i with X_new
for each learner X_i, randomly select another learner X_j
   Compare the fitness of X_i and X_j considering Eqs. (3.2) and (3.3)
if X_new is better than X_i, replace X_i with X_new
   Evaluate fitness of the new population
end while
Return X_teacher
```

3.1.2 Search and rescue optimization

SAR algorithm [194] is motivated by the behavior of search and rescue missions carried out by humans. It basically mimics the dynamic process of locating the victims in unknown or hazardous environments. It is proposed to handle multi-objective continuous optimization problems by effectively balancing exploration and exploitation in the search space [194]. SAR is a powerful algorithm for working out real-world optimization problems involving complex and high-dimensional search space. Its flexibility, adaptability and robust performance make it a popular choice for dealing with diverse optimization challenges [195].

Phase 1: Social phase

In this algorithm, new solutions are generated during both the social and individual phases while utilizing a clues matrix. This matrix is a vital component guiding the search process. The key matrices, i.e. *P*, *M* and *C*, are continuously revised in each phase of human searches, ensuring that the agents' movements are based on the latest information. These matrices collectively help balance exploration (social phase) and exploitation (individual phase) of the search space, improving the optimization process. Equation (3.4) is applied to develop the clues matrix (*C*).

$$C = \begin{bmatrix} P \\ M \end{bmatrix} \quad (3.4)$$

where *M* and *P* denote the position matrices of memory and humans, respectively.

The search direction is determined using Eq. (3.5), and a random clue is selected from the set of identified clues, as explained in the first phase.

$$SD_i = (P_i - C_k),\ \ k \neq i,\ \ (i = 1,\ldots,N) \tag{3.5}$$

where P_i, C_k and SD_i represent the i^{th} agent (human) position, k^{th} clue position and i^{th} agent search direction, respectively. The index k is a random integer between 1 and $2N$ (N is the total number of agents).

The search would continue in the direction of SD_i and around the present position. When it comes to the social phase, Eq. (3.6) is used.

$$P'_{i,j} = \begin{cases} \begin{cases} C_{k,j} + r_1 \times \left(P_{i,j} - C_{k,j}\right), & if\ f(C_k) > f(P_i) \\ P_{i,j} + r_1 \times \left(P_{i,j} - C_{k,j}\right), & \text{otherwise} \end{cases} & if\ r_2 < SE\ or\ j = j_{rand}\ \ (j = 1, \ldots, D) \\ P_{i,j} & \text{otherwise} \end{cases} \tag{3.6}$$

where the new position of i^{th} agent in j^{th} dimension is denoted as $P'_{i,j}$, and $C_{k,j}$ is j^{th} dimension position of k^{th} clue being influenced by the objective function values $f(C_k)$ and $f(P_i)$. The above equation employs two uniformly distributed random numbers, i.e. r_1 ranging from [−1, 1] and r_2 ranging from [0, 1]. The value of *SE* (social effect) is set as 0.05.

Phase 2: Individual phase

In this phase, humans search within the vicinity of their present positions, utilizing the concept of linking various clues, which is commonly applied in the social phase. Compute the revised position of i^{th} human based on the following equation:

$$P'_i = P_i + r_2 \times (C_k - C_m) \qquad i \neq k \neq m \tag{3.7}$$

where m is a random integer lying between 1 and $2N$. To avoid movement along the redundant clues, k and m are selected so that $i \neq k \neq m$.

Phase 3: Boundary control

The solutions generated from both the social and individual phases must reside within the defined solution space. If any solution falls outside this space, it needs to be adjusted accordingly. To modify the new position of i^{th} human, Eq. (3.8) is employed.

$$P'_{i,j} = \begin{cases} \frac{(P_{i,j} + P_j^{max})}{2}, & if\ P'_{i,j} > P_j^{max} \\ \frac{(P_{i,j} + P_j^{min})}{2}, & if\ P'_{i,j} > P_j^{min} \end{cases} \quad (j = 1, \ldots . D) \tag{3.8}$$

where P_j^{max} is the maximum and P_j^{min} is the minimum threshold values for j^{th} dimension, respectively.

Phase 4: Update position

The newly generated solution, P'_i, obtained from either of the two phases, would be placed in an arbitrary location within the matrix M. This would become the new position if it proves to be superior to the previous solution, P'_i. Otherwise, the memory remains unchanged, discarding the new result. This process is described using Eqs. (3.9) and (3.10).

$$M_n = \begin{cases} P_i, & if\ f(P_i') > f(P_i) \\ M_n, & \text{otherwise} \end{cases} \tag{3.9}$$

$$P_i = \begin{cases} P_i', & if\ f(P_i') > f(P_i) \\ P_i, & \text{otherwise} \end{cases} \tag{3.10}$$

where the location of n^{th} stored clue in the memory matrix is denoted by M_n. A random integer between 1 and *N* is represented by *n*.

Phase 5: Abandon clues

Missing individuals may have sustained injuries, making it crucial to search the area as quickly as possible. Therefore, humans must stop searching around the unsuccessful clues after a certain amount of effort. To simulate this behavior, the number of failed search attempts by each individual is tracked. The unsuccessful search number (*USN*) for each individual is initially set to zero. Additionally, when a human finds better clues, *USN* is reset to zero; otherwise, it increases by 1 with each unsuccessful search, as presented in Eq. (3.11).

$$USN_i = \begin{cases} USN_i + 1, & if\ f(P_i') < f(P_i) \\ 0, & \text{otherwise} \end{cases} \tag{3.11}$$

When a solution fails to improve after a specified number of iterations, it is discarded. Subsequently, a new solution is formed applying Eq. (3.12), replacing the previous solution with the new one.

$$P_{i,j} = P_j^{min} + r_2 \times \left(P_j^{max} - P_j^{min}\right) \quad (j = 1, \ldots\ldots, D) \tag{3.12}$$

Algorithm 3.1.2: Pseudo-code of SAR

Initiate with a population having 2*N* solutions
Sort the solution and find P_{best}
Use the sorted solutions for *P* and *M*
Define *SE*, maximum unsuccessful search number (*MU*) and $USN_i = 0$
While stopping criterion is not fulfilled do
  for *i* =1 to *N* do
Social phase
Revise *C* using Eq. (3.4)
Generate SD_i applying Eq. (3.5)
for *j* =1 to *D* do
    Calculate new position of j^{th} dimension for i^{th} human using Eq. (3.6)
    Modify boundary control based on Eq. (3.8)
  end for
Revise M_n using Eq. (3.9)
  Update P_i applying Eq. (3.10)
Revise USN_i using Eq. (3.11)
Individual phase
Revise *C* employing Eq. (3.4)
Calculate P_i' using Eq. (3.7)
Modify boundary control applying Eq. (3.8)
Revise M_n of n^{th} memory using Eq. (3.9)
  Revise P_i of i^{th} human considering Eq. (3.10)
  if $USN_i > MU$ do

for j = 1 to D do
Calculate $P_{i,j}$ using Eq. (3.12)
end for
$USN_i = 0$
end if
end for
Identify the present best position and modify P_{best}
end while
Return P_{best}

3.1.3 Teamwork optimization algorithm

TOA is developed on the principle of how teams of individuals collaborate to achieve a common goal [196]. Its basic idea is to simulate a group of agents (team members), each contributing to the improvement of a solution, and collectively working towards an optimal solution through coordination and communication. TOA mimics the dynamics of teamwork, where individuals contribute their strengths while learning from others. Its adaptive nature, flexibility, capability of self-improvement, collaboration, rapid convergence towards high quality solutions and robust performance are responsible for its immense popularity for solving complex optimization problems [196].

Stage 1: Supervisor selection and guidance

Supervisors are responsible for guiding and leading a team of members. Their performance tends to be more efficient compared to that of the team members. Equation (3.13) is considered to describe how the supervisor would be selected, and Eqs. (3.14)-(3.16) are used to update the supervisor's step.

$$S = Z_s \tag{3.13}$$

$$Z_i^{S1}: z_{i,d}^{S1} = z_{i,d} + r_i \times \left(S_d - I \times z_{i,d}\right) \tag{3.14}$$

$$Z_i = \begin{cases} Z_i^{S1}, & V_i^{S1} < V_i \\ Z_i, & \text{else} \end{cases} \tag{3.15}$$

$$I = round(1 + r_i) \tag{3.16}$$

where S represents the team supervisor, Z is the population matrix, s refers to the row number of the team member having minimum value in vector V of the objective function, Z_i^{S1}indicates the current state of i^{th} team member guided by the supervisor, the objective function value is V_i^{S1}, while $z_{i,d}^{S1}$ represents the updated value for d^{th} variable in the considered problem, as proposed by i^{th} team member under the supervisor's direction. The update factor, along with the random numbers, I and r_i (between 0 and 1), plays an important role in the update process.

Stage 2: Sharing information

Team members are those whose performance is less effective compared to the supervisors. Each member of the team strives to better its performance by leveraging the knowledge of their teammates to perform better than them. Equations (3.17)-(3.19) are considered to update the team members.

$$Z^{M,i}: z_d^{M,i} = \frac{\sum_{j=1}^{N_i} z_{j,d}^{g,i}}{N_i} \tag{3.17}$$

$$Z_i^{S2}: z_{i,d}^{S2} = z_{i,d} + r_i \times \left(z_d^{M,i} - I \times z_{i,d}\right) \times sign\left(V_i - V^{M,i}\right) \tag{3.18}$$

$$Z_i = \begin{cases} Z_i^{S2}, & V_i^{S2} < V_i \\ Z_i, & \text{else} \end{cases} \tag{3.19}$$

where $Z^{M,i}$ is the arithmetic mean of the team member assumed to be better as compared to that of i^{th} member of the team, and $V^{M,i}$ denotes its objective function value. On the other hand, N_i represents the number of team members having superior performance than i^{th} member of the team, $z_{j,d}^{g,i}$ denotes the value of d^{th} variable identified by j^{th} better team member for i^{th} member of the team, Z_i^{S2} denotes the updated status for i^{th} member of the team and V_i^{S2} indicates the corresponding objective function value.

Stage 3: Individual activity of team member

Every team member within a team strives to improve his/her performance through individual effort, thereby resulting in collective success of the team. This phase of updating the team members is represented by Eqs. (3.20) and (3.21).

$$Z_i^{S3}: z_{i,d}^{S3} = z_{i,d} + (-0.01 + r_i \times 0.02) \times z_{i,d} \tag{3.20}$$

$$Z_i = \begin{cases} Z_i^{S3}, V_i^{S3} < V_i \\ Z_i, \quad \text{else} \end{cases} \tag{3.21}$$

where V_i^{S3}denotes the objective function value and $Z_i^{S3}$is the updated state of i^{th} team member.

Algorithm 3.1.3: Pseudo-code of TOA

```
Initialize the variables and fitness function of the problem
Set N and IT
Generate Z at random
Evaluate the fitness function
while the stopping criterion (e.g. maximum iteration number) is not fulfilled
 for t = 1 : IT
      Update S using Eq. (3.13)
        for i =1 : N
           Update Z_i (based on stage 1) applying Eqs. (3.14) and (3.16)
           Determine Z^{M,i} and N_i using Eq. (3.17)
Update Z_i (based on stage 2) employing Eqs. (3.18) and (3.19)
Update Z_i (based on stage 3) using Eqs. (3.20) and (3.21)
        end for
    end for
Return the best solution found
```

3.1.4 Human conception optimizer

HCO [197] is an innovative metaheuristic algorithm influenced by the biological process of human reproduction. It is framed on the principles of natural conception, where both exploration (diversity) and exploitation (fine-tuning) are the key elements for producing high quality solutions. This algorithm models conception, i.e. only healthful sperm can penetrate into the cervix due to selective function of cervical gel situated in the female reproductive system. Additionally, sperm are guided by the mucus gel to help them follow a path through the genital tract in the direction of the egg located in

the Fallopian tube. The asymmetrical motion of the sperm's flagellum enables them to navigate through the reproductive system, and the hyper-activation process equips the sperm with the ability to fertilize the egg. This algorithm has several advantages, like balance between exploration and exploitation, efficient convergence, and adaptability to various types of optimization challenges. While combining traits from the parents and introducing mutation, it can efficiently avoid premature convergence helping it to reach the optimal solutions [197].

Stage 1: Initialization

The positions of sperm cells represent their locations in the search space. Each sperm cell acts as a candidate solution for the given optimization problem. The initial positions of the sperm cells (x) are determined using Eq. (3.22).

$$x_i^j = x_{i,min}^j + r_i \times \left(x_{i,max}^j - x_{i,min}^j\right), \tag{3.22}$$

where i = 1,…,N (N denotes the number of sperm cells or search agents) and j = 1,…,d (d is the dimension of the problem). In the above equation, x_i^j is the initial position of i^{th} sperm cell, $x_{i,max}^j$ is the maximum bound and $x_{i,min}^j$ is the minimum bound for j^{th} decision variable of i^{th} sperm cell.

While evaluating a candidate solution x for a given problem, its opposite solution, X_{oppo}, may offer a better alternative. This opposite solution, X_{oppo}, is often nearer to the global optimum. The population of solutions in the opposite direction can be estimated using the following equation:

$$X_{oppo} = a + b - X \tag{3.23}$$

where a and b symbolize the lower limit and upper limit of the search agent, respectively.

Stage 2: Choice of a healthy population

During natural fertilization, healthy spermatozoa can only penetrate the cervix and fertilize a mature egg. Similarly, in this algorithm, a substantial starting population is created, from which the fittest individuals are chosen according to a probability function. The selected group of those individuals advances to the further stages of the algorithm. The ideal solution is referred to as the first optimal solution (fittest sperm cell), although the inferior solution is also recognized. The efficacy of the remaining solutions is assessed against that of the original optimal solution, with a probability, P_{fit}. Compute the probability of selecting the best candidates using Eq. (3.24) and select the healthy population based on Eq. (3.25).

$$P_{fit} = [f(x_{worst}) - f(x_{best})] \times w + f(x_{best}) \tag{3.24}$$

$$x_{healthy} = x \dots \dots when\ F(x) \leq P_{fit} \tag{3.25}$$

where w is the weight factor.

For an objective function, the fitness of the starting population depends on the type of the optimization problem. Additionally, using Eq. (3.26), determine the best fitness ($f(x_{best})$) and worst fitness ($f(x_{worst})$) from the fitness matrix.

$$F\left(x_{healthy}\right) = \begin{bmatrix} f(x_{healthy_1}) \\ \vdots \\ f(x_{healthy_n}) \end{bmatrix} = \begin{bmatrix} f(x^1_{healthy_1}, x^2_{healthy_2}, \ldots\ldots) \\ \vdots \\ f(x^1_{healthy_n}, x^2_{healthy_n}, \ldots\ldots) \end{bmatrix} \quad (3.26)$$

where the fitness matrix, $F\left(x_{healthy}\right)$ consists of the fitness values of all the healthy sperm cells and *n* represents size of the fit population.

Stage 3: Update sperm velocity

When one sperm cell in the healthy population achieves the best position among others in a given iteration, it is designed as the global best solution (S_{gbest}). The current iteration's position of a sperm cell would be compared to its prior position, and the best performing position would be nominated as the current best solution (S_{pbest}). The sperm cell moves on the curvature of sperm movement by travelling in a sinusoidal pattern, with its updated velocity. Equation (3.27) is considered to compute the velocity of i^{th} sperm cell, based on its fitness value, $f(x_i)$ and the velocity in the search space is updated applying Eq. (3.28).

$$v_i = \frac{\gamma(R^2 - r^2)}{4\eta L} \quad (3.27)$$

$$\vec{V}_i(j+1) = w \times \left(\vec{V}_i(j) + v_i(j)\right) + C_1 \times A_1 \times \sin\left(2\pi \frac{j}{j_{max}}\right) + C_2 \times A_2 \times \sin\left(2\pi \frac{j}{j_{max}}\right), \quad (3.28)$$

where $R = f(x_{best}) - f(x_i)$, v_i is velocity of i^{th} sperm cell, specific radius $(r) = f\left(x_{avg}\right) - f(x_i)$, length of pipe $(L) = f(x_{best}) - f\left(x_{avg}\right)$, η is dynamic viscosity, γ is a random number between 0 and 1, $f(x_{best})$ is the health of the best solution, $f\left(x_{avg}\right)$ is the arithmetic mean health of the population, $A_1 = (S_{pbest} - S_i)$, $A_2 = (S_{gbest} - S_i)$, and C_1 and C_2 are two constants. Equation (3.29) is applied to update position of the sperm.

$$\vec{x}_i(j+1) = \vec{x}_i(j) + \vec{V}_i(j+1) \quad (3.29)$$

where $\vec{x}_i(j)$ denotes the position and $\vec{V}_i(j)$ is the velocity of i^{th} sperm in j^{th} iteration.

Algorithm 3.1.4: Pseudo-code of HCO

```
Initialize population for each variable using Eq. (3.22)
Evaluate fitness f(x_i) of each sperm (x_i) for each variable
Calculate fitness f(x_oppo) for sperm x_oppo
    if f(x_i) > f(x_oppo)
    Select x_i
    else
    Select x_oppo
    end if
Find the best fitness f(x_best) and worst fitness f(x_worst) employing Eq. (3.26)
Derive P_fit using Eq. (3.24)
    if f(x_i) ≤ P_fit
    Update x_healthy based on Eq. (3.25)
else reject and inspect the next healthful sperm
    end if
Again initialize the healthy population of sperm
```

Identify $f(x_{avg})$ and $(f(x_{best}))$
Identify (S_{gbest}) attained at present iteration
Identify (S_{pbest}) at current iteration
Evaluate v_i using Eq. (3.27)
Update $\vec{V}_i(j+1)$ considering Eq. (3.28)
Update $\vec{x}_i(j+1)$ applying Eq. (3.29)
Return the best solution (fittest sperm)

3.1.5 Queuing search algorithm

QSA [198] is also a novel optimization technique inspired by the behavior of queues in real-life systems, such as waiting lines in supermarkets, traffic or network data packets. It mimics how elements (customers, vehicles or data) enter, wait, and move through queues based on certain priorities and service times. This algorithm utilizes the queuing model to solve complex optimization problems while organizing the potential solutions into a queue, processing them according to their fitness or quality, and updating them iteratively. QSA effectively prioritizes high quality solutions by compromising a balance between exploration and exploitation. Its adaptability and scalability make it acceptable for a variety of optimization challenges, from single-objective to multi-objective problems [198].

Business 1:

Business 1 model contains three queues, i.e. $queue_{11}$, $queue_{12}$ and $queue_{13}$, each managed by a dedicated staff member. The population is distributed among these three queues, where they are attended to by the respective staff. The state of each customer in the queue can be updated in two different ways, as outlined in Eqs. (3.30) and (3.31). These equations govern how the position or status of each customer is updated during his/her stay time in the queue, ensuring progress and optimization within the process.

Case = 1: $X_i^{new} = A + \left(\beta \times \alpha \times (E \circ |A - X_i|) + e \times (A - X_i)\right), \quad i = 1, \dots, N$ (3.30)

Case = 2: $X_i^{new} = X_i + (\beta \times \alpha \times (E \circ |A - X_i|), \quad i = 1, \dots, N$ (3.31)

where the update patterns are represented by case = 1 and case = 2. A equals to A_{1n} when $customer_i$ is in $queue_{1n}$ (where n = 1, 2, 3). On the other hand, X_i^{new} denotes the updated state of $customer_i$, α is a randomly generated number within [−1, 1], and E is a D^{th} dimensional vector consisting of D random numbers. The variable e is another random number created using the Erlang distribution. The symbol ∘ denotes the dot product, || denotes the absolute value, total number of customers is N for business 1, the state of $customer_i$ is denoted by vector X_i, and β is used to control the range of variation, which can be defined using Eq. (3.32).

$$\beta = e^{\left(In\frac{1}{t}\right)\times\left(\frac{t}{IT}\right)^{0.5}} \quad (3.32)$$

From the above equation, it becomes evident that when t increases, β would decrease.

Business 2:

This business also has three queues, i.e. $queue_{21}$, $queue_{22}$ and $queue_{23}$. A probability is allocated to each consumer using Eq. (3.33).

$$Pr_i = \frac{rank(f_i)}{N}, \quad i = 1, \dots, N \tag{3.33}$$

where $rank(f_i)$ is the ranking of f_i of customer$_i$.

A random number (r_i) is generated for each customer. If $r_i < Pr_i$, the state of customer$_i$ would be modified by this business. In the second business, there are also two methods for updating state of the customer, as outlined in Eqs. (3.34) and (3.35).

$$X_i^{new} = X_i + e \times (X_{r1} - X_{r2}), \quad i = 1, \dots, N_2 \tag{3.34}$$

$$X_i^{new} = X_i + e \times (A - X_{r1}), \quad i = 1, \dots, N_2 \tag{3.35}$$

where N_2 is the number of customers in business 2. A becomes equal to A_{2n} when customer$_i$ is in queue$_{2n}$ (where n = 1, 2, 3). X_{r1} and X_{r2} are two randomly selected customers. The confusion degree value, c_v, defined in Eq. (3.36), determines selection of these two methods' (X_{r1} and X_{r2}) update patterns.

$$c_v = \frac{T_{21}}{T_{22}+T_{23}} \tag{3.36}$$

where T_{2n} is the potential service time for the best three customers obtained in business 2 with respect to queue$_{21}$, queue$_{22}$ and queue$_{23}$.

A r_i is generated for each customer. If $r_i < c_v$, Eq. (3.34) is employed to adjust the customer's status. Otherwise, Eq. (3.35) is implemented.

Business 3:

There is a single queue in this business. The customer's state is changed based on multiple dimensions. A r_i is generated for d^{th} dimension of customer$_i$. If $r_i > Pr_i$, d^{th} dimension of customer$_i$'s status would be updated utilizing Eq. (3.37).

$$X_i^{new} = X_{r1,d} + e \times \left(X_{r2,d} - X_{i,d}\right), \quad i = 1, \dots\dots, N_3, d = 1, \dots, D \tag{3.37}$$

where N_3 is the number of customers in business 3.

Algorithm 3.1.5: Pseudo-code of QSA

```
Initialize N, IT and D
Evaluate f(X_i)
While the stopping criterion is not fulfilled
    for i = 1 : N
Business 1:
        if i ≤ queue11
            if i = 1; case = 1; end if
            A = A11
        else if i > queue11 and i ≤ queue11 + queue12
            if i = queue11 + 1; case = 1; end if
            A = A12
        else
            if i = queue11+ queue12 + 1; case = 1; end if
           A = A13
        end if
        revise β
        if case = 1
           obtain X_i^new using Eq. (3.30)
```

if X_i^{new} has better fitness values, replace X_i with X_i^{new}
case = 1;
else
case = 2;
end if
else
derive X_i^{new} applying Eq. (3.31)
if X_i^{new} has better fitness values, replace X_i with X_i^{new}
case = 2;
else
case = 1;
end if
end if
Business 2:
if $i \leq \text{queue}_{21}$
$A = A_{21}$
else if $i > \text{queue}_{21}$ and $i \leq \text{queue}_{21} + \text{queue}_{22}$
$A = A_{22}$
else
$A = A_{23}$
end if
if $r_i < Pr_i$
select X_{r1} and X_{r2}
if $r_i < c_v$
obtain X_i^{new} employing Eq. (3.34)
else
obtain X_i^{new} using Eq. (3.35)
end if
accept X_i^{new} if it has superior fitness values
end if
Business 3:
for $d = 1 : D$
if $r_i > Pr_i$
select X_{r1} and X_{r2}
obtain X_i^{new} using Eq. (3.37)
end if
end for
accept X_i^{new} if it has better fitness values
end for
end while
Return the better fitness values (X_i^{new})

3.2 Mating behavior-based metaheuristics

The mating process of insects or animals is very similar to an optimization process, wherein each species attempts to produce a brood with superior genes (a perfect state), as the species with superior genes has a higher chance of survival. Similarly, an optimization procedure looks for a global solution

or ideal state, where each solution's quality is evaluated based on an objective function (fitness) as the criterion. In engineering optimization, a solution vector is usually developed by assigning values to the decision variables in the search space [199, 200]. Mating behavior-based metaheuristic algorithms have several advantages, like they allow generation of new offspring and effective exploration of the solution space, having excellent balance between exploration and exploitation strategies while outperforming other optimization techniques with respect to solution accuracy and reliability. The mating properties, which include minimum control parameters, ease of implementation, efficacy in solving real-time challenges and simplicity in construction, are also responsible for their excellent performance.

Table 3.3 provides values of all the tuning parameters considered for the five mating behavior-based algorithms. Their performance greatly depends on proper selection of those parameters, influencing search behavior, convergence speed and solution quality. To derive the best results and increase efficiency of the considered algorithms, values of the corresponding tuning parameters are set based on trial and error method, and considering the recommendations of the past researchers.

Table 3.3 Values of different tuning parameters for the considered algorithms

Algorithm	Parameter	Value
FMA	γ	0.2
	Crossover rate	1.0
	Mutation rate	0.2
NMR	bp	0.5
	λ	U(0,1)
BMA	B_s	0.02
	b_1	0 to 1
	b_2	3
BMOA	Pl	7
AZOA	pc	0.1
	P	0.5
	Crossover	Mean

3.2.1 Firefly mating algorithm

The primary characteristic of FMA, proposed by Ritthipakdee et al. [201], is based on a unique technique for selecting mating pairs, drawing inspiration from two natural firefly's mating behavior. Fireflies of both sexes are of multiple-mating type, meaning that they mate with several opposite-sex partners. Males and females mate because of their mutual attraction [201]. The original FFA is an efficient metaheuristic algorithm employed to solve many real-world optimization problems, such as scheduling, economic load dispatch etc. [202]. On the other hand, FMA which uses GA as the core of the algorithm, can deal with multimodality and automated subdivision, based on attraction, which diminishes with distance. Furthermore, the parameters of FMA can be adjusted to manage unpredictability as iterations go on, allowing faster convergence, making it adaptable to solve continuous, grouping and categorization, and combinatorial optimization problems [203]. The performance of this algorithm has already been tested with various benchmark functions, outperforming

other popular metaheuristics, like FFA, cognitive behavior optimizer, DE, lighting search algorithm, multithreaded parallel dual population GA, mixed swarm-based cooperative PSO etc. It is influenced by the flashing behavior of fireflies to attract another firefly and has the following three main stages [201]:

a) The male firefly chooses the female depending on the pheromone released by the female, according to wind speed and direction.
b) A female uses the flash's brilliance to choose a guy for mating.
c) Until the male runs out of sperm or the female's spermatheca is full, he or she mates repeatedly, creating more capable offspring for the next generation

The FMA proceeds following the six subsequent phases, as mentioned below:

Phase 1: Initialization

Generate an initial population of N fireflies, and allocate half to males and the other half to females randomly. There are d genes in every firefly, which is same as the number of variables in the optimization problem. In addition, real numbers between 0 and 1 representing sizes of each male's sperm reservoir and each female's spermathecal are also generated at random. Finally, each firefly's fitness value is evaluated.

Phase 2: Mating pair's selection

This step suggests the logic behind FMA for identifying the mating partners. Initially, a female firefly's position (based on her pheromone trail) attracts males; and then, the males are chosen by the female according to their brightness. Figure 3.1 illustrates that all female fireflies can mate with each male firefly, and all male fireflies can mate with all other female fireflies. By incorporating this idea into FMA, it becomes possible to avoid the problem of the roulette wheel selection strategy of being frequently trapped into the local optima. The next phase explains the new selection process in detail.

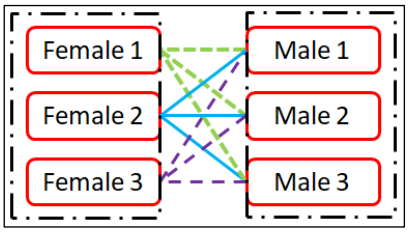


Figure 3.1 Pairing strategy of fireflies during the mating process

Phase 3: Female pheromone calculation

The male firefly determines the degree of concentration at which each female's pheromone reaches him. The pheromone concentration level (P_{ji}) that the female i reaches the male j, is determined using the following equation:

$$P_{ji} = f_i \times \left(\vec{D} \cdot \vec{W}\right) \tag{3.38}$$

where f_i is the fitness value of a female firefly i, W is the wind vector randomly generated in each iteration, and D is the difference vector between the position vector of a male and that of a female.

Phase 4: Male firefly's brightness calculation

The female fireflies assess the attractiveness of every male within their range. Like FFA, a male firefly's attractiveness is closely correlated with his brightness.

$$A_{ij} = f_j e^{-\gamma r^2} \tag{3.39}$$

where A_{ij} is the appeal of the male firefly j sensed by the female firefly i, γ is the light absorption coefficient in the range [0,1], r is the Euclidean distance between the male and female fireflies, and f_j is the fitness value of the male firefly j.

Phase 5: Mutual attraction

A variety of males and females are matched as possible parents based on how much they both find to be attractive. The initial mating couple is chosen considering maximum mutual attraction among all the mating pairs. The term MA_{ij}, describing the mutual attraction between the male firefly j and the female i, can be defined using the following expression:

$$MA_{ij} = A_{ij} + P_{ji} \tag{3.40}$$

The quantity of sperm in the female's spermatheca and the male's sperm storage are updated after each mating.

Update procedure for male's sperm reservoir and female's spermatheca.

The size of each female's spermatheca and quantity of sperm in each male's sperm reservoir are first randomly initialized within the range [0, 1]. The quantity of sperm in a male's sperm reservoir decreases when a pair of male and female mates because the sperm is moved from the male's sperm reservoir to the female's spermatheca. The amount of sperm a male offers to the female he selects for mating depends on how fit she is. The following equation determines how many sperm a male can offer to the female of his choice:

$$n_{ij} = \delta_j \times f_i \tag{3.41}$$

where η_{ij} is the number of sperms transferred to the female at the time of mating and δ_j is the number of sperms in a male's sperm reservoir.

A male's sperm reservoir and a female's spermatheca contain different numbers of sperm, which may be updated using Eqs. (3.42) and (3.43).

$$\delta_j^{new} = \delta_j^{old} - n_{ij} \tag{3.42}$$

$$\omega_i^{new} = \omega_i^{old} + n_{ij} \tag{3.43}$$

where ω_i is the number of sperms in a female's spermatheca. Until no suitable firefly remains to form a mating pair, mating continues.

Phase 6: Crossover operation

In this stage, certain genes are crossed over to create two new offspring when a male and a female mate. A particular crossover procedure is adopted based on the following two cases to create offspring.

Case 1: 2-point crossover operator if the parents have never been mated.

Case 2: *n*-point crossover operator if either parent has previously been mated.

Mutation operation and selecting the next generation

Once an incipient offspring is generated through mating, a portion of its genes is randomly altered (mutated) to new values within the range of the variables. This process is carried out to encourage population diversity and prevent the offspring from being stuck into the local optima.

Once all the offspring have undergone mutation, the best fireflies from the combined population of the parents and offspring are chosen to replace the old population of parents, i.e. only the more productive fireflies are chosen to be the population of the next generation. This is accomplished by first classifying the members according to their fitness values, and choosing only those individuals with higher fitness values to constitute the next population.

Termination.

The present iteration terminates after choosing a new population for the next generation. Once the maximum number of iterations is achieved or the best solution is not improved after a certain number of iterations, the algorithm stops.

Algorithm 3.2.1: Pseudo-code of FMA [204]

```
Initialize population of fireflies
    Calculate fitness of the fireflies and save the fitness in Light;
Define coefficient γ
while (t < IT)
for any two fireflies (X_i, X_j)
    if Light(X_i) less than Light(X_j)
    X_i fly towards X_j according to Eq. (3.39)
        Calculate fitness of firefly X_i and update Light(X_i);
    end if
end for
Mating phase
for any firefly
    Select two fireflies randomly from the parent fireflies
    Generate a child firefly according to Eq. (3.40)
    Apply greedy selection
end for
Memorize the best result found so far
end while
Until termination condition
```

3.2.2 Naked mole-rat algorithm

The mating patterns of NMRs present in nature are simulated in NMR algorithm [205]. NMRs are eusocial creatures residing in a group of around 295 individuals, with an average of 70-80 members. A female queen serves as the group's leader and divides the population into two different groups, i.e. workers and breeders. Workers do the required jobs, and the best are replaced by the breeders; while the breeders compete among themselves to mate with the queen. The queen picks the best breeder in the pool for mating, while the breeders are exclusively intended for mating only. The most physically

fit worker becomes a new breeder, while those who become infertile are sent to the worker's group. This phenomenon is modified to develop the NMR algorithm [205]. For both unimodal and multimodal functions, it exhibits strong exploitation and exploration, respectively. For majority of the benchmark functions, it has performed best in terms of convergence speed and quality of solution. Salgotra and Singh [205] proved its computational superiority over many other state-of-the-art algorithms for 27 well-known test functions. It can also prevent local minima and achieve a higher degree of consistency while locating the global optima [206]. This algorithm is divided into the following three phases:

Phase 1: Initialization

It creates a uniformly distributed random population of n NMRs, where each NMR within the range [1, 2,…,n] is a D-dimensional vector. Here, D denotes the number of variables in the optimization problem. Each NMR is initialized based on the following equation:

$$NMR_{i,j} = NMR_{\min,j} + U(0,1) \times (NMR_{\min,j} - NMR_{\max,j}) \tag{3.44}$$

where $i \in [1,2,...,n]$, $j \in [1,2,...,D]$, $NMR_{i,j}$ is i^{th} solution in j^{th} dimension, $NMR_{\min,j}$ and $NMR_{\max,j}$ are the lower and upper bounds of the problem function respectively, and $U(0, 1)$ is a uniformly distributed random number.

After initialization, based on the fitness, B breeders and W workers are identified, and the overall initial optimal solution (d) is determined. The worker and breeder phase search procedure is repeated on the NMR population through cycles or iterations.

Phase 2: Worker phase

In this phase, the workers often attempt to become fitter so they may eventually mate with the queen and become breeders. Thus, when the new NMR's fitness is assessed; if it proves to have a better mating fitness, the old solution is discarded and the new one is stored into the memory. If not, the previous solution is retained. The ultimate fitness of each worker is retained once they have all finished their search process. The NMR employs Eq. (3.45) to create a new solution from the old one.

$$w_i^{t+1} = w_i^t + \lambda\left(w_j^t - w_k^t\right) \tag{3.45}$$

where w_i^{t+1} is the new solution or worker, w_i^t denotes i^{th} worker in t^{th} iteration, λ is the mating factor, whose value is derived from a uniform distribution in the range of [0, 1], and w_j^t and w_k^t are the two random solutions selected from the worker's pool.

Phase 3: Breeder phase

To be chosen for mating and continue as a breeder, breeders must also keep themselves updated. Based on a breeding probability (bp) relative to the overall best d, the breeder NMRs are updated. The value of bp is arbitrary, ranging between 0 and 1. The following equation is adopted by the breeders to adjust their positions:

$$b_i^{t+1} = (1-\lambda) b_i^t + \lambda\left(d - b_i^t\right) \tag{3.46}$$

where b_i^{t+1} is the new solution of i^{th} breeder and b_i^t corresponds to the position of i^{th} breeder in t^{th}

iteration.

Algorithm 3.2.2: Pseudo-code of NMR algorithm

Input: Initialize naked mole rats: n
Breeders B: $n/5$
Workers W: $(n - B)$
Define bp
Define D-dimensional objective function, $f(x)$
Output: find the overall best d
do until $t < IT$
for $i = 1: W$
Perform worker phase using Eq. (3.45)
evaluate w_i^{t+1}
end for
for $i = 1: B$
if $U(0,1) > bp$
Perform breeder phase applying Eq. (3.46)
evaluate b_i^{t+1}
end for
combine the new worker and breeder populations
evaluate the population
update the overall best d
update iteration count
end until
save the final best (d)

3.2.3 Butterfly mating optimization algorithm

The BOA was proposed by Jada et al. [207]. The primary means of reproduction for butterflies is ultraviolet (UV) reflectance by males and absorbance by females. This mechanism operates under the assumption that there is no distinction between males and females, and each butterfly simultaneously absorbs and reflects UV radiation it receives from all other butterflies. It has been proven to be quite efficient in capturing multiple peaks while solving multimodal optimization problems. Its solution efficacy has also been tested on various benchmark functions, providing satisfactory solutions and identifying the local peaks [207]. In this algorithm, each butterfly selects its partner adaptively, anywhere in the search space, for each iteration; which is referred to as a local mate or *l*-mate. As compared to PSO and GA, BOA has faster convergence rate which may be attributed to many properties, like few control parameters, simplicity in construction, ease of implementation etc. [208-209]. This algorithm is described as below:

Phase 1: UV update

In this phase, the UV of each butterfly is updated in proportion to its fitness, i.e. function value at the current location of the butterfly, as given by Eq. (3.47)

$$UV_i(t_m) = \max\{0, b_1 \times UV_i(t_m - 1) + b_2 \times f(t_m)\} \quad (3.47)$$

In the above equation, UV is updated at time index 't_m' for i^{th} butterfly, providing more importance to

the current fitness and less to the previous *UV*. Values of b_1 and b_2 are accordingly chosen so that $0 \leq b_1 \leq 1$ and $b_2 > 1$.

Phase 2: UV distribution

Here, each butterfly distributes its *UV* to other butterflies such that the nearest butterfly receives more share than the farthest one. Distribution of *UV* to other butterflies follows a particular strategy. The *UV* value of i^{th} butterfly with UV_i is reflected to j^{th} butterfly at a distance d_{ij}, which is measured using Eq. (3.48).

$$UV_{i\to j} = UV_i \times \frac{d_{ij}^{-1}}{\sum_k d_{ik}^{-1}} \tag{3.48}$$

where N is the number of butterflies ($i = 1, 2,...,N$; $j = 1, 2,...,N$; $k = 1, 2,...,N$; $j \neq i$; $k \neq i$), $UV_{i\to j}$ is the *UV* absorbed by j^{th} butterfly from i^{th} butterfly, d_{ij} is the Euclidean distance between i^{th} and j^{th} butterflies; and d_{ik} is the Euclidean distance between i^{th} and k^{th} butterflies.

Phase 3: *l*-mate selection

When a butterfly selects a partner with the maximum *UV* and proceeds towards it, all of the butterflies would eventually converge to a single peak. If the *l*-mate selection process becomes more adaptive, possibility for formation of separate groups of butterflies would arise. Thus, BOA employs *l*-mate selection strategy mentioned as follows:

The i^{th} butterfly initially puts all other butterflies in descending order according to the quantity of *UV* it has obtained from them. Now, this may result in some sort of localization and peak sensing, if each butterfly selects the first butterfly in its descending rank as its *l*-mate and moves towards that. However, to record local peaks concurrently, the i^{th} butterfly must also take into account the *UV* of the remaining butterflies. For this, it must sequentially compare UV_i with *UV* values of other butterflies arranged in descending order. Based on this comparison, it selects the first butterfly that satisfies the following condition as its *l*-mate (receiving more UV than it has).

$$UV(i^{th}\ Butterfly < j^{th}\ Butterfly) \tag{3.49}$$

where $i = 1, 2,...,N$; $j = 1, 2,...,N-1$; and j is an index of butterflies in the descending order of i^{th} butterfly.

Phase 4: Movement

In this phase, each butterfly moves in the direction of its *l*-mate based on the following equation:

$$x_i(t_m + 1) = x_i(t_m) + B_s \times \left\{\frac{x_{l-mate}(t_m) - x_i(t_m)}{\|x_{l-mate}(t_m) - x_i(t_m)\|}\right\} \tag{3.50}$$

where $x_i(t_m)$ is the position of i^{th} butterfly in time index 't_m' and B_s is the butterfly step size.

Algorithm 3.2.3: Pseudo-code of BOA
Randomly initialize butterflies
$\forall i$, set $UV_i = UV(0)$
Set *IT*
Set $t = 1$
while ($t \leq IT$) do

```
{
    for each butterfly i do
        UV update; % using Eq. (3.47)
        UV distribution; % using Eq. (3.48)
    for each butterfly i do
        Select l-mate; % using l-mate selection phase
        Update position; % using Eq. (3.50)
t = t + 1
}
```

3.2.4 Barnacle mating optimization algorithm

The primary source of inspiration for BMOA, proposed by Sulaiman et al. [210], is originated from the mating behavior of barnacles in nature. Barnacles, which are hermaphrodite micro-organisms, are present in or very near to seawater, and have both male and female reproductions. For creating new offspring, they need to be fertilized by a neighbor. They have their long penises, almost seven times longer than their bodies, to adapt to the shifting tides and their sedentary lifestyle. BMOA can escape the local optima, making it able to derive the global solution having benefits of few control parameters, capacity of utilizing the search space and discovering new promising areas [211-213]. Its optimization performance has also been tested on various benchmark multi-dimensional functions, identifying both the local and global minimal solutions. In BMOA, exploitation and exploration processes required for generating new offspring are inspired by the Hardy-Weinberg principle and sperm cast situation, respectively.

Hardy-Weinberg principle for offspring generation

According to this principle, in absence of external evolutionary factors, genotype and allele frequencies in a population would not change from generation to generation. In the simplest situation represented by two alleles of D and M ($f(D) = p$ and $f(M) = q$) (D and M respectively denote *Dad* and *Mum*), the expected genotype frequencies can be expressed as $f(DD) = p^2$ for homozygotes of type DD, $f(MM) = q^2$ for homozygotes of type MM, and $f(DM) = 2pq$ for heterozygotes. The sum of the entries is $p^2 + 2pq + q^2 = 1$, making it evident that $p + q = 1$. Thus, the p and q of the barnacles' parents determine generation of the new offspring.

Initialization

In this algorithm, it is assumed that the candidate solutions in the barnacles' population can be represented using the following vector:

$$X = \begin{bmatrix} X_1^1 & \dots & X_1^N \\ \dots & \dots & \dots \\ X_n^1 & \dots & X_n^N \end{bmatrix} \tag{3.51}$$

where N denotes the number of control variables and n is the population (number) of barnacles. Evaluation of the vector X is initially executed, and the sorting process is performed to identify the best solution so far at the top of vector X.

Selection

The selection process is modelled based on the behavior of barnacles, dependent on certain assumptions. Specifically, each barnacle may only be fertilized by one barnacle at a time, and selection is done randomly, but is limited to the length of the barnacles' penises, represented by *pl*. Sperm cast mating occurs if the selection at a certain iteration is more than the predetermined *pl*. The following selection rule is adopted here:

$$barnacle\_d = randperm(n) \tag{3.52}$$

$$barnacle\_m = randperm(n) \tag{3.53}$$

where *barnacle_d* and *barnacle_m* are the parents to be mated.

Offspring generation

The Hardy-Weinberg principle, which has been discussed earlier, serves as the model for new offspring generation in BMOA. The new offspring of barnacle's parents are produced using the following expressions:

$$x_i^{N\_new} = p \times x_{barnacle\_d}^{N} + q \times x_{barnacle\_m}^{N} \text{ for } k \le pl \tag{3.54}$$

$$x_i^{N\_new} = r_i() \times x_{barnacle\_m}^{N} \quad for\ k > pl \tag{3.55}$$

where $k = |barnacle\_d - barnacle\_m|$, p is the normally distributed pseudo-random number, $q = (1 - p)$, $x_{barnacle\_d}^{N}$ and $x_{barnacle\_m}^{N}$ are the variables of *Dad* and *Mum* of barnacles, respectively, selected using Eqs. (3.52) and (3.53), and $r_i()$ is a random number between [0, 1]. It can be said that p and q respectively represent the percentage of characteristics of *Dad* and *Mum*, embedded in the generation of new offspring. Thus, the offspring inherits behaviors of *Dad* and *Mum* based on probability of the random number between 0 to 1.

Algorithm 3.2.4: Pseudo-code of BMOA

```
Initialize population of barnacles X_i
Estimate fitness of each barnacle
Sort to locate the best result at the top of the population (S = best solution)
while (t < IT)
Set the value of pl
Selection using Eqs. (3.52) and (3.53)
Barnacle_d = randperm(n)
Barnacle_m = randperm(n)
                    if the selection of dad and mum ≤ pl
        for each variable
        Generate off-spring applying Eq. (3.54)
        end for
else if selection of dad and mum > pl
        for each variable
        Generate off-spring employing Eq. (3.55)
        end for
end if
Bring the current barnacle back if it goes outside the boundaries
```

Calculate the fitness of each barnacle
Sort and update S if there exists a better solution
$t = t + 1$
end while
Return S

3.2.5 American zebra optimization algorithm

Development of this algorithm by Mohapatra and Mohapatra [214] is based on the American zebras which are unique among animals because of their intriguing social nature and leadership style, forcing young zebras to leave their herd before their adulthood and joining a different herd unrelated to them. The efficiency of AZOA has already been tested for numerous benchmark functions and compared against many of the state-of-the-art metaheuristics while solving practical engineering problems, exhibiting excellent balance between exploration and exploitation, robustness, and ability to achieve global optimal solutions with ease and resilience. In this algorithm, the zebra baby's departure from its own group promotes diversity by impeding intra-family mating.

Phase 1: Zebra group's formation

In the wild, zebras follow a stallion that serves as their leader, dividing the entire population into multiple groups that appear to be distinct from one another. The total number of groups (N) is determined using the formula $N = S \times Ps$, where Ps denotes the stallion probability in the entire population (S). The formula $Z_{i,j} = (Z_{\max} - Z_{\min})r_i + Z_{\min}$ is considered to derive location of i^{th} zebra in j^{th} group ($Z_{i,j\in N} = \{ Z_{ij1}, Z_{ij2}, Z_{ij3},...,Z_{ijn}\}$) in n-dimensional search space, where $Z_{\max}$ and $Z_{\min}$, respectively, determine the upper and lower extremities of the search region. The random value between [0, 1] is represented by the symbol 'r_i'. This guarantees that there are N distinct zebra crowds, each with a unique stallion.

Phase 2: Feeding activity

The following equations are adopted to simulate the feeding behavior of the American zebras.

$$\overline{Z}_i^j = \begin{cases} 2R_1 \sin(2\pi R_2) \times (Z_S^j - Z_i^j) + Z_S^j, & \text{if } R_3 < 0.5; \\ 2R_1 \cos(2\pi R_2) \times (Z_S^j - Z_i^j) + Z_S^j, & \text{otherwise} \end{cases} \quad \forall i \in N_j \tag{3.56}$$

$$Z_i^j = \begin{cases} \overline{Z}_i^j, & \overline{F}_i^j < F_i^j \\ Z_i^j, \text{otherwise} \end{cases} \tag{3.57}$$

where Z_S^j and Z_i^j denote the position of the stallion and i^{th} zebra of j^{th} group, respectively, N_j is the total members in j^{th} group, R_1 specifies a uniform random value between [−2, 2], R_2 is the adaptive parameter which is evaluated based on Eq. (3.58), R_3 is a random value ranging in [0, 1], sin and cos functions help movement of other i^{th} members in multiple angles around the leader of the family, $\overline{Z}_i^j$ denotes the new update of i^{th} member position during feeding, and $\overline{F}_i^j$ is the fitness value of i^{th} zebra.

$$R_2 = 1 - t \times \left(\frac{1}{IT} \right) \tag{3.58}$$

Phase 3: Breeding activity

The following equations are considered to model the zebras' breeding activity.

$$Z_j^q = Crossover\left(Z_i^a, Z_j^b\right) \text{ if } r < pc, i \neq j \tag{3.59}$$

$$Z_k^q = Crossover\left(Z_i^a, Z_k^c\right) \text{ if } r \geq pc, i \neq k, \quad \forall i, j, k \in N \tag{3.60}$$

where Z_j^q and Z_k^q are the positions of zebra q in j^{th} group and k^{th} group, respectively; Z_i^a is the position of the baby zebra a from i^{th} group; Z_j^b is the position of zebra b from j^{th} group; Z_k^c is the position of the zebra c from k^{th} group; pc is the probability of crossover and r is the crossover ratio.

Phase 4: Group leadership

The zebra group which is more powerful than the other group is still in control of the grasslands and water reservoir. Subsequently, it becomes accessible to others. The following equations are utilized to model this strategy.

$$\overline{Z}_S^j = \begin{cases} 2R_4 \sin(2\pi R_5) \times (WR - Z_S^j) + WR, & \text{if } R_6 < 0.5; \\ 2R_4 \cos(2\pi R_5) \times (WR - Z_S^j) + WR, & \text{otherwise} \end{cases} \tag{3.61}$$

$$Z_S^j = \begin{cases} \overline{Z}_S^j, & \overline{F}_S^j < F_i^j \\ Z_S^j, \text{otherwise} \end{cases} \tag{3.62}$$

where R_4 is a uniform random number in the range of [−2, 2], R_5 is the adaptive parameter which is determined using Eq. (3.63), R_6 is a uniform random number in the range of [0, 1], WR denotes the water reserves, Z_S^j is j^{th} group leader stallion at current position, $\overline{Z}_S^j$ is j^{th} group leader stallion at the next position, and $\overline{F}_S^j$ is the fitness value of stallion in j^{th} group.

$$R_5 = 1 - t \times \left(\frac{1}{IT}\right) \tag{3.63}$$

Phase 5: Leadership transition

In this stage, a strategy is adopted to replace the group leader if it exhibits signs of weakness. To choose a new leader, the leadership transition stage is modelled as follows:

$$Z_S^j = Z_i^j \text{ if } F(Z_i^j) < F(Z_S^j), \forall i \in N_j \tag{3.64}$$

where $F(Z_S^j)$ is the fitness value of the leader stallion and $F(Z_i^j)$ is the fitness value of i^{th} zebra.

Algorithm 3.2.5: Pseudo-code of AZOA

```
Input: AZOA parameters: pc and Ps
Output: Optimal solution obtained by AZOA
Set N and IT
Initialize the American zebra population randomly
Determine fitness values of American zebras
Determine stallion and foal number by Stallion = Ps×N and foal = N − stallion, respectively
While termination criterion is not satisfied
    for j = 1 to number of stallions
```

for i = 1 to number of foals
Phase 1: Feeding activity
Calculate the new position of i^{th} foal using Eq. (3.56)
Update the position of i^{th} foal using Eq. (3.57)
Phase 2: Breeding activity
if *rand* > *pc*
Update position of i^{th} foal using Eq. (3.59)
else
Update position of i^{th} foal using Eq. (3.60)
end
end
Phase 3: Group leadership
if *rand* > 0.5
Calculate new position of $\overline{Z}_S^j$ using Eq. (3.61)
Update position of $\overline{Z}_S^j$ using Eq. (3.62)
end
Phase 4: Leadership transition stage of selecting a leader
if $F(Z_i^j) < F(Z_S^j)$
$Z_S^j = Z_i^j$
end
end
end while
Set the best stallion as the optimal solution

3.3 Preying behavior-based metaheuristics

Finding out wild food resources is known as foraging. It includes all the methods related to location and consumption of food sources, their retrieval and storage, within a large community. On the other hand, preying is associated with chasing, capturing and killing of animals for food. Prey is a term used to describe an animal that is hunted and killed by a predator for sustenance. Predators may be solitary or group hunters. During preying, predators in group require less effort to kill the prey than in solitary modes. They hunt with little or no coordination in the simple form of group hunting, while in the complex form, they adopt different roles to herd and hunt prey. Preying behavior-based metaheuristic algorithms have several advantages, like (a) the new generation is not affected by a member with the worst performance, (b) unlike GA and PSO, a solution member always follows the best solutions, and (c) other algorithms perform the update using a memory of the best solutions, but updating in these algorithms does not involve any memory of the previous iterations and it solely depends on the current solutions [215, 216].

In Table 3.4, values of different algorithm-specific parameters for each of the metaheuristics are provided. These values of various algorithm-specific parameters are chosen based on the recommendations of the past researchers who mainly relied on trial and error method, for having the optimal performance of the preying behavior-based metaheuristics under consideration.

Table 3.4 Values of different algorithm-specific parameters

Algorithm	Parameter	Value
SFO	P_P	0.1
	A	4
	E	0.0001
AO	δ	0.1
	α	0.1
HHO	Dimension	4
BA	$f_{\min}$	0
	$f_{\max}$	100
GWO	Dimension	4

3.3.1 Sailfish optimizer

SFO, proposed by Shadravan et al. [217], is one of the preying behavior-based algorithms, inspired by the attack alternation strategy adopted by the hunting sailfish in pursuit of sardines. The sailfish is classified as a social predator due to its behavior to work in a group to catch and hunt its prey. In cooperative hunting, predators use different strategies to kill the prey. For example, a group of sailfishes is characterized by the alternation of attacking strategies. At a given moment, each member of the group attacks alone the school of prey (sardines) and injures or hunts some of them, while other members of the group save their power. When a sailfish attacks the prey school, it can update its position around them. In addition, the sailfish can also update its position to occupy empty space around the prey school and simulate encircling the prey. On the other hand, in order to escape from the attack of the sailfishes, the prey group (sardines) changes their positions when a member of their group is injured. The advantages of SFO are its strong search ability, easy implementation and good robustness, although there is a chance of diversity loss of the populations of sailfish and sardine late in the iteration process [218, 219]. It has a low balance of exploitation due to lack of selection in the local search process. The general procedure of SFO is presented as below [220, 221]:

a) Predator attacking strategy

Randomly define the population of predators (sailfishes) and preys (sardines). Assign the corresponding position to each predator and prey as X^t_{Sf} and Y^t_{Sd} respectively, where $Sf \in$ {sailfishes}, $Sd \in$ {sardines} and t is the current number of iterations. Mathematically, the new position of the predator (sailfish) can be defined using the following equation:

$$\mathrm{X}^t_{new\_Sf} = \mathrm{X}^t_{elite\_sf} - \lambda_t \times \left(r_i \times \left(\frac{X^t_{elite\_Sf} + Y^t_{injured\_Sd}}{2} \right) - X^t_{old\_Sf} \right) \tag{3.65}$$

where $X^t_{elite\_Sf} \in$ {position of the elite predator} made until now, $Y^t_{injured\_Sd} \in$ {best position of the injured prey} made so far, $X^t_{old\_Sf} \in$ {current position of the predator}, and λ_t is a coefficient at t^{th} iteration which can be defined using Eq. (3.66):

$$\lambda_t = 2 \times rand \times Dp - Dp \tag{3.66}$$

where Dp is the density of prey (sardines) showing the number of prey at each iteration. The following equation can be utilized to estimate the density of prey parameter:

$$Dp = 1 - \left(\frac{N_{Sf}}{N_{Sf} + N_{Sd}} \right) \tag{3.67}$$

where $N_{Sf} \in$ {number of predators} and $N_{Sd} \in$ {number of prey} in each cycle of the optimizer. As the quantity of prey is assumed to be initially higher than that of the predator, N_{Sf} is determined using the following expression:

$$N_{Sf} = N_{Sd} \times P_P \tag{3.68}$$

where P_P is the percentage of prey population that forms the initial predator population.

b) Prey catching strategy

At the start of the hunting process, the catching power of predator and escape capability of the prey are initially very high. Therefore, at the time of hunt, the predator just injures the prey without being able to catch them. Steadily, the attack power of predators decreases due to strong and repeated attacks, and the escape ability of the prey would also subsequently reduce. Hence, the catching success rate of the predators becomes high. Considering the action of prey against the attacks of predator, each prey would change its position based on the following equation:

$$Y^t_{new\_Sd} = r_i \times (X^t_{elite\_Sf} - Y^t_{old\_Sd} + AP) \tag{3.69}$$

where $Y^t_{old\_Sd} \in$ {current position of the prey} and $AP \in$ {amount of the predator's attack power at each iteration}.

In the concluding stage of the hunting process, it is assumed that catching of prey would occur when the prey becomes healthy than its corresponding predator. In this situation, position of the predator is replaced with the current position of the hunted prey to increase the chance of catching new prey. The following equation expresses the above condition:

$$X^t_{Sf} = Y^t_{sd}, \text{ if } f(Sd_t) < f(Sf_t) \tag{3.70}$$

where $X^t_{Sf} \in$ {current position of the predator at t^{th} iteration} and $Y^t_{sd} \in$ {current position of the prey at t^{th} iteration}.

Algorithm 3.3.1: Pseudo-code of SFO algorithm

```
Initialize population of sailfishes as N and sardines as N_S
Initialize parameters (A = 4, ε = 0.0001, P_P = 0.1)
Compare the fitness of sailfish and sardine
while the termination condition is not satisfied
    for each sailfish
        Update the position of sailfish using Eq. (3.65)
    end for
    Calculate attack power as {A× (1 − (2 ×t× ε))}(A is the coefficient for decreasing value
    of the power attack)
        if attack power (AP) < 0.5
            Calculate α as {N_Sd× AP}
            Calculate β as {d_t× AP}(d_t is the number of variables at t^th iteration)
            Select a set of sardines based on the values of α and β
            Update the positions of all sardines by Eq. (3.69).
```

```
            end if
        Calculate the fitness of all sardines
            if there is a better solution for sardine population
                Replace a sailfish with an injured sardine using Eq. (3.70).
                Replace the hunted sardine from population
                Update the best sailfish and best sardine
            end if
        end while
Return best sailfish
```

3.3.2 Harris Hawks optimizer

HHO is a newly proposed preying behavior-based optimization algorithm [222]. The foraging and hunting behavior of Harris Hawks significantly differs from other birds, as they continue to forage with other family members of the same species. They adopt a collaborative action and hunting style called the 'surprise pounce' (also known as 'seven kills' approach) to ambush the prey. During this attack, a few other hawks use to ambush in a number of directions and converge on the target rabbit, and the attack would be over in a matter of seconds. They employ different hunting styles based on the escape behavior of the prey and dynamic change in instances. For example, the Harris Hawks use switching tactics when the leader hawk dives quickly to attack the prey, and the prey is trying to escape from the leader hawk, and then, another hawk in the party team would immediately continue the chase. These switching tactics confuse the targeted prey, attempt to eliminate the identified prey and increase its risk. Finally, the tired prey cannot escape the hawks' team, as one of the mighty hawks slaughters the tired prey and shares it with the party members. This algorithm has powerful neighborhood search characteristic, adaptability, scalability and flexibility [223]. But, it has low accuracy, slow rate of convergence, chance of easily being trapped into local optima, and there is no theatrical converging study frame [224]. Its mathematical model is explained, as shown below [225-227]:

Phase 1: Exploration period: Initialize the position

The Harris Hawks perch at random locations, and wait in various areas to observe and monitor the prey before attacking. The leader from the family of hawks perches on the target based on the spot. This phenomenon is described based on a mathematical equation for changing the distance between the prey and a bunch of hawks, as given below:

$$X(t+1) = \begin{cases} X_{r_i}(t) - r_1\left|X_{r_i}(t) - 2r_2X(t)\right|, q \geq 0.5 \\ X_{rabbit}(t) - X_m(t) - r_3\left(LB + r_4(UB - LB)\right), q < 0.5 \end{cases} \quad (3.71)$$

where $X(t+1) \in$ {position vector of hawks in the next iteration t}, $X_{rabbit}(t) \in$ {position of rabbit}, $X(t) \in$ {current position vector of hawks}, r_1, r_2, r_3 and $r_4 \in$ {random numbers in [0,1] threshold}, which are updated in each iteration, q is equal chance for each perching strategy changing randomly, LB and $UB \in$ {upper and lower bounds of the variables}, $X_{ri}(t) \in$ {randomly selected hawk from the current population}, and $X_m(t) \in$ {average position of the current population of hawks}. The value of $X_m(t)$ is obtained using the following equation:

$$X_m(t)=\frac{1}{N}\sum_{i=1}^{N}X_i(t) \tag{3.72}$$

where N is the total number of hawks and $X_i(t)$ is the location of each hawk at iteration t.

Phase 2: Transition from exploration to exploitation

In this phase, HHO alters from the exploration to exploitation phase, while changing between different exploitative behaviors according to the escaping energy (E) of the prey. The prey escape energy can be modelled using Eq. (3.73).

$$E = 2E_0\left(1-\frac{t}{IT}\right) \tag{3.73}$$

where $E \in$ {escaping energy of the prey}, and $E_0 \in$ {initial energy randomly changing within [−1, 1]}.

Phase 3: Exploitation period

In this phase, the Harris Hawks perform surprise pounce by attacking the intended prey detected in the previous phase. However, preys often attempt to escape from the dangerous situations. Hence, different chasing styles occur in real situations. According to escaping behavior of the prey and chasing behavior of the Harris Hawks, four possible strategies are proposed in HHO to model the attacking stage. To model this strategy, and enable the HHO to switch between soft and hard besiege processes, the E parameter is utilized [222].

Strategy 1: Soft besiege

When r (chance of a prey in successfully escaping) ≥ 0.5 and $|E| \geq 0.5$, the rabbit still has enough energy.

$$X(t+1)=\Delta X(t)-E\left|JX_{rabbit}(t)-X(t)\right| \tag{3.74}$$

$$\Delta X(t)=X_{rabbit}(t)-X(t) \tag{3.75}$$

where $\Delta X(t)$ is the difference between the position vector of the rabbit and current location in iteration t, $J = 2(1 - r_5)$ represents the random jump strength of the rabbit throughout the escaping procedure, and r_5 is a random number within [0, 1]. The value of J changes randomly in each iteration to simulate the nature of rabbit motions.

Strategy 2: Hard besiege

When $r \geq 0.5$ and $|E| < 0.5$, the prey is so exhausted having a low escaping energy.

$$X(t+1)=X_{rabbit}(t)-E\left|\Delta X(t)\right| \tag{3.76}$$

Strategy 3: Soft besiege with progressive rapid dives

When still $|E| \geq 0.5$, but $r_i < 0.5$, the rabbit has enough energy to successfully escape and a soft besiege is constructed before the surprise pounce. This procedure is more intelligent than the previous case. It is supposed that the Harris Hawks can evaluate their next movement based on the following equation:

$$Y=X_{rabbit}(t)-E\left|JX_{rabbit}(t)-X(t)\right| \tag{3.77}$$

where Y is the location of hawks in soft besiege phase. They compare the movement to the previous dive to detect if the previous dive is better. If not, they would dive based on the Levy flight (LF) pattern using Eq. (3.78):

$$Z = Y + S \times LF(D) \tag{3.78}$$

where Z is the dive based on LF-based pattern in soft besiege phase, D is the dimension of the problem, S is a random vector by size $1 \times D$ and LF is the Levy flight function.

Finally, the updating strategy of the hawks in this phase is presented as below:

$$X(t+1) = \begin{cases} Y \text{ if } F(Y) < F(X(t)) \\ Z \text{ if } F(Z) < F(X(t)) \end{cases} \tag{3.79}$$

It should be noted that in all search agents, only the better position of Y or Z would be selected to the next iteration.

Strategy 4: Hard besiege with progressive rapid dives

When $|E| < 0.5$ and $r_i < 0.5$, the rabbit has not enough energy to escape, and a hard besiege is constructed before the surprise pounce to catch and kill the prey. The hawks try to decrease their distances from the escaping prey. So, the following rule is performed with different Y and Z:

$$X(t+1) = \begin{cases} Y_m \text{ if } F(Y_m) < F(X(t)) \\ Z_m \text{ if } F(Z_m) < F(X(t)) \end{cases} \tag{3.80}$$

where $Y_m = X_{rabbit}(t) - E\left|JX_{rabbit}(t) - X_m(t)\right|$, $Z_m = Y_m + S \times LF(D)$, Y_m is the location of hawks in hard besiege phase and Z_m is the next location of hawks for the new iteration in hard besiege phase.

Algorithm 3.3.2: Pseudo-code of HHO algorithm

Inputs: N and IT
Outputs: Location of rabbit and its fitness value
Initialize random population X_i (i = 1, 2,...,N)
while (stopping condition is not met) do
    Calculate fitness values of hawks
    Set X_{rabbit} as the location of rabbit (best location)
    for (each hawk (X_i)) do
        Update the initial energy E_0 and jump strength J
[$E_0 = 2r_i() - 1$, $J = 2(1 - r_i())$]
        Update E using Eq. (3.73)
        if ($|E| \geq 1$) then [Exploration phase]
            Update location vector using Eq. (3.71)
        if ($|E| < 1$) then [Exploitation phase]
            if ($r_i \geq 0.5$ and $|E| \geq 0.5$) then [Soft besiege]
                Update the location vector using Eq. (3.74)
            else if ($r_i \geq 0.5$ and $|E| < 0.5$) then [Hard besiege]
                Update location vector using Eq. (3.76)
            else if ($r_i < 0.5$ and $|E| \geq 0.5$) then [Soft besiege]
with progressive rapid dives
                Update location vector using Eq. (3.79)
            else if ($r_i < 0.5$ and $|E| < 0.5$) then [Hard besiege]
with progressive rapid dives
                Update location vector using Eq. (3.80)
Return X_{rabbit}

3.3.3 Aquila optimizer

AO, suggested by Abualigah et al. [228], is encouraged by the Aquila's behavior in nature during the hunting process of the prey. Aquilas are the most common species of eagles and are known as the most popular birds of prey in the Northern Hemisphere [229]. It has been proved that AO has fast convergence rate, higher stability and consistency in solving complex optimization problems [230]. But, it has relatively weak local development ability [231]. Aquila has four main hunting behaviors for different types of prey. It can quickly switch its hunting techniques for different preys, and attack with its fast speed, and strong feet and claws [232]. This algorithm has four steps, i.e. a) selecting the search space by high soar with the vertical stoop, b) exploring within a diverged search space by contour flight with short glide attack, c) exploiting within a converged search space by low flight with slow descent attack, and d) swooping by walk and grab prey. A brief description of these strategies is provided as below:

a) Expanded exploration (X_1): Select the search space by high soar with vertical stoop

In this strategy, an Aquila determines the prey-catching area, and takes the best-catching area by flying high to search the prey location with a high glide on a vertical slope. This behavior is mathematically presented using the following equation:

$$X_1(t+1) = X_{best}(t) \times \left(1 - \frac{t}{IT}\right) + (X_m(t) - X_{best}(t) \times r_i) \tag{3.81}$$

where $X_{best}(t) \in$ {best position} and $X_m(t) \in$ {average position of all Aquila's in the current iteration}.

b) Narrowed exploration (X_2): Contour flight with short slide attack

In this strategy, the AO utilizes the twisting flight technique for hunting above the prey and it uses short gliding to attack the prey. The updated position is mathematically defined in Eq. (3.82):

$$X_2(t+1) = X_{best}(t) \times LF(D) + X_R(t) + (y - x) \times r_i \tag{3.82}$$

where $X_2(t+1) \in$ {solution of the next iteration}, which is generated by the second search method (X_2), $X_R(t) \in$ {random solution taken in the range of [1, N] at t^{th} iteration}, $LF(D) = s \times (r_i \times \sigma)/|r_i|^{(1/\beta)}$, $y = r \times \cos(\theta)$, $x = r \times \sin(\theta)$, $r = r_0 + U \times D_1$, $\theta = -\omega \times D_1 + \theta_1$ and $\theta_1 = (3 \times \pi)/2$.

where ω = a small value fixed to 0.005, U = a small value fixed to 0.00565, D_1 = integer number from 1 to the length of the search space, s = a small value fixed to 0.01, β = a small value fixed to 1.5, σ = a constant which depends on the value of β, and r_0 = value between 1 and 20 for fixed number of search cycles.

c) Expanded exploitation (X_3): Short flight with delayed attack

In this strategy, when the catching area of prey is accurately selected, the Aquila drops away vertically towards the prey with a front attack to discover its reaction. This method is called a short flight with delayed attack. The hunting behavior of AO in this strategy is defined using Eq. (3.83):

$$X_3(t+1) = (X_{best}(t) - X_m(t)) \times \alpha - r_i + ((UB - LB) \times r_i + LB) \times \delta \tag{3.83}$$

where $X_{best}(t) \in$ {approximate location of the prey until t^{th} iteration (the best-obtained solution)}, and α and $\delta \in$ {exploitation adjustment parameters (0.1)}.

d) Narrowed exploitation (X_4): Diving by walking and catching prey

In this strategy, the Aquila hunts the prey on the land according to their stochastic movements. This behavior of AO is mathematically expressed in Eq. (3.84).

$$X_4(t+1) = QF \times X_{best}(t) - (G_1 \times X(t) \times r_i) - G_2 \times LF(D) + r_i \times G_1 \quad (3.84)$$

where $QF = t^{\frac{2 \times r_i - 1}{(1-IT)^2}}$, $G_1 = 2 \times r_i - 1, G_2 = 2 \times \left(1 - \frac{t}{IT}\right)$, $QF \in$ {quality function used to equilibrium the search strategies}, $G_1 \in$ {various motions of AO for tracking the prey}, $G_2 \in$ {decreasing values from 2 to 0}, which denotes the flight slope of AO used to follow the prey during the slope from the first location to the last location, and $X(t) \in$ {current solution at t^{th} iteration}.

Algorithm 3.3.3: Pseudo-code of AO algorithm

```
Initialize population X
Initialize parameters (i.e. α, δ)
while (final condition is not reached) do
    Calculate fitness function value
    Obtain the best solution, X_best(t)
for i = 1,2,...,N do
        Update mean value of the current solution, X_m(t)
        Update x, y, G1, G2, LF(D)
                Step 1: Expanded exploration (X1)
                Update current solution using Eq. (3.81)
                if Fitness(X1(t + 1)) < Fitness(X_best(t)) then
X_best(t) = X1(t + 1)
                end if
            else
                Step 2: Narrowed exploration (X2)
                Update current solution using Eq. (3.82)
                if Fitness(X2(t + 1)) < Fitness(X_best(t)) then
                    X_best(t) = X2(t + 1)
                end if
        else
                Step 3: Expanded exploitation (X3)
                Update current solution using Eq. (3.83)
                if Fitness(X3(t + 1)) < Fitness(X_best(t)) then
                    X_best(t) = X3(t + 1)
                end if
            else
                Step 4: Narrowed exploitation (X4)
                Update current solution using Eq. (3.84)
                if Fitness(X4(t + 1)) < Fitness(X_best(t)) then
                    X_best(t) = X4(t + 1)
                end if
        end for
end while
Return the best solution (X_best)
```

3.3.4 Bat algorithm

The BA, proposed by Yang [233], is encouraged by microbats' echolocation capability and social behavior. Among all bats, microbats use echolocation to separate their prey, escape obstacles and recognize their roosting gaps in the dark. Microbats can also use time delay between their ears and noise deviations to sense three-dimensional surroundings. Mainly, some features of the echolocation are chosen in the optimization problem so that they can be linked with the objective function, which makes it possible to express a smart algorithm. Microbats emit loud sounds and hear back the echo that comes from the nearby substances. The sound beat they use differs based their catching plans. Studies have expressed that microbats release sound waves with frequencies in the range of 20-150 kHz. They also have decent sight and hearing skill. To determine position and size of their prey, they trust the frequency of the echo reaching their ears. The BA is expressed by idealizing the echolocation behavior of bats, which contains behavior of the microbats and acoustics of echolocation in catching their prey [234-236].

For simplicity, the following approximate or idealized rules are considered:

a) All bats use echolocation to sense distance, and they also know the difference between food/prey and background barriers in some magical way.

b) Bats fly randomly with velocity v_i at position x_i with a fixed frequency $f_{\min}$, varying wavelength λ and loudness A_0 to search for prey. They can automatically adjust the wavelength (or frequency) of their emitted pulses and rate of pulse emission $r_e \in [0,1]$, depending on the proximity of their target.

c) Although the loudness can vary in many ways, it is assumed that the loudness changes from a large (positive) A_0 to a minimum constant value $A_{\min}$.

In BA, the rules for how their positions x_i and velocities v_i in a d-dimensional search space are updated, need to be defined. The new solutions $x_i^{t_m}$ and velocities $v_i^{t_m}$ at time step t_m are given using the following equations:

$$f_i = f_{\min} + (f_{\max} - f_{\min}) \times \beta \tag{3.85}$$

$$v_i^{t_m} = v_i^{t_m - 1} + (x_i^{t_m} - x_*) \times f_i \tag{3.86}$$

$$x_i^{t_m} = x_i^{t_m - 1} + v_i^{t_m} \tag{3.87}$$

where $\beta \in$ {random vector drawn from a uniform distribution [0,1]}, and x_* is the current global best location (solution) which is identified after comparing all the solutions among all the N bats. As the product $\lambda_i f_i$ is the velocity increment, either f_i (or λ_i) can be used to adjust the velocity change while fixing the other factor λ_i (or f_i), depending on the type of the problem of interest. In this paper, the values of $f_{\min}$ and $f_{\max}$ are set as 0 and 100 respectively, depending on the domain size of the considered problem. Initially, each bat is randomly assigned a frequency which is drawn uniformly from $[f_{\min}, f_{\max}]$. The update of velocities and positions of bats have some similarity to the procedure in the standard PSO algorithm as f_i essentially controls the pace and range of movement of the swarming particles. To a

degree, BA can be considered as a balanced combination of PSO, and intensive local search controlled by the loudness and pulse rate. Although BA is simple to implement, robust and has few algorithm-specific parameters, but its performance may decline and exploration capacity may become poorer with increase of problem dimensions, making it almost impossible to converge to the global optimal solution [237].

Algorithm 3.3.4: Pseudo-code of BA

Objective function $f(x)$, $x = (x_1, ...,x_d)^T$
Initialize bat population x_i (i = 1,2,…,N) and v_i
Define pulse frequency f_i at x_i
Initialize pulse rate r_e and loudness A_i
While ($t < IT$)
Generate new solutions by adjusting frequency, and update velocities and locations/solutions
    if ($r_i > r_e$)
    Select a solution among the best solutions
    Generate a local solution around the selected best solution
    end if
    Generate a new solution by flying randomly
    if ($r_i < A_i$ and $f(x_i) < f(x_*)$)
    Accept new solutions
    Increase r_i and reduce A_i
    end if
Rank bats and find the current best x_*
end while
Post-process results and visualization

3.3.5 Grey wolf optimizer

GWO is also a nature-inspired metaheuristic algorithm encouraged by the leadership hierarchy-based hunting technique of grey wolves [238]. There are four classes of grey wolves in the leadership pyramid, i.e. alpha (α), beta (β), delta (δ) and omega (ω); and the three main phases of hunting are searching for prey, encircling the prey and attacking the prey. For encircling the prey, the following mathematical models are proposed [239, 240]:

$$\vec{D} = \left| \vec{C} \times \vec{X}_p(t) - \vec{X}(t) \right| \tag{3.88}$$

$$\vec{X}(t+1) = \vec{X}_p(t) - \vec{A} \times \vec{D} \tag{3.89}$$

The corresponding vectors are calculated using the following equations:

$$\vec{A} = 2\vec{a} \times \vec{r}_1 - \vec{a} \tag{3.90}$$

$$\vec{C} = 2 \times \vec{r}_2 \tag{3.91}$$

where $\vec{A}$ and $\vec{C} \in$ {coefficient vectors}, $\vec{X}_p(t) \in$ {position vector of the prey}, $\vec{X} \in$ {position vector of a grey wolf}, $\vec{a}$ is linearly decreased from 2 to 0 based on the progress of iterations, and $\vec{r}_1$ and $\vec{r}_2 \in$ {random vectors in [0, 1]}.

Once the encircling of prey is over, a grey wolf starts to catch and hunt the prey (i.e. best solution). The best nominees from the wolves (α, β and δ) would know the exact position and information about the prey. With respect to the positions of the best three wolves, ω wolves would also change their positions. The hunting technique of grey wolves can mathematically be defined as follows:

$$\vec{D}_\alpha = \left|\vec{C}_1 \times \vec{X}_\alpha - \vec{X}\right| \tag{3.92}$$

$$\vec{D}_\beta = \left|\vec{C}_2 \times \vec{X}_\beta - \vec{X}\right| \tag{3.93}$$

$$\vec{D}_\delta = \left|\vec{C}_3 \times \vec{X}_\delta - \vec{X}\right| \tag{3.94}$$

$$\vec{X}_1 = \left|\vec{X}_\alpha - \vec{A}_1 \times \vec{D_\alpha}\right| \tag{3.95}$$

$$\vec{X}_2 = \left|\vec{X}_\beta - \vec{A}_2 \times \vec{D_\beta}\right| \tag{3.96}$$

$$\vec{X}_3 = \left|\vec{X}_\delta - \vec{A}_3 \times \vec{D_\delta}\right| \tag{3.97}$$

$$\vec{X}(t+1) = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3} \tag{3.98}$$

Finally, grey wolves attack on prey and finish the hunting process when it stops moving. To signify it mathematically, the value of $\vec{a}$ is reduced. The variation range of $\vec{A}$ is also reduced by $\vec{a}$, $\vec{A}$ is a random value in the interval [-*a*, *a*]. When the random values of $\vec{A}$ are in [-1, 1], the next location of the search agent can be anywhere between its current location and location of the prey. The values of $|A| < 1$ would force the wolves to attack the prey and $|A| > 1$ would force the grey wolves to diverge from the prey to find out a fitter prey, hopefully. After hunting the prey, the grey wolves search for the new prey in the next iteration. This process would continue until a termination criterion is satisfied. Although GWO has low solving accuracy and poor local searching ability, it has several other advantages, like easy to implement, simple principles and fewer parameters [241, 242].

Algorithm 3.3.5: Pseudo-code of GWO

```
Initialize grey wolf population X_i (i = 1,2,...,N)
Initialize a, A and C
Calculate fitness of each search agent
X_α = the best search agent
X_β = the second-best search agent
X_δ = the third-best search agent
while (t < IT)
        for each search agent
                Update the position of the current search agent
        end for
        Update a, A and C
        Calculate the fitness of all search agents
        Update X_α, X_β and X_δ
        t = t + 1
```

end while
return X_α

3.4 Foraging behavior-based metaheuristics

Swarm intelligence-based algorithms have been developed conceptualizing social behavior and intelligence of a group of living things, like ants, birds, swarms, schools etc. Some of them are based on the animal's foraging and preying behaviors. Finding natural food sources is the process of foraging. In a large community, it includes all the methods for discovering, utilizing, recovering and storing food resources. Foraging behavior-based metaheuristics have many advantages over their peers, like capability to automatically zoom onto a region of promising solutions leading to faster convergence, excellent local and global search abilities, better solution accuracy, superior hybridization ability with other algorithms, capability to preserve variety of solutions within the population as the iteration goes on, better parametric control etc. Lack of effective exploring skill, need of transition between exploration and exploitation at appropriate time requiring an enhanced control technique etc. are the drawbacks of these algorithms [51].

It is worth mentioning here that choosing appropriate values of different algorithm-specific parameters has significant roles as they directly influence the search behavior, convergence speed and solution quality of these algorithms. Adjusting their values is extremely pivotal for achieving optimal solutions and improving efficacy of the considered metaheuristics. Table 3.5 provides values of the essential parameters considered for all the five metaheuristics.

Table 3.5 Values of different algorithm specific parameters

Algorithm	Parameter	Value
DOA	$s = a$	0.1
	c	0.7
	$f = e$	1
AVOA	P_1	0.6
	P_2	0.4
	P_3	0.6
	L_1	0.8
	L_2	0.2
	u	2.5
GOA	$C_{\min}$	0.00001
	$C_{\max}$	1.0
FOA	-	-
BSA	P	[0.8, 1.0]
	$C = S$	1.5
	$a_1 = a_2$	1.0
	FL	[0.5, 0.9]
	FQ	3

3.4.1 Dragonfly optimization algorithm

The DOA [243] is based on static and dynamic activities, including hunting (exploration) and migration (exploitation) of dragonflies. As compared to other well-known algorithms, like GA and PSO, it is capable of enhancing the initial random population for a specific problem, and converging to the optimal solution globally, while providing extremely economical outcomes [243]. Like other metaheuristic algorithms, it also carries out the target search process. The dragonflies build sub-swarm groups that fly back and forth over small areas for food and attract flying prey, like butterflies and mosquitoes in the exploration phase. A static swarm is mostly characterized by local motions and quick variations in the flying track. In the exploitation stage, a large number of dragonflies migrate in one track over lengthy distances to distract from outward enemies. As the primary goal of every swarm is survival, thus, everyone in the group should be drawn to food sources and distract outer adversaries. Due to huge amount of social behaviors in its construct, exploration and exploitation stages may not be properly balanced [244]. In this algorithm, exploration and exploitation adopt five different operators, like separation, alignment, cohesion, attraction and distraction. The following mathematical models describe behaviors of each swarm in this algorithm [243]: Separation is computed using Eq. (3.99):

$$SN_i = -\sum_{j=1}^{N} Y - Y_j \tag{3.99}$$

where SN_i is the separation of i^{th} individual, Y is the position of the current individual, Y_j is the position of j^{th} neighboring individual and N is the number of neighboring individuals.

The following equation represents alignment:

$$AL_i = \frac{\sum_{j=1}^{N} V_j}{N} \tag{3.100}$$

where AL_i is the alignment of i^{th} individual and V_j is the velocity of j^{th} neighboring individual.

Cohesion can be represented as follows:

$$CH_i = \frac{\sum_{j=1}^{N} Y_j}{N} - Y \tag{3.101}$$

where CH_i is the cohesion of i^{th} individual.

Attraction towards a food source can be represented using Eq. (3.102):

$$FS_i = Y^{+} - Y \tag{3.102}$$

where FS_i is the attraction of i^{th} individual towards the food source and Y^{+} is the food source position.

On the other hand, Eq. (3.103) represents distraction from an adversary:

$$EP_i = Y^{-} - Y \tag{3.103}$$

where EP_i is the distraction of i^{th} individual and Y^{-} is the position of the adversary.

The positions of the artificial dragonflies are updated in the search space and their movements are simulated. For this purpose, two vectors, i.e. step vector (ΔY) and position vector (Y) are considered. The step vector which defines direction of movement of the dragonflies is represented as follows:

$$\Delta Y_{t+1} = (s\,SN_i + a\,AL_i + c\,CH_i + f\,FS_i + e\,EP_i) + w\Delta Y_t \tag{3.104}$$

where s is the weight of separation, a is weight of alignment, c is cohesion weight, f is the food factor, e is the adversary factor and w is the weight of inertia.

Based on the calculated values of step vector, the position vectors are estimated as follows:

$$Y_{t+1} = Y_t + \Delta Y_{t+1} \tag{3.105}$$

where Y_{t+1} is the position vector at $(t + 1)^{th}$ iteration and ΔY_{t+1} is the step vector at $(t + 1)^{th}$ iteration. To improve exploration, randomness and stochastic behavior of the artificial dragonflies, they should fly around the search space based on the random walk (Levy flight) in absence of any neighboring solution. Thus, the position of the dragonflies is modified employing the following equation:

$$Y_{t+1} = Y_t + Levy(d) \times Y_t \tag{3.106}$$

where d denotes the dimension of the position vectors. The Levy flight is calculated as follows:

$$Levy(y) = 0.01 \times \frac{r_1 \times \mu}{|r_2|^{\frac{1}{\beta}}} \tag{3.107}$$

where β is a constant (considered as 1.5), r_1 and r_2 are the two random numbers in [0, 1], and μ is computed as shown below:

$$\mu = \left(\frac{\Gamma(1+\beta) \times \sin\left(\frac{\pi\beta}{2}\right)}{\Gamma\left(\frac{1+\beta}{2}\right) \times \beta \times 2^{\left(\frac{\beta-1}{2}\right)}} \right)^{\frac{1}{\beta}} \tag{3.108}$$

where $\Gamma(y) = (y - 1)!$

Algorithm 3.4.1: Pseudo-code of DOA

```
Objective: Positions and fitness values of all dragonflies
Generate initial dragonfly population Y_i (i = 1, 2,…,N)
Generate initial step vectors ΔY_i (i = 1, 2,...,N)
while (t ≤ IT) do
    Calculate fitness value for each dragonfly
    Update food source and adversary position
Update w, s, a, c, f and e
Calculate SN, AL, CH, FS and EP using Eqs. (3.99)-(3.103)
    Update neighbouring radius
if a dragonfly has at least one neighboring dragonfly
        Update velocity vector employing Eq. (3.104)
        Update position vector applying Eq. (3.105)
else
        Update position vector using Eq. (3.106)
end if
    Check and correct the new positions based on the variable boundaries
t = t + 1
end while
```

3.4.2 African vultures optimization algorithm

In AVOA [245], the African vultures' foraging behavior and movement patterns are simulated. Based on the distinct despotic characteristic of the vultures' behavior, this algorithm divides a large number of vultures into two groups. To categorize the vultures, it first determines fitness function of each solution in the initial population. The best solution is assigned to the best and first vulture, while the second solution is considered to represent the second-best vulture. One of the best two vultures is moved or replaced by other members of the population based on the evaluated performance [245]. The main natural purpose of vultures, i.e. group living in search of food, justifies separation of their groups in this algorithm. Each group of vultures has a unique difficulty in locating food and consuming it. Vultures have the characteristic of consuming large quantities of food, and their hours-long search for food makes them escaping from the hunger trap. Based on the assumption that the weakest and hungriest vulture represents the worst solution, the vultures try to avoid the worst solution and turn up with the best solution. The two best solutions are recognized as the strongest and best vultures, while others aim to be nearer to the best solution [246].

The exploration and exploitation mechanisms of AVOA are more intensified than other metaheuristic algorithms. Although there are few limitations, like presence of exploitation in the precise exploration stage and replacement of weaker vultures at random [50], application of a random search method boosts both the exploitation and exploration abilities of this algorithm [247]. Based on the fundamental ideas of the African vultures, AVOA is developed consisting of the following fundamental phases:

Phase 1: Identification of the best vulture in any group

Once the initial population is formulated, the best solution is considered as the best vulture of the first group, and the second-best solution is treated as the best vulture of the second group. The following equation is applied to rank the best solutions for the first and second groups against other potential solutions.

$$K(i)=\begin{cases} BestVulture_1 \; if \; p_i = L_1 \\ BestVulture_2 \; if \; p_i = L_2 \end{cases} \tag{3.109}$$

where L_1 and L_2 are the two parameters (between 0 and 1) to be determined prior to the search operation such that their sum must be one, and $K(i)$ is the probability of selecting the vultures to move towards one of the best solutions in each group. From each group, the probability of selecting each of the best solutions is estimated using the roulette wheel principle.

$$p_i = \frac{f_i}{\sum_{i=1}^{n} f_i} \tag{3.110}$$

where f_i is the fitness value of the first and second group vultures, and n is the total number of first and second group vultures.

Phase 2: Calculation of starvation rate of the vultures

After the vultures are satisfied with food, they have high levels of energy allowing them to search for food over longer distances. But, if they are hungry, they lack energy to fly long or search for food against other more powerful vultures, making them aggressive. This phenomenon can be mathematically modeled using the following equations:

$$m = h \times \left(\frac{\pi}{2} \times \frac{t}{IT}\right) + cos\left(\frac{\pi}{2} \times \frac{t}{IT}\right) - 1 \tag{3.111}$$

$$F = (2 \times r_i + 1) \times r_1 \times \left(1 - \frac{t}{IT}\right) + m \tag{3.112}$$

where F signifies that the vultures are satisfied, r_1 is a random number in [-1, 1], h is selected randomly between -2 and 2, and r_i has a uniformly distributed random value in [0, 1]. When r_1 is less than zero, the vulture is starving; and if it is equal to zero, the vulture is satisfied. In Eq. (3.112), sin and cos are the functions of sine and cosine respectively, and u is a variable with a fixed value established prior to the optimization process.

Phase 3: Exploration

The African vultures have excellent vision, a great ability to locate food, and a strong ability to spot weak, dying animals in the wild. Each vulture carries out a random search of its surroundings for food which can be modeled using Eq. (3.113).

$$P(i+1) = \begin{cases} if\ P_1 \geq r_{P1}, then\ Eq.(3.114) \\ if\ P_1 < r_{P1}, then\ Eq.(3.116) \end{cases} \tag{3.113}$$

$$P(i+1) = K(i) - D(i) \times F \tag{3.114}$$

$$D(i) = \left|X \times K(i) - P(i)\right| \tag{3.115}$$

where $P(i + 1)$ is the next vulture vector position, $P(i)$ is the current vulture vector position, and X is the position where the vultures randomly move to defend their prey from others. X is a coefficient vector responsible to increase the random motion, and is equal to $(2 \times r_i)$, r_{P1} is an exploitation phase uniformly distributed random number in [0, 1], and P_1 is a parameter (between 0 and 1) employed for strategy selection.

$$P(i+1) = K(i) - F + r_i \times ((UB - LB) \times r_2 + LB) \tag{3.116}$$

where K(i) denotes one of the best vultures selected using Eq. (3.109) and r_2 is considered to increase the coefficient of random nature.

Phase 4: Exploitation

The AVOA reaches the exploitation stage when $|F| < 1$. Two parameters P_2 and P_3 control the degree of selecting each strategy in each internal phase, and their values range between 0 and 1. Selection of the strategies in the first phase is controlled by parameter P_2, while parameter P_3 helps in choosing the strategy in the second phase. In the first phase of exploitation (when $|F|$ is between 0.5 and 1), two different rotating flight and siege-fight strategies are considered. As mentioned earlier, P_2 is used to determine the choice of each strategy. Initially, the value of r_{P_2} (uniformly distributed random number in [0, 1]) is generated. After comparing its value with P_2, any of the two strategies (siege-flight or rotating flight) can be implemented, which is illustrated using the following equation:

$$P(i+1)=\begin{cases} if\ P_2 \geq r_{P2}, then\ Eq.(3.118) \\ if\ P_2 < r_{P2}, then\ Eq.(3.122) \end{cases} \quad (3.117)$$

The vultures become relatively satisfied having sufficient energy when $|F| \geq 0.5$. When a large number of vultures gather around a single food source, serious food conflicts may arise. Physically stronger vultures avoid sharing of food with the weaker ones, but the weaker vultures attempt to gather around the stronger vultures starting conflicts in an effort to exhaust them and getting food from them. This stage can be modeled using Eqs. (3.118) and (3.119).

$$P(i+1)=D(i)\times(F+r_i)-d(t) \quad (3.118)$$

$$d(t)=K(i)-P(i) \quad (3.119)$$

The vultures occasionally make a rotational flight to model their spiral motion. The rotational flight is expressed using the following equations:

$$S_1=K(i)\times\left(\frac{r_i\times P(i)}{2\pi}\right)\times\cos(P(i)) \quad (3.120)$$

$$S_2=K(i)\times\left(\frac{r_i\times P(i)}{2\pi}\right)\times\sin(P(i)) \quad (3.121)$$

$$P(i+1)=K(i)-\left(S_1+S_2\right) \quad (3.122)$$

where S_1 and S_2 are the spiral motions between all vultures and one of the two best vultures respectively during the rotational flight. The location of the vultures is finally updated using Eq. (3.122).

The siege and violent struggle for food is carried out during the second phase of exploitation when the value of $|F|$ is less than 0.5. At the start, a uniform random number, r_{P3} is generated between 0 and 1. Its value is then compared with parameter P_3, and based on the mathematical condition, there may be accumulation of several types of vultures over the food source or aggressive siege-fight. This process is described using Eq. (3.123):

$$P(i+1)=\begin{cases} if\ P_3 \geq r_{P3}, then\ Eq.(3.124) \\ if\ P_3 < r_{P3}, then\ Eq.(3.127) \end{cases} \quad (3.123)$$

The following two equations are developed to model movement of vultures based on accumulation of various species of vultures on one food source.

$$A_1=BestVulture_1(i)-\frac{BestVulture_1(i)\times P(i)}{BestVulture_1(i)-P(i)^2}\times F \quad (3.124)$$

$$A_2=BestVulture_2(i)-\frac{BestVulture_2(i)\times P(i)}{BestVulture_2(i)-P(i)^2}\times F \quad (3.125)$$

where the best vulture of the first group in the current iteration is denoted by $BestVulture_1(i)$, and the best vulture of the second group in the current iteration is represented by $BestVulture_2(i)$.

The aggregation of all the vultures is carried out applying Eq. (3.126).

$$P(i+1)=\frac{A_1+A_2}{2} \quad (3.126)$$

When the value of $|F|$ is less than 0.5, the leader vultures become starved and weak, not having sufficient energy to fight with the others. Thus, other vultures become aggressive in search of their food, moving

in different directions towards the leader vulture. This aggressive competition for food is modeled using the following equation:

$$P(i+1) = K(i) - |d(t)| \times F \times Levy(d) \tag{127}$$

where $d(t)$ is the separation of the vulture from one of the best vultures of the two groups. In this equation, Levy flight, which can be computed using Eqs. (3.107)-(3.108), is considered to enhance efficacy of this algorithm.

Algorithm 3.4.2: Pseudo-code of AVOA

Objective: Position and fitness value of vulture
Initialize N and IT
First best location: $P_{Bestvulture1}$ is the position of African vulture
Second best location: $P_{Bestvulture2}$ is the position of African vulture
Generate initial vulture population P_i (i = 1, 2,…,N) randomly
While stopping criterion is not met do
for each African vulture (AV) in population do
    Calculate fitness for AV
end for
Find best AV
    for (each vulture (P_i)) do
        Use Eq. (3.109) to calculate $K(i)$
            Use Eq. (3.113) to update F
            if ($|F| \geq 1$) and ($P_1 \geq r_{P1}$) then
                Use Eq. (3.114) to update position of AV
            else
        Use Eq. (3.116) to update position of AV
end for
end while
    if ($0.5 \leq |F| < 1$) then
        if ($P_2 \geq r_{P2}$) then
            Use Eq. (3.118) to update position of AV
        else
            Use Eq. (3.122) to update position of AV
    else
        if ($P_3 \geq r_{P3}$) then
            Use Eq. (3.126) to update position of AV
        else
            Use Eq. (3.127) to update position of AV
Return $P_{Bestvulture1}$

3.4.3 Grasshopper optimization algorithm

Like all other metaheuristic algorithms, GOA [248] also performs the target search procedure by hunting (exploration) and migration (exploitation), and these are their static and dynamic behaviors. The distinctive feature of grasshopper swarms is that they exhibit swarming behavior both as nymphs and as adults. The grasshoppers in the swarm move slowly and take small movements when they are in the larval stage. But when they become adults, they move abruptly over longer distances. Another

important feature of the grasshopper swarm is their search process for food sources [249]. This algorithm searches for a global optimal solution to a problem using swarm population in nature, and has some advantages, like higher efficiency in solving both constrained and unconstrained optimization problems with excellent accuracy, capability of deriving good solutions, and ease of development. It has also certain limitations, including a tendency to enter local optima and a sluggish convergence rate [250]. The mathematical model utilized to simulate the swarming behavior of grasshoppers is documented as below:

$$X_i = SI_i + GF_i + A_{W_i} \tag{3.128}$$

where X_i is i^{th} grasshopper's position, SI_i is the social collaboration, GF_i is i^{th} grasshopper's gravity force and A_{W_i} is the wind advection.

To provide random behavior, the above equation can be remodeled as below:

$$X_i = r_1 SI_i + r_2 GF_i + r_3 A_{W_i} \tag{3.129}$$

where r_1, r_2 and r_3 are three random numbers in [0,1].

In Eq. (3.129), the parameter SI_i can be defined using Eq. (3.130):

$$SI_i = \sum_{\substack{j=1 \\ j \neq i}}^{N} s(d_{ij}) \hat{d}_{ij} \tag{3.130}$$

where N is the number of grasshoppers, $\hat{d}_{ij}$ is a unit vector from i^{th} grasshopper to j^{th} grasshopper, s is a function to describe intensity of social forces, and d_{ij} is the distance between i^{th} grasshopper and j^{th} grasshopper, i.e. $d_{ij} = |x_j - x_i|$.

$$\hat{d}_{ij} = \frac{\left| x_j - x_i \right|}{d_{ij}} \tag{3.131}$$

$$s = f e^{\frac{-q}{l}} - e^{-q} \tag{3.132}$$

where f and l indicate intensity of attraction and attractive length scale respectively, and q is a real value in [0, 1] defining the probability of changing positions.

By substituting values of components, SI_i, GF_i and A_{W_i} in Eq. (3.128), the following equation can be derived:

$$X_i = \sum_{\substack{j=1 \\ j \neq i}}^{N} s\left(\left| x_j - x_i \right| \right) \frac{x_j - x_i}{d_{ij}} - g\hat{e}_g + v\hat{e}_w \tag{3.133}$$

$$s(q) = f e^{\frac{-q}{l}} - e^{-q} \tag{3.134}$$

where g is the gravitational constant, v is a constant drift, $\hat{e}_g$ denotes a unity vector towards center of the earth, and $\hat{e}_w$ is a unity vector in the direction of wind. As the swarm fails to converge to a predefined point and the grasshoppers quickly reach their comfort zone, the above mathematical model

cannot be easily solved. To resolve this problem, the following improved version has been proposed [248]:

$$X_i^d = c\left(\sum_{\substack{j=1 \\ j\neq 1}}^{N} c\frac{UB_d - LB_d}{2} s\left(\left|x_j - x_i\right|\right)\frac{x_j - x_i}{d_{ij}}\right) + \widehat{T_d} \tag{3.135}$$

where $\widehat{T_d}$ denotes the target value of D^{th} dimension, UB_d and LB_d are the upper and lower bounds in D^{th} dimension respectively, and X_i^d is the position of i^{th} grasshopper in D^{th} dimension. In the above equation, the first parameter, c reduces movements of the grasshoppers around the target (similar to inertia weight in PSO); while the second parameter, c minimizes attraction, comfort and repulsion zones between the grasshoppers. For proper balance between exploration and exploitation, value of the first parameter needs to be proportionally reduced with the number of iterations. On the other hand, the second parameter which is involved in proportionally reducing the comfort zone along with the number of iterations, can be expressed using Eq. (3.136):

$$c = c_{\max} - t \times \frac{c_{\max} - c_{\min}}{IT} \tag{3.136}$$

where $c_{\max}$ and $c_{\min}$ are the maximum and minimum values of the parameter c respectively.

Algorithm 3.4.3: Pseudo-code of GOA

```
Objective: Current position
Generate initial population X_i (i = 1, 2,…,N)
Initialize c_max and c_min values
Initialize IT
Calculate fitness values of all grasshoppers (search agents)
Best search agent = T
while (t < IT)
    Use Eq. (3.136) to update the value of c
    for each grasshopper
        Use Eqs. (3.128)-(3.132) to normalize distances between the search agents
        Use Eq. (3.135) to update the current grasshopper position
        if the current grasshopper goes outside the boundaries, then bring it back
    end for
    update T if there is a better solution
    t = t + 1
end while
return T
```

3.4.4 Fruit-fly optimizer

The FFA [251] is developed utilizing techniques for optimization based on swarm intelligence, being motivated by fruit flies' foraging habits. The fruit flies have greater sensory abilities than the other species, especially with respect to eyesight and smell. They have osphresis organs to detect a wide range of odors in the atmosphere. They can also use their keen vision once they are close to the food source to locate food and other flies' location [251]. This algorithm offers the benefits of a clear principle, few input variables and simplicity. However, there are some disadvantages, like tendency

towards local extremum, and poor ability to solve highly dimensional, nonlinear problems [252]. The procedural steps of FFA are presented as below [253].

Step 1: Initialize location of fruit fly swarm

$$X\_axis = r_i(LR), Y\_axis = r_i(LR) \tag{3.137}$$

Step 2: For each fruit fly, specify a random direction and distance for food search employing osphresis

$$Xi = X\_axis + r_i(FR), Yi = Y\_axis + r_i(FR) \tag{3.138}$$

where i is i^{th} fruit fly in the swarm, FR is the fly range and LR is the swarm location range.

Considering unknown food source, calculate the distance to the origin (D_i) and smell concentration judgment value (fitness function) (S_i).

$$D_i = \sqrt{X_i^2 + Y_i^2} \tag{3.139}$$

$$S_i = \frac{1}{D_i} \tag{3.140}$$

Now, using the following equation, determine the smell concentration value ($Smell_i$):

$$Smell_i = fitness\ (Si) \tag{3.141}$$

The fruit fly having the maximum smell concentration value is identified among the swarm with its corresponding location, as given below:

[*bestSmell*, *bestIndex*] = max(*Smell*) (3.142)

Step 3: Vision search phase

In this step, the maximum smell concentration is maintained along with the related X and Y coordinates of that fly. The fruit fly swarm starts moving towards the location based on their vision.

Smellbest = *bestSmell*, *X_axis* = *X*(*bestIndex*), *Y*_axis = *Y*(*bestIndex*) (3.143)

Step 4: Repeat steps (2)-(3) until the maximum number of iterations is reached. Save the fruit fly's location index.

Algorithm 3.4.4: Pseudo-code of FOA

```
Objective: Maximum smell concentration and location
Generate initial LR and FR
Initialize IT
X_axis = rand(LR), Y_axis = rand(LR)
X_i = X_axis + rand(FR)
Y_i = Y_axis + rand(FR)
Calculate swarm fitness smell using Eqs. (3.139)-(3.141)
[bestSmell, bestIndex] = max(Smell)
Smellbest = bestSmell, X_axis = X(bestIndex), Y_axis = Y(bestIndex)
for i = 1: IT, all fruit flies
        X_i = X_axis + rand(FR), Y_i = Y_axis + rand(FR)
        Use Eqs. (3.139)-(3.141) to fitness Smell
        [bestSmell, bestIndex] = max(Smell)
        If bestSmell < Smellbest
                Smellbest = bestSmell, X_axis = X(bestIndex), Y_axis = Y(bestIndex)
        end if
```

end for
return *bestSmell*, *X*(*bestIndex*), *Y*(*bestIndex*)

3.4.5 Bird swarm algorithm

The BSA [254] is developed based on the social behavior and interaction of birds in nature. Foraging, flight and vigilance are the three primary social behaviors of birds which provide the principle of this algorithm [254]. BSA offers benefits of rapid convergence, higher precision, simplicity, ease of implementation, and excellent performance in solving continuous optimization problems. But, it faces some difficulty in solving combinatorial optimization problems (such as travelling salesman problem), and may fall into local optimum [255]. This algorithm is based on the following idealized rules describing behaviors of the birds as a whole [254]:

i) Each bird can select either vigilance or foraging behavior. The selected behavior can be modeled as a probabilistic decision.
ii) During foraging, each bird can keep track, and update both of its own and the swarm's prior best experience about the food source. It also helps them to find a path for food and their movement.
iii) During vigilance, each bird would make an effort to travel towards the center of the swarm. The birds having higher reserve would be more likely to stay nearer to the swarm's center than those with lower reserve, so that the chances of other predators attacking them in the center are low.
iv) Birds can periodically move from one location to another. During flying, they can shift between producing and scrounging. A producer would be the bird with the highest reserve, and a scrounger would have the lowest reserve. However, other birds can randomly be chosen as producers and scroungers.
v) The producer birds actively participate for food search, while scroungers follow a producer bird at random.

The above-mentioned rules can now be utilized to develop the corresponding mathematical model of BSA, presented as below:

a) According to the past and collective experience of the swarm, each bird forages the food employing Eq. (3.144):

$$y_{i,j}^{t+1} = y_{i,j}^{t} + (bp_{i,j} - y_{i,j}^{t}) \times C \times r_i(0,1) + (sp_j - y_{i,j}^{t}) \times S \times r_i(0,1) \quad (3.144)$$

where $y_{i,j}^{t}$ is the position of j^{th} element of i^{th} bird at t^{th} generation, and $bp_{i,j}$ and sp_j represent the i^{th} bird's best prior position and swarm's best prior position respectively. The two positive constants *C* and *S* are referred to as cognitive and social accelerated coefficients respectively. This foraging behavior is implemented considering Rule (ii).

b) The vigilance behavior, modeled using Rule (iii), can be described based on the following equations:

$$y_{i,j}^{t+1} = y_{i,j}^{t} + (mean_j - y_{i,j}^{t}) \times A_1 \times r_i(0,1) + (bp_{k,j} - y_{i,j}^{t}) \times A_2 \times r_i(-1,1) \quad (3.145)$$

$$A_1 = a_1 \times \exp\left(-\frac{bpFt_i}{sumFt + \varepsilon} \times N\right) \tag{3.146}$$

$$A_2 = a_2 \times \exp\left(\left(\frac{bpFt_i - bpFt_k}{\left|bpFt_k - bpFt_i\right| + \varepsilon} \times N\right)\frac{bpFt_k}{sumFt + \varepsilon} \times N\right) \tag{3.147}$$

where N is the number of virtual birds foraging for food and flying in a D-dimensional space, k is a positive integer ($k \neq i$) (randomly selected between 1 and N), $bpFt_i$ is the best fitness value of i^{th} bird, $sumFt$ is the sum of all the birds' best fitness values, a_1 and a_2 are two positive constants in [0, 2], ε is a small constant considered to escape zero-division, and $mean_j$ is the j^{th} element of the average position of the entire swarm. It is worthwhile to mention here that as each bird attempts to move nearer to the center of the swarm, the value of $A_1 \times r_i(0, 1)$ should be always less than one. When a bird reaches to the center of the swarm, A_2 is considered to simulate the induced effect caused due to a specific interference.

c) After reaching a new position, some birds act as the producers, while others try to feed themselves from the food source identified by the producers. These producing and scrounging behaviors (Rule (iv)) can be modeled employing the following equations:

$$y_{i,j}^{t+1} = y_{i,j}^{t} + rand_n(0,1) \times y_{i,j}^{t} \tag{3.148}$$

$$y_{i,j}^{t+1} = y_{i,j}^{t} + \left(y_{k,j}^{t} - y_{i,j}^{t}\right) \times \mathrm{FL} \times rand_n(0,1) \tag{3.149}$$

where *randn* is a Gaussian distributed random number having mean 0 and standard deviation value 1, $k \in [1,2,\ldots,N]$ ($k \neq i$) and $FL \in [0, 2]$ which signifies that the scrounger would follow the producer in search of food. It is assumed that at every FQ unit interval, each bird goes to a different place, where FQ is a positive integer.

Algorithm 3.4.5: Pseudo-code of BSA

```
Objective: Best solution and fitness values of birds
Initialize the population, t = 0 and define the related parameters
Generate number of birds in the population (N)
Generate IT
Probability of foraging for food = P
while (t < IT)
    if (t % FQ ≠ 0)
        for i = 1: N
            if r_i(0, 1) < P
                Birds forage for food [Use Eq. (3.144)]
            else
                Birds remain vigilance [Use Eq. (3.145)]
            end if
        end for
    else
        Separate the swarm into producers and scroungers
        for i = 1: N
            if i is a producer
                Producing [Use Eq. (3.148)]
```

```
                else
                Scrounging [Use Eq. (3.149)]
            end if
        end for
    end if
    Find new solutions
    if the new positions are better than the prior solutions, update them
t = t + 1
end while
```

3.5 Physics-based metaheuristics

Physics-based algorithms have been proposed based on the laws of gravity, molecular mass, fluid principle, electric current-produced magnetic field, nuclear reaction etc. Physics-based algorithms draw inspiration from the natural physical phenomena rather than living creatures. They have been developed by imitating specific chemical and physical rules. Compared to their counterparts, physics-based metaheuristics offer numerous benefits, such as ability to meet time constraints providing quick solutions, ease of implementation, minimum tuning parameters, quick convergence, clarity of execution, higher accuracy and robustness, suitability for nonlinear optimization problems, ability to generate workable and precise solutions, adaptive learning capacity etc. Their shortcomings include inefficiency to solve large-scale problems, and some of them require higher computation time, less sample space exploration and early convergence [256]. Acknowledging their benefits, this research work investigates suitability of five such physics-based algorithms, i.e. AOA, ASO, NRO, EFO and GSA, to optimize machining processes based on the past experimental data. Thus, these algorithms, which have commonly been applied to solve multi-objective optimization problems in a variety of engineering areas, are evaluated here based on their effectiveness, dependability and overall performance.

It is important to note that selecting the appropriate values for various algorithm-specific parameters is extremely crucial since they affect the search behavior, convergence speed and quality of the solutions. Table 3.6 enlists the values of the essential parameters taken into account for each of the five algorithms.

Table 3.6 Specific parameter values for different physics-based algorithms

Algorithm	Parameter	Value
AOA	C_1	2
	C_2	6
	C_3	2
	C_4	0.5
ASO	$\propto$	50
	β	0.1 to 1
NRO	P_{Fi}	0.75
	P_β	0.1
	freq	0.05
EFO	*Ps_rate, R_rate*	0.1 to 0.4
	P_field	0.05 to 0.1
	N_field	0.4 to 0.5
	φ	1.6180339887498948
GSA	G_0	100
	α	20
	K_0	50

3.5.1 Archimedes optimization algorithm

The AOA [2573] is a recently developed population-based metaheuristic algorithm, motivated by the intriguing rule of physics known as Archimedes' Principle. It simulates the idea that an object's buoyant force, whether fully or partially immersed in a fluid, is directly proportional to the weight of the displaced fluid. Several advantages of AOA include its straightforward interface, minimal tuning parameters and ease of implementation. However, there are still some drawbacks, like a drop-trap local optimum, lack of diversity in the search exploring capacity, and incapacity to strike a balance between exploration and exploitation [258, 259]. Mathematically, its application steps are explained as follows:

Stage 1: Initialization

Use Eq. (3.150) to set all object locations for initialization.

$$P_i = lb_i + r_i\,(ub_i - lb_i); (i = 1, 2, \ldots, N) \tag{3.150}$$

where P_i is the i^{th} object in a population consisting of N objects, lb_i is the lower bound and ub_i is the upper bound of the search space.

For each i^{th} object, initialize volume (vol) and density (den) using Eq. (3.151).

$$den_i = r_i,\ vol_i = r_i \tag{3.151}$$

where r_i is a D-dimensional vector to randomly generate a number between [0, 1].

Finally, for i^{th} object, initialize acceleration (*acc*) using Eq. (3.152).

$$acc_i = lb_i + r_i\,(ub_i - lb_i) \tag{3.152}$$

Step 2: Update *den* and *vol*

For iteration (t + 1) and object i, *den* and *vol* are updated using the following equations:

$$den_i^{t+1} = den_i^t + r_i\,(den_{best} - den_i^t)$$

$$vol_i^{t+1} = vol_i^t + r_i\,(vol_{best} - vol_i^t) \tag{3.153}$$

where vol_{best} and den_{best} are the volume and density related to the best object derived so far.

Step 3: Determination of transfer operator and density factor

Initially, there are object collisions and then, over time, the objects attempt to achieve an equilibrium condition. With the aid of transfer operator (*TF*), it is accomplished in AOA by changing the search process from exploration to exploitation.

$$TF = \left(\frac{t - IT}{IT}\right) \tag{3.154}$$

where *TF* progressively increases with time until reaching 1.

In a similar way, density-reducing factor (*df*) also helps AOA in switching from global to local searches.

$$df^{t+1} = \exp\left(\frac{IT-t}{IT}\right) - \left(\frac{t}{IT}\right) \tag{3.155}$$

where df^{t+1} decreases with time enabling the algorithm to converge in the previously identified potential region. Note that, AOA would maintain a balance between exploration and exploitation if the value of this parameter is properly selected.

Step 4.1: Exploration

If there is an object collision and *TF* < 0.5, choose a random material (*mr*) and use Eq. (3.156) to update its acceleration for iteration (*t* + 1).

$$acc_i^{t+1} = \frac{den_{mr} + vol_{mr} \times acc_{mr}}{den_i^{t+1} \times vol_i^{t+1}} \tag{3.156}$$

where den_i is the density, vol_i is the volume and acc_i is the acceleration of i^{th} object. On the other hand, den_{mr} is the density, vol_{mr} is the volume and acc_{mr} is the acceleration of the random material. A value of $TF \leq 0.5$ confirms exploration during 1/3 of the iterations, which is an important point to note. Exploration-exploitation behavior would shift when a value other than 0.5 is considered.

Step 4.2: Exploitation

A value of *TF* > 0.5 indicates that there is no object collision, and use Eq. (3.157) to modify acceleration of the object for iteration (*t* + 1).

$$acc_i^{t+1} = \frac{den_{best} + vol_{best} \times acc_{best}}{den_i^{t+1} \times vol_i^{t+1}} \tag{3.157}$$

where acc_{best}, den_{best} and vol_{best} are the acceleration, density and volume of the best object, respectively.

Step 4.3: Normalize acceleration

Determine the change percentage for normalizing acceleration based on Eq. (3.158):

$$acc_{i-norm}^{t+1} = u \times \frac{acc_i^{t+1} - \min(acc)}{\max(acc) - \min(acc)} + l \tag{3.158}$$

where u = 0.9 and l = 0.1 are the range of normalization, and acc_{i-norm}^{t+1} calculates the percentage of step that each agent would alter. A high acceleration value indicates that the object is far away from the global optimum in the exploration phase; otherwise, it is in the exploitation phase. This process defines how the search transformation from exploration to exploitation phase takes place.

Step 5: Update position

Employ Eq. (3.159) to determine the i^{th} object's position for the next iteration (t + 1) if $TF \leq 0.5$ (exploration phase):

$$x_i^{t+1} = x_i^t + C_1 \times r_i \times acc_{i-norm}^{t+1} \times df \times (x_{r_i} - x_i^t) \quad (3.159)$$

where C_1 is a constant having value 2. Otherwise, adopt Eq. (160) to update the position of the object, if $TF > 0.5$ (exploitation phase).

$$x_i^{t+1} = x_{best}^t + F \times C_2 \times r_i \times acc_{i-norm}^{t+1} \times df \times (T \times x_{best} - x_i^t) \quad (3.160)$$

where C_2 is a constant with value equal to 6. On the other hand, $T = C_3 \times TF$ would increase with time and it is directly proportional to the transfer operator. Initially, the value of T deviates a specific percentage from the optimal position as it changes over time in the range [C_3×0.3, 1] (C_3 is a constant having value 2). F is the flag to change the direction of motion, estimated using Eq. (3.161):

$$F = \begin{cases} +1 \; if \; P \leq 0.5 \\ -1 \; if \; P > 0.5 \end{cases} \quad (3.161)$$

where $P = 2 \times r_i - C_4$ (C_4 is a constant having value 0.5).

Step 6: Evaluation

Utilizing the objective function f, assess each object and remember the best solution identified so far, as $vol_{best}, acc_{best}, den_{best}$ and x_{best}.

Algorithm 3.5.1: Pseudo-code of AOA

```
Initialize population with Pi, den, vol, IT, C1, C2, C3 and C4
Evaluate initial population
Set t = 1
while t ≤ IT do
    for each object i do
Update deni and voli of each object using Eq. (3.153)
        Update TF and df using Eqs. (3.154) and (3.155), respectively.
        if TF ≤ 0.5 (Exploration phase) then
            Update acc_i^{t+1} using Eq. (3.156) and acc_{i-norm}^{t+1} applying Eq. (3.158)
            Update position using Eq. (3.159)
        else (Exploitation phase)
            Update acc_i^{t+1} using Eq. (3.157) and acc_{i-norm}^{t+1} based on Eq. (3.158)
            Update flag F using Eq. (3.161)
            Update position employing Eq. (3.160)
        end if
    end for
    Evaluate each object and choose the one having the best fitness value
    Set t = t + 1
end while
Return object with best fitness value.
```

3.5.2 Atom search optimization

Being a novel physics-inspired metaheuristic approach, ASO was proposed by Zhao et al. [260], developed based on the principle of fundamental molecular dynamics. It states that every atom in the search space either attracts or repels every other atom; as a result, the lower mass atoms move in the direction of heavier mass atoms. Higher mass makes an atom accelerate less quickly, making it more aggressive in its hunt for better solutions in the local search space. Furthermore, low mass atoms accelerate faster, covering a wider region to discover new approximate locations in the search space [261]. It has been intended to address a variety of complex multi-modal optimization problems. In the atomic motion model, atoms interact by Lennard-Jones potential-derived interaction forces and bond-length potential-derived constraint forces. These are mathematically modelled and mimicked in ASO algorithm [260]. It can increase accuracy of complex optimization problems, having the advantages of quick computing time and excellent optimization effect [262]. A few limitations include weak exploitation, possibility of being trapped in local optima, and improper exploration-exploitation balance [263]. Let, an unconstrained optimization problem be stated as shown below:

Minimize $f(x), \quad x = (x^1, \ldots, x^D)$ (3.162)

$Lb \leq x \leq Ub, \;\; Lb = [lb^1, \ldots, lb^D], \;\; Ub = [ub^1, \ldots, ub^D]$ (3.163)

where $x^d (d = 1, \ldots, D)$ denotes the d^{th} element of the search space, lb^D and ub^D are the d^{th} components of the lower and upper limits, respectively, and *D* represents dimension of the search space.

To address the above problem, it is assumed that there are *N* atoms in the population and x_i indicates the location of i^{th} atom.

$x_i = \left[x_i^1, \ldots, x_i^D\right], \qquad i = 1, \ldots . N$ (3.164)

During the early stages of ASO, attraction or repulsion between all the atoms occurs. A more complete investigation of the solution space is made possible by the repulsion, which can lessen the atom's over-concentration and early algorithm convergence. As iterations go on, the attraction strengthens and the repulsion drops, suggesting that exploitation is increasing and exploration is decreasing. In the last stages, each atom is attracted by the others, ensuring that the algorithm has a high potential for exploitation. Its application steps are mentioned as below:

Step 1: Initialize a set of atoms *X* and their velocity *v*

Step 2: Using Eqs. (3.165) and (3.166), calculate the total force F_i and constraint force G_i.

$F_i^d(t) = \sum_{j \in Kbest} r_{i_j} F_{ij}^d(t)$ (3.165)

$G_i^d(t) = -\lambda(t) \nabla \theta_i^d(t) = -2\lambda(t)(x_i^d(t) - x_{best}^d(t))$ (3.166)

where $F_{ij}^d(t)$ is the total force acting on i^{th} atom from j^{th} atom in d^{th} dimension at t^{th} iteration, $rand_j$ is a random number in the range of [0,1], $\lambda(t)$ is the Lagrangian multiplier and $\theta_i(t)$ is the constraint of i^{th} atom.

Then, substituting $2\,\lambda \rightarrow \lambda$, G_i can be redefined as:

$G_i^d(t) = \lambda(t)(x_{best}^d(t) - x_i^d(t))$ (3.167)

The Lagrangian multiplier can be defined using the following equation:

$$\lambda(t) = \delta e^{-\frac{20t}{IT}} \tag{3.168}$$

where δ is the multiplier weight.

Step 3: With F_i and G_i, calculate acceleration of i^{th} atom at time t using Eq. (3.169)

$$a_i^d(t) = \frac{F_i^d(t)}{m_i^d(t)} + \frac{G_i^d(t)}{m_i^d(t)} = -\propto \left(1 - \frac{t-1}{IT}\right)^3 e^{-\frac{20t}{IT}} \sum_{j \in Kbest} \frac{r_{ij}[2 \times \left(h_{ij}(t)\right)^{13} - \left(h_{ij}\right)^7}{m_i(t)}$$

$$\frac{(x_j^d(t) - x_i^d(t))}{\|x_i(t), x_j(t)\|_2} + \delta e^{-\frac{20t}{IT}} \frac{x_{best}^d(t) - x_i^d(t)}{m_i(t)} \tag{3.169}$$

where $m_i(t)$ is the mass of i^{th} atom at t^{th} iteration, $m_i^d(t)$ is the mass of i^{th} atom in d^{th} dimension at t^{th} iteration, $h_{ij}(t)$ is the limit of attraction and repulsion between i^{th} and j^{th} atoms, and $\propto$ is the depth weight.

Step 4: Calculate mass of i^{th} atom using Eqs. (3.170) and (3.171)

$$M_i(t) = e^{-\frac{Fit_i(t) - Fit_{best}(t)}{Fit_{worst}(t) - Fit_{best}(t)}} \tag{3.170}$$

$$m_i(t) = \frac{M_i(t)}{\sum_{j=1}^{N} M_j(t)} \tag{3.171}$$

where $M_i(t)$ is the mass of i^{th} atom, $M_j(t)$ is the mass of j^{th} atom, $Fit_{best}(t)$ and $Fit_{worst}(t)$ are the fitness values of the best and worst atoms at t^{th} iteration, and $Fit_i(t)$ is the fitness function of i^{th} atom at t^{th} iteration. $Fit_{best}(t)$ and $Fit_{worst}(t)$ are expressed applying the following equations:

$$Fit_{best}(t) = \min_{i \in \{1,2,\ldots,N\}} Fit_i(t) \tag{3.172}$$

$$Fit_{worst}(t) = \max_{i \in \{1,2,\ldots,N\}} Fit_i(t) \tag{3.173}$$

Step 5: Using Eqs. (3.174) and (3.175), velocity and position of the atoms are continuously updated

$$v_i^d(t+1) = r_i^d v_i^d(t) + a_i^d(t) \tag{3.174}$$

$$x_i^d(t+1) = x_i^d(t) + v_i^d(t+1) \tag{3.175}$$

Step 6: Each atom must interact with as many atoms with better fitness values as its K neighbours to improve exploration in the first step of iterations. The atoms need to interact with as few atoms with superior fitness values as its K neighbors to maximize exploitation in the final step of iterations. As a result, K gradually decreases with the number of iterations over time. The value of K can be computed based on Eq. (3.176):

$$K(t) = N - (N-2) \times \sqrt{\frac{t}{IT}} \tag{3.176}$$

Algorithm 3.5.2: Pseudo-code of ASO

```
Initialize randomly a set of atoms X with velocity v, and Fit_best = Inf
while the stop criterion is not satisfied do
    for each object X_i do
        Compute the fitness value Fit_i
        if Fit_i < Fit_best then
```

$Fit_{best} = Fit_i$
$X_{best} = X_i$
end if
Calculate F_i and G_i using Eqs. (3.165) and (3.167), respectively
Calculate $a_i^d(t)$ based on Eq. (3.169)
Calculate mass applying Eqs. (3.170) and (3.171)
Calculate its *K* neighbours using Eq. (3.176)
Update $v_i^d(t+1)$ employing Eq. (3.174)
Update $x_i^d(t+1)$ using Eq. (3.175)
end for
end while
Find the best solution X_{best}

3.5.3 Nuclear reaction optimization

The NRO, presented by Wei et al. [264], is a new and powerful physics-based algorithm. It mimics the two stages of a nuclear reaction, i.e. nuclear fusion (NFu) and nuclear fission (NFi). For proper exploration and exploitation in NFi phase, respectively, the Gaussian walk and differential evolution operators between the nucleus and neutron are utilized. In NFu phase, which comprises ionization and fusion phases, variations of differential evolution operator are employed for exploration. Furthermore, in every stage of NFu phase, random searching is employed to break out from the local optima using variations on the Levy flight algorithm. When the two stages are combined, its capabilities for exploration and exploitation is balanced [264]. It has the following steps:

Phase 1: Initialize population of the nuclei

In NRO algorithm, a uniformly distributed population of *N* nuclei is randomly formulated, and each nucleus individual, X_i ($i \in \{1,2,\ldots,N\}$) is a *D*-dimensional vector. Equation (3.177) is employed to initialize i^{th} nucleus in the population.

$$X_{i,d} = lb_d + r_i(ub_d - lb_d) \quad (3.177)$$

where X_i denotes the state of i^{th} nucleus in a closed volume, ub_d is the upper bound and lb_d is the lower bound of d^{th} variable in the search space.

Phase 2: Fission phase

Nuclear fission is converted into a mathematical model in NFi phase using a Gaussian walk to generate new nuclei after nuclear fission. It is believed that heated neutrons can be produced by nuclear fusion of two distinct random nuclei in the cycle between nuclear fission and fusion based on Eq. (3.178).

$$Ne_i = \frac{(X_i + X_j)}{2} \quad (3.178)$$

where Ne_i is the i^{th} heated neutron, and X_i and X_j denote i^{th} nucleus and j^{th} nucleus, respectively.

Step 1: Two kinds of fission products occur when the odd nuclei undergo fission by absorbing hot neutrons, i.e. primary fission and secondary fission (SF). The process of developing an SF product can be simulated using Eq. (3.179) when $rand \leq P_\beta$ is true, where P_β is the probability of β decay.

$$X_i^{Fi} = Gaussian(X_{best}, \sigma_1) + (r_i \times X_{best} - P_{ne} \times Ne_i) \quad (3.179)$$

$$\sigma_1 = \left[\frac{\log(t)}{t}\right] \times |X_i - X_{best}| \quad (3.180)$$

$$P_{ne} = round(r_i + 1) \quad (3.181)$$

where X_i^{Fi} is i^{th} fission product nucleus, X_{best} is the present best nucleus and P_{ne} is the mutation factor.

Step 2: According to an analysis of primary fission generation by an odd nucleus after absorbing a heated neutron, primary fission can be modelled as follows:

$$X_i^{Fi} = Gaussian(X_i, \sigma_2) + (randn \times X_{best} - P_{ne} \times Ne_i) \quad (3.182)$$

$$\sigma_2 = \left[\frac{\log(t)}{t}\right] \times |X_r - X_{best}| \quad (3.183)$$

$$P_{ne} = round(r_i + 2) \quad (3.184)$$

where X_r is randomly selected from the nuclei population that is different from i^{th} index of the current nucleus X_i, and *randn* denotes a normally distributed random number.

Step 3: The current nucleus's information is preserved and originates from the Gaussian walk, represented by Eq. (3.185).

$$X_i^{Fi} = Gaussian(X_i, \sigma_2) \quad (3.185)$$

If the general fission process indicates that $(\text{primary fission})_i$ is a likely possibility, then NF_i phase assumes that there are two fission nuclei lighter than the initial heavy nucleus. Another possibility, P_β can determine whether the fission would undergo β decay.

Phase 3: Nuclear fusion

Depending on the reaction parameters, a nuclear fusion process can be classified as hot or cold. Hot nuclear fusion often happens, when the nuclei are heated to the plasma state and the resulting heavier nucleus can replace the original light nuclei. The NFu phase, which is further subdivided into ionization and fusion phases, mainly contributes to exploration with regard to NFi phase for exploitation. The NRO algorithm adopts the following steps:

Step 1: Ionization

Initially, all nuclei are ordered based on their fitness ratings for the ionization stage. A uniform distribution-based probability value is assigned to i^{th} nucleus as follows:

$$Pa_i = \frac{rank(fitX_i^{Fi})}{N} \quad (3.186)$$

where $fitX_i^{Fi}$ is the fitness value of X_i^{Fi}, $rank(fitX_i^{Fi})$ is the rank of i^{th} nucleus X_i^{Fi} among the other nuclei in the population and N is the total number of nuclei.

For the current nucleus X_i^{Fi} in the fusion reaction population, if $Pa_i < rand$, d^{th} variable of X_i^{Fi} is ionized using Eqs. (3.187) and (3.188).

$$X_{i,d}^{Ion} = X_{r_1,d}^{Fi} + r_i\left(X_{r_2,d}^{Fi} - X_{i,d}^{Fi}\right), r_i \leq 0.5 \quad (3.187)$$

$$X_{i,d}^{Ion} = X_{r_1,d}^{Fi} - r_i\left(X_{r_2,d}^{Fi} - X_{i,d}^{Fi}\right), r_i > 0.5 \quad (3.188)$$

where $X_{i,d}^{Ion}$ denotes d^{th} variable of i^{th} ion after ionization, and $X_{i,d}^{Fi}$, $X_{r_1,d}^{Fi}$ and $X_{r_2,d}^{Fi}$ signify d^{th} variables of i^{th}, r_1^{th} and r_2^{th} fission nuclei, respectively.

If $Pa_i \leq rand$, more stable fission nuclei remain unionized by fission energy and undergo minimal or no change. To enhance the exploitation performance, $X_{i,d}^{Fi}$ is modified by considering a narrow disturbance based on Eq. (3.189):

$$X_{i,d}^{Ion} = X_{i,d}^{Fi} + round(r_i) \times r_i(X_{worst,d}^{Fi} - X_{best,d}^{Fi}) \quad (3.189)$$

where $X_{worst,d}^{Fi}$ is d^{th} variable of the worst fission product nucleus.

Step 2: Fusion

For improved exploration, this process adopts variations of difference operators, much like the ionization process. Prior to the fusion stage, all of the ions obtained during ionization are ranked applying Eq. (3.190), similar to Eq. (3.186).

$$Pc_i = \frac{rank(fitX_i^{Ion})}{N} \quad (3.190)$$

where Pc_i is the probability value of i^{th} ion and $rank(fitX_i^{Ion})$ is treated as the rank of i^{th} ion in the population.

The strong nuclear force causes the ions of two light nuclei to overcome the Coulomb repulsive force and collide for fusion. Various strategies inspired by variations of difference operators are applied to simulate fusion and collision to enhance the quality of exploration. The following equation expresses how the fusion product takes place:

$$X_i^{Fu} = X_i^{Ion} + r_i\left(X_{r_1}^{Ion} - X_{best}^{Ion}\right) + r_i\left(X_{r_2}^{Ion} - X_{best}^{Ion}\right) - e^{-norm\left(X_{r_1}^{Ion} - X_{r_2}^{Ion}\right)} \times \left(X_{r_1}^{Ion} - X_{r_2}^{Ion}\right) \quad (3.191)$$

where X_i^{Fu} is i^{th} fusion product, $X_{r_1}^{Ion}$ and $X_{r_2}^{Ion}$ respectively denote r_1^{th} and r_2^{th} ions, *norm* is a population size-based norm, and r_1 and r_2 are not the same. In Eq. (3.191), X_i^{Ion} serves as the foundation for X_i^{Fu}, the difference between $X_{r_1}^{Ion}$ and X_{best}^{Ion} is considered in the second item, the difference between $X_{r_2}^{Ion}$ and X_{best}^{Ion} in the third item provides information on another part of fusion, and the difference between $X_{r_1}^{Ion}$ and $X_{r_2}^{Ion}$ in the last item denotes capability of the ions to overcome the Coulomb repulsive force.

The Coulomb force plays a crucial role in controlling the characteristic change of ions when they are unable to resist it and fail to fuse together. If fusion does not take place, the Coulomb force may repulse the opposite movement or reduce the approaching velocity. The corresponding mathematical models are shown as below:

$$X_i^{Fu} = X_i^{Ion} - 0.5 \times \left[\sin(2\pi \times freq \times t + \pi) \times \frac{IT - t}{IT} + 1\right]\left(X_{r_1}^{Ion} - X_{r_2}^{Ion}\right), r_i > 0.5 \quad (3.192)$$

$$X_i^{Fu} = X_i^{Ion} - 0.5 \times \left[\sin(2\pi \times freq \times t + \pi) \times \frac{t}{IT} + 1\right]\left(X_{r_1}^{Ion} - X_{r_2}^{Ion}\right), r_i \leq 0.5 \quad (3.193)$$

where $freq$ is the sinusoidal function frequency.

Step 3: Levy distribution strategy

The Levy distribution approach is applied in NFu phase to assist the solution in breaking out of the local optimum. Owing to Levy flight's benefit, a thorough search for Levy flight is adopted during fusion and ionization stages. In Eqs. (3.187) and (3.188) of the ionization stage, if $X_{r_2,d}^{Fi} = X_{i,d}^{Fi}$ is true, the

Levy flight strategy is defined using Eq. (3.195).

$$X_{i,d}^{Ion} = X_{i,d}^{Fi} + \left(\alpha \otimes Levy(\beta)\right)_d (X_{i,d}^{Fi} - X_{best,d}^{Fi}) \tag{3.194}$$

where α is a scale factor whose value depends on the scales of the problem solved, $\otimes$ is the entry-wise multiplications, and Levy(β) is the Levy flight step size, described as below:

$$Levy(\beta) = \frac{\mu}{|v|^{\frac{1}{\beta}}} \tag{3.195}$$

where μ and v are obtained from the normal distribution $N\left(0, \sigma_\mu^2\right)$ and $N(0, \sigma_v^2)$, respectively. σ_μ and σ_v are defined as follows:

$$\sigma_\mu = \left[\frac{\Gamma(1+\beta)\sin(\frac{\pi\beta}{2})}{\Gamma[\frac{(1+\beta)}{2}]\beta \times 2^{(\beta-1)/2}}\right]^{1/\beta}, \sigma_v = 1. \tag{3.196}$$

where Γ is a Gamma function, and $\alpha = 0.01$ and $\beta = 1.5$.

If $X_{r_2,d}^{Fi} \neq X_{\mathrm{i},d}^{Fi}$, Eqs. (3.188) and (3.189) are employed. During later period of iteration, due to decreasing search step, it is possible that $X_{\mathrm{i},d}^{Fi} = X_{\mathrm{best},d}^{Fi}$. Therefore, another Levy flight is proposed in Eq. (3.197).

$$X_{i,d}^{Ion} = X_{i,d}^{Fi} + \left(\alpha \otimes Levy(\beta)\right)_d (ub_d - lb_d) \tag{3.197}$$

For ionization stage in Eq. (3.189), to avoid getting stuck in local optimum, if $X_{worst,d}^{Fi} = X_{best,d}^{Fi}$ is true, Eq. (3.197) is utilized; otherwise, Eq. (3.189) is implemented.

In the fusion stage of NFu phase, if the condition of $X_{r_1}^{Ion} = X_{r_2}^{Ion}$ is likely to appear, the Levy flight strategy is proposed as shown below:

$$X_i^{Fu} = X_i^{Ion} + \alpha \otimes Levy(\beta) \otimes (X_i^{Ion} - X_{best}^{Ion}) \tag{3.198}$$

Algorithm 3.5.3: Pseudo-code of NRO

```
Setting: lb, ub, N, D, IT
Initialize the nuclei population using Eq. (3.177) and estimate the fitness function;
while (t < IT) do
    t = t + 1;
        //NFi phase
          for i = 1 to N do
   Calculate neutron vector by Eq. (3.178);
              Update population of fission products using Eq. (3.179), (3.182) or (3.185);
              Verify the boundary and evaluate the fitness function for X_i^Fi;
              Update X_i^t with X_i^Fi;
          end for
        // NFi phase
            /* Ionization stage*/
 Compute Pa applying Eq. (3.186);
               for i = 1 to N do
                 for d = 1 to D do
                    Execute the Levy flight strategies by Eqs. (3.194), (3.197) and (3.198), and
                    value of Pa;
```

Update states of ions using Eq. (3.187), (3.188) or (3.189);
end for
Verify the boundary and evaluate fitness function for ion X_i^{Ion}
Update $X_i^{Fi}$with X_i^{Ion};
end for
/* Fusion stage */
Calculate *Pc* using Eq. (3.190);
for *i* = 1 to *N* do
Execute the Levy flight strategy by Eq. (3.198) and value of *Pc*;
Update population of fusion products using Eqs. (3.191), (3.192) and (3.193);
Verify the boundary and evaluate the fitness function for X_i^{Fu};
Update X_i^{Ion} with X_i^{Fu};
end for
end while
Output the best solution.

3.5.4 Electromagnetic field optimization

EFO is also a physics-based metaheuristic algorithm, proposed by Abedinpourshotorban et al. [265]. It basically utilizes a ratio found in nature called the golden ratio and is motivated by the behavior of electromagnets with varying polarities. A potential solution in EFO is an electromagnetic particle composed of electromagnets, with the number of electromagnets equal to the number of variables in the optimization problem. It operates on a population basis and the global optima is reached by the particles in this algorithm, which divides the population into three fields, i.e. positive, negative and neutral. The electromagnets in each of these fields experience attraction and repulsion forces. Particles can converge quickly and effectively when the ratio of attraction to repulsion forces is determined by the golden ratio. It has several benefits, such as good search space, manageable programming and implementation, efficiency in obtaining global optima, and acceptable convergence and execution time [266]. It has the following application steps:

Step 1: Initialization

Equation (3.199) is employed to generate population of the electromagnetic particles (EMPs). Each particle is randomly generated between the upper (*UB*) and lower (LB) bounds of the search space, after which the fitness values of all the EMPs are evaluated and they are sorted according to their fitness values in descending order. In EFO, an EMP created by *N_var* is referred to as a solution, where *N_var* is the number of electromagnets of EMP (problem variables).

$$EMP_j^i = lb + r_i(UB - LB) \tag{3.199}$$

where *j* = 1,…,*N_var*, and *i* = 1,…,N_{emp}.

Step 2: Classification

In this stage, the EMPs are classified into three groups, i.e. positive field (*P_field*), negative field (*N_field*) and neutral field (*Nt_field*), depending on their polarities. Here, positive field contains a portion of the population having the best fitness, negative field consists of a portion of the population

with the worst fitness, and neutral field contains a small portion of the remaining EMPs. In Eqs. (3.200)-(3.202), the classification process of the EMPs is explained.

$$P_j = r_i\left(1, \lfloor N_{emp} \times P\_field \rfloor\right) \tag{3.200}$$

$$N_j = r_i\left(\lfloor N_{emp}(1 - N\_field), N_{emp} \rfloor\right) \tag{3.201}$$

$$Nt_j = r_i\left(\lceil N_{emp} \times P\_field \rceil, \lceil N_{emp}(1 - N\_field) \rceil\right) \tag{3.202}$$

where N_{emp} is the number of EMPs in the population, and P_j, N_j and Nt_j are the index of a random particle from the positive field, negative field and neutral field, respectively.

Step 3: New particle candidate generation

In this step, a new EMP is generated and its fitness is assessed. The new EMP is introduced into the population if its fitness exceeds that of the worst existing EMP. Based on the inserted EMP's position in the sorted population, the corresponding group and polarity are established. Additionally, to maintain an equivalent amount of EMPs, the worst EMP is removed from the population.

The probability of choosing electromagnets of the generated particle from the positive field (*Ps_rate*) is compared against a randomly generated number. The new electromagnet is created using Eq. (3.203), if this random number is smaller than *Ps_rate*.

$$EMP_j^{New} = EMP_j^{P_j} \tag{3.203}$$

Otherwise, the new electromagnet is created based on Eq. (3.204).

$$EMP_j^{New} = EMP_j^{Nt_j} + \left(\varphi \times r_i \times \left(EMP_j^{P_j} - EMP_j^{Nt_j}\right)\right) - \left(r_i \times \left(EMP_j^{N_j} - EMP_j^{Nt_j}\right)\right) \tag{3.204}$$

where φ is the golden ratio.

Step 4: Randomization

The randomization step involves changing only one electromagnet of the generated EMP with a random electromagnet within its boundary as follows:

$$EMP_{RI}^{new} = lb + r_i(\text{UB} - \text{LB}) \quad \text{if } r_i < R\_rate \tag{3.205}$$

where *RI* indicates an integer counter to denote the index of the electromagnet selected for randomization. Its value is initially set as 1 or any value between [1, *N_var*], and is increased by 1 after each randomization. When its value reaches its upper limit *N_var*, it is again set to 1. Here, *R_rate* denotes the probability of changing one electromagnet by another.

The above explanations make clear that many EMPs collaborate to create a new EMP which is a crucial aspect of EFO algorithm [267, 268].

Algorithm 3.5.4: Pseudo-code of EFO

Input: *N_var*, N_{emp}, *P_field*, *N_field*, *R_rate*, *Ps_rate*, φ
Initialization of the field of a random electromagnet
for i = 1 to N_{emp} do
    for j = 1 to *N_var* do
        calculate EMP using Eq. (3.199)
            $EMP_j^{New} = EMP_j^i$

end for

$fit^i = f(EMP^{New})$

end for

while

for i = 1 to *N_var* do

Calculate P_j, N_j and Nt_j using Eqs. (3.200), (3.201) and (3.202), respectively

if (r_i < *Ps_rate*) then

Use Eq. (3.203) to generate the new electromagnet

Else

Use Eq. (3.204) to generate the new electromagnet

end if

If ($EMP_i^{New} > ub \ or \ EMP_i^{New} < lb$ then

$EMP_i^{new} = lb + r_i(UB - LB)$

end if

end for

if (r_i < *R_rate*) then

$EMP_{RI}^{new} = lb + r_i(UB - LB)$

if (*RI* > *N_var*) then

RI = 1

end if

end if

$New\_fit = f(EMP^{New})$

if (New_fit < worst(fit)) then

Insert in the sorted population based on fitness (EMP^{New})

end if

end while

return

3.5.5 Gravitational search algorithm

Based on the laws of physics (inspired by the ideas of mass and gravity), Rashedi et al. [269] developed GSA as a revolutionary nature-inspired optimization technique. In this algorithm, each object in the system attracts every other object through the force of gravity, and the agents are assumed as distinct objects with certain masses. An objective function is considered to calculate the mass of each agent, and each agent's position provides a potential solution to the problem [269]. Attraction between the objects causes a global movement towards the heavier mass objects. The heavier mass objects correspond to good solutions, move slowly than the lighter mass objects, ensuring exploitation step of this algorithm. Numerous GSA variations have been proposed till date, and most of them have been inspired by the ideas connected to gravity, like astronomy and relativity [270]. GSA has a number of advantages, including its suitability for nonlinear optimization problems, ease of implementation, high precision results and capacity for adaptive learning. However, it has also some drawbacks, like inability to search quickly at the last stages of iteration, high computational time and cost, and tendency to become stuck in local minima [57]. The procedural steps of this algorithm are stated as below:

Define the position of i^{th} agent in the search space considering a system with *N* agents.

$$X_i = \left(X_i^1, \ldots, X_i^d, \ldots, X_i^n\right) \text{ for } i = 1, 2, \ldots, N \quad (3.206)$$

where X_i^d is the position of i^{th} agent in d^{th} dimension and n is the problem's dimension.

All the agents are first randomly placed in the search space. The gravitational force exerted on mass i from mass j at a given time t_m, over all iterations, is defined as follows:

$$F_{ij}^d(t_m) = G(t_m)\frac{M_{pi}(t_m) \times M_{aj}(t_m)}{R_{ij}(t_m) + \varepsilon}\left(X_j^d(t_m) - X_i^d(t_m)\right) \quad (3.207)$$

where $G(t_m)$ is a gravitational constant at time t_m, M_{pi} is the passive gravitational mass associated with i^{th} agent, M_{aj} is the active gravitational mass associated with j^{th} agent, $R_{ij}(t_m)$ is the Euclidian distance between two agents i and j, and ε is a small constant.

The following two equations are employed to determine the Euclidian distance and gravitational constant between two agents, i and j:

$$G(t_m) = G_0 \times \exp\left(-\alpha \times \frac{t}{IT}\right) \quad (3.208)$$

$$R_{ij}(t_m) = \left\|X_i(t_m), X_j(t_m)\right\| \quad (3.209)$$

where G_0 is the initial gravitational constant and α is the coefficient of decrease.

In the problem space, the total force acting on agent i in d-dimension is the randomly weighted sum of d^{th} components and is calculated using the following equation:

$$F_i^d(t_m) = \sum_{j=1, j\neq i}^{N} r_{i_j} F_{ij}^d(t_m) \quad (3.210)$$

where r_{i_j} is a random number between [0,1].

To improve the performance of GSA, only the K_{best} agents would attract the others in order to control exploration and exploitation. K_{best} is a function of time, starting at a value of K_0 and decreasing over time. It signifies that all the agents apply force at the beginning, but with time, K_{best} decreases linearly, and at the end, only one agent would exert force on the others. As a result, Eq. (3.210) can be changed as follows:

$$F_i^d(t_m) = \sum_{j\in K_{best}, j\neq i} r_{i_j} F_{ij}^d(t_m) \quad (3.211)$$

where K_{best} is the initial K agent with the best fitness value and highest mass.

This algorithm also considers Newton's law of motion, which states that a mass's acceleration is proportional to the applied force and inverse to its mass. As a result, the acceleration of each agent can be determined as follows:

$$a_i^d(t_m) = \frac{F_i^d(t_m)}{M_{ii}(t_m)} \quad (3.212)$$

where a is the acceleration of agent i at time t_m and M_{ii} is the inertial mass of agent i.

The agent's velocity and position are respectively calculated using Eqs. (3.213) and (3.214):

$$v_i^d(t_m + 1) = r_i \times v_i^d(t_m) + a_i^d(t_m) \quad (3.213)$$

$$X_i^d(t_m + 1) = X_i^d(t_m) + v_i^d(t_m + 1) \quad (3.214)$$

The fitness evaluation makes it simple to calculate the masses owing to inertia and gravity. A heavier mass indicates higher efficiency of an agent. As a result, better agents move more slowly and have

higher attraction. The mass values are obtained applying the map of fitness, assuming that the gravitational and inertia masses are equal. The gravitational and inertial masses can be updated using the following expressions:

$$M_{ai} = M_{pi} = M_{ii} = M_i, \quad i = 1, 2, \ldots\ldots, N. \tag{3.215}$$

$$m_i(t_m) = \frac{fit_i(t_m) - worst(t_m)}{best(t_m) - worst(t_m)} \tag{3.216}$$

$$M_i(t_m) = \frac{m_i(t_m)}{\sum_{j=1}^{N} m_j(t_m)} \tag{3.217}$$

where $fit_i(t_m)$ is the fitness value of i^{th} agent at time t_m. On the other hand, $worst(t_m)$ is the weakest agent at time t_m and $best(t_m)$ is the fittest agent at time t_m, which can be defined as follows (for minimization problem):

$$best(t_m) = \min_{j \in \{1,\ldots,N\}} fit_j(t_m), \qquad worst(t_m) = \max_{j \in \{1,\ldots,N\}} fit_j(t_m) \tag{3.218}$$

For maximization problem

$$best(t_m) = \max_{j \in \{1,\ldots,N\}} fit_j(t_m), \qquad worst(t_m) = \min_{j \in \{1,\ldots,N\}} fit_j(t_m) \tag{3.219}$$

Algorithm 3.5.5: Pseudo-code of GSA

Input**:** Let N search agents and a gravitational constant G_0
Initialization**:** Generate a random population of objects using Eq. (3.206)
while
    Evaluate the fitness of all agents
    Update $G(t_m)$ by Eq. (3.208)
    Calculate $M_i(t_m)$ using Eq. (3.217)
    Calculate $F_i^d(t_m)$ and $a_i^d(t_m)$ applying Eqs. (3.211) and (3.212)
    Update v_i^d and X_i^d using Eqs. (3.213) and (3.214)
end while
Return g_{best}

4. OPTIMIZATION OF A TURNING PROCESS USING HUMAN-INSPIRED METAHEURISTIC ALGORITHMS

The optimization of machining operations has gained significant attention due to its potential to enhance productivity, reduce costs and improve surface quality. This research work focuses on optimization of the turning operation on AISI 6061-T6 aluminum using five human-inspired metaheuristic algorithms, i.e. TLBO, SAR, TOA, HCO and QSA. Comparative studies are also performed to evaluate their performance with reference to accuracy and variability of the solutions, rate of convergence and computation time. The study utilizes the past experimental data of Camposeco-Negrete [271] to optimize three turning parameters, i.e. SS, FR and DOC. The procedural steps for parametric optimization of the said turning operation are depicted in Figure 4.1, through a flow diagram. To intensify the comparison, the results of Friedman's mean rank test and Wilcoxon rank sum test are also considered. The corresponding Pareto optimal front for the best human-inspired metaheuristic is developed to obtain the distinct parametric settings along with development of the related parallel plot to simplify identification of the optimal response values. Additionally, the corresponding SP and HV values are examined to pick out the best performing human-inspired algorithm. While the 'no free lunch' theorem [191] implies that no method can successfully manage all the optimization problems, it is advisable to gauge performance of a cluster of metaheuristics proposed on the basis of similar guiding principle and identify the one excelling over its competitors with reference to some specific evaluation criteria.

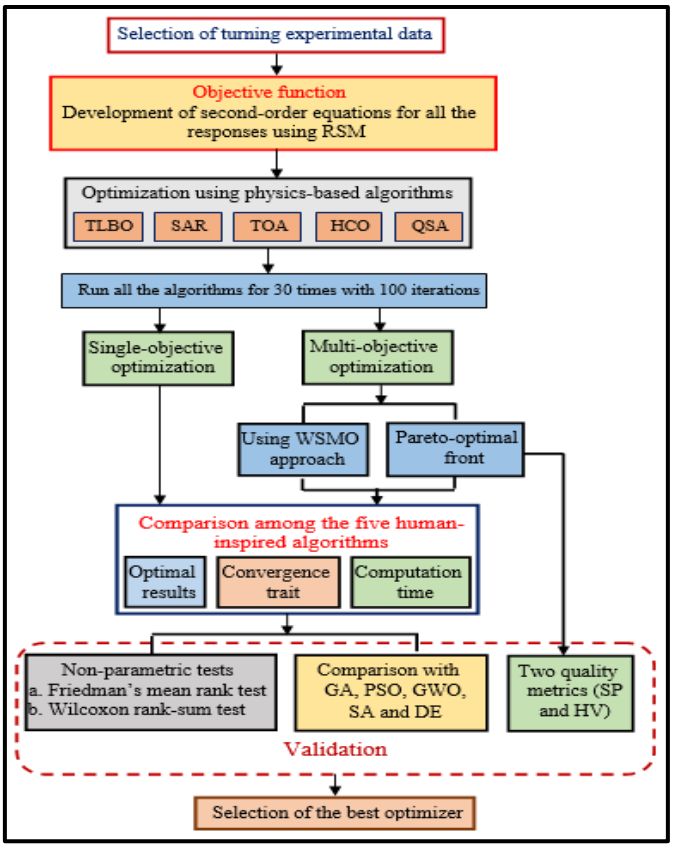


Figure 4.1 Flow diagram for optimization of the turning process

All the related Python codes are developed in Spyder (Anacoda-3), and executed on a Windows 10 Pro system using an Intel® Core™ i3-6006U CPU (2.00 GHz) and 4 GB RAM platform. Each of the algorithms is independently run 30 times, using a population size of 50 and 100 iterations per run. In total, 3000 trials (30 runs×100 iterations) are conducted to perform the analysis. The findings of this work contribute to the selection of an ideal algorithm for the said turning operation, improving machining efficiency and performance.

Camposeco-Negrete [271] performed 20 turning experiments based on a CCD plan, including eight factorial points, six axial points and six center point replications, on AISI 6061-T6 aluminium, considering three input parameters, i.e. CS (x_1) (in m/min), FR (x_2) (in mm/rev) and DOC (x_3) (in mm); and three responses, i.e. SCEC (J/mm^3), SR (μm) and MRR (mm^3/s). The experiments were conducted while setting values of the three turning parameters at five distinct levels, i.e. CS {266, 300, 350, 400, 434 m/min}, FR {0.12, 0.15, 0.20, 0.25, 0.28 mm/rev} and DOC {0.66, 1.00, 1.50, 2.00, 2.34 mm}. After every turning experiment, the power consumption was monitored using a LabVIEW interface at intervals of 0.1 sec, while the corresponding SR values were reported utilizing an SJ-201 Mitutoyo surface roughness tester. It is important to note that SR and SCEC are 'smaller-the-better' responses, but MRR is a 'larger-the-better' response. Experimentally, it was shown that FR had been the most significant parameter in minimizing both SCEC and SR, and minimum SCEC values could only be achieved at lower CS, and higher FR and DOC. On the other hand, a lower SR could be obtained with smaller FR values, while CS and DOC had negligible influences on SR. Larger settings of all the input variables would contribute to maximum MRR [271]. After combining RSM with DFA, Camposeco-Negrete [271] observed that the optimal combination of various turning parameters as CS = 434 m/min, FR = 0.14 mm/rev and DOC = 2.3 mm would provide the maximum values of MRR, and minimum values of SCEC and SR as 2329.13 mm^3/s, 84.48 J/mm^3 and 0.86 μm, respectively. The following second-order equations for the responses were formulated employing Design Expert software (Version 13.0), which would be later employed as the objective functions while working out the single- and multi-objective optimization problems related to the said turning process using five human-inspired metaheuristic algorithms.

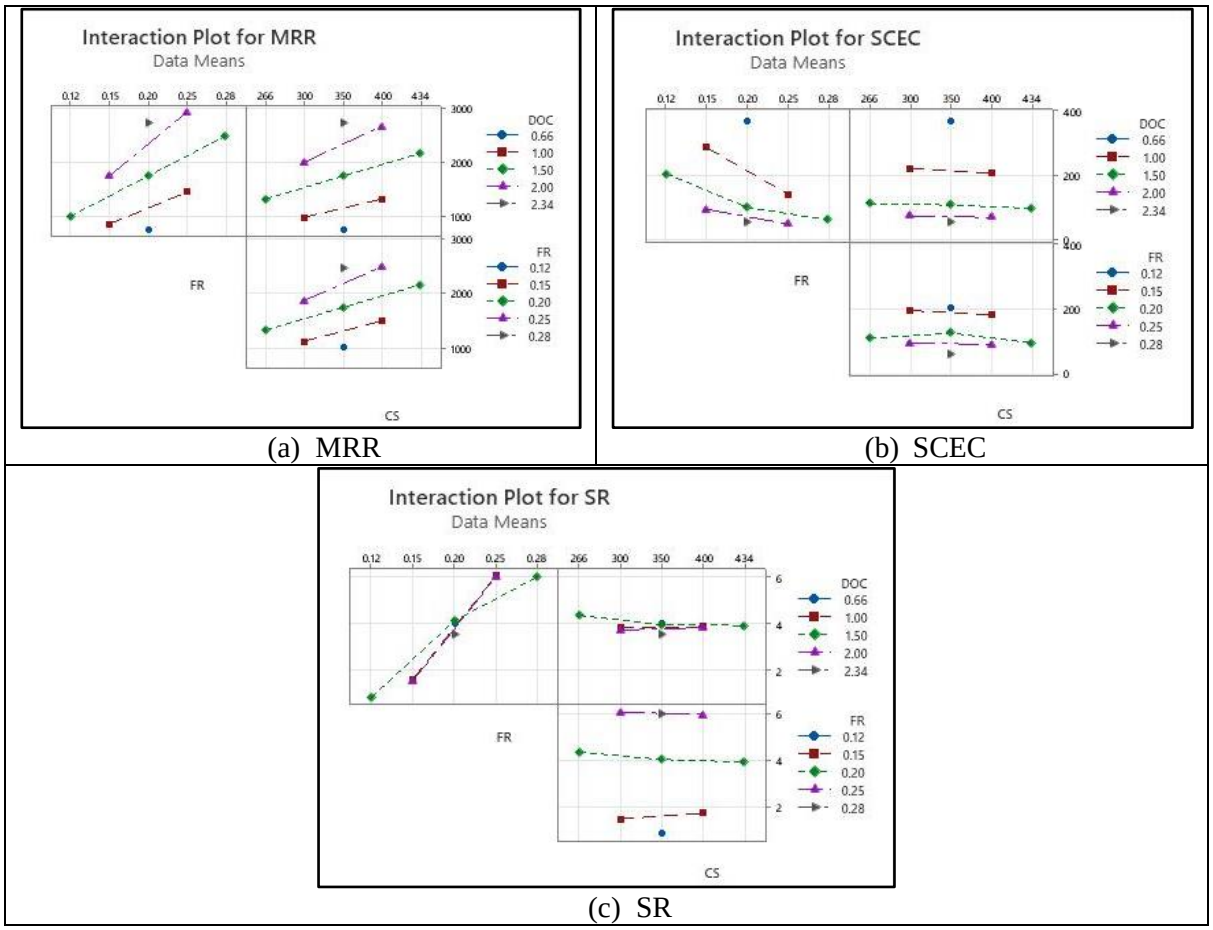
$$Y(MRR) = 1750 + 583.33\times x_1 + 437.5\times x_2 + 250\times x_3 + 145.83\times x_1\times x_2 + 83.33\times x_1\times x_3 + 62.5\times x_2x_3 - 0.0000000000000621\times x_1^2 - 0.0000000000000986\times x_2^2 - 0.0000000000000902\times x_3^2 \quad (4.1)$$

$$Y(SR) = 4.123 - 0.044\times x_1 + 1.146\times x_2 - 0.016\times x_3 + 0.003\times x_1x_2 + 0.004\times x_1x_3 - 0.032\times x_2x_3 - 0.04\times x_1^2 - 0.082\times x_2^2 + 0.006\times x_3^2 \quad (4.2)$$

$$Y(SCEC) = 98.59 - 79.95\times x_1 - 45.45\times x_2 - 4.27\times x_3 + 24.66\times x_1x_2 + 2.33\times x_1x_3 + 2.2\times x_2x_3 + 37.72\times x_1^2 + 9.81\times x_2^2 - 0.33\times x_3^2 \quad (4.3)$$

While performing the turning operation on AISI 6061-T6 aluminium and developing the corresponding regression equations, it would also be wise to study the possible interactions between the considered input parameters. In the developed interaction plots, as shown in Figure 4.2, parallel lines indicate no interaction between the turning parameters, while non-parallel lines with higher slope

suggest stronger interactions. For MRR and SCEC responses, the interaction plots reveal significant interactions between DOC and FR, as well as between FR and CS for SCEC and SR responses. Marginal interactions are noticed between DOC and FR, and DOC and CS for SR, whereas, DOC and CS are found to be nearly independent parameters for MRR and SCEC responses. Additionally, no significant interactions are evident between FR and CS for MRR response.



(a) MRR (b) SCEC

(c) SR

Figure 4.2 Interaction plots between the turning parameters

Sensitivity analysis play a key role in evaluating and enhancing a model, thereby improving its reliability and interpretability. In the context of metaheuristic-driven optimization of the considered tuning operation, it helps to identify the most influential input parameter, investigate the effects of DOC, FR and CS on the responses, and refine the model for better performance. Based on the approach proposed by Ragavendran et al. [272], sensitivity analysis studies are performed and the derived results are presented in Figures 4.3-4.5. From Figure 4.3(a), it is noted that the sensitivity of DOC on MRR is always positive and remains unaffected at its varying values. On the other hand, SCEC increases with higher DOC values. However, for SR, an increase in DOC results in a reduction in its sensitivity. In Figures 4.3(b) and 4.3(c), the sensitivity of DOC on MRR and SR increases with higher levels of both FR and CS. For SCEC, the sensitivity of DOC also increases with an increase in FR parameter, but remains negative and unaffected by changes in CS.

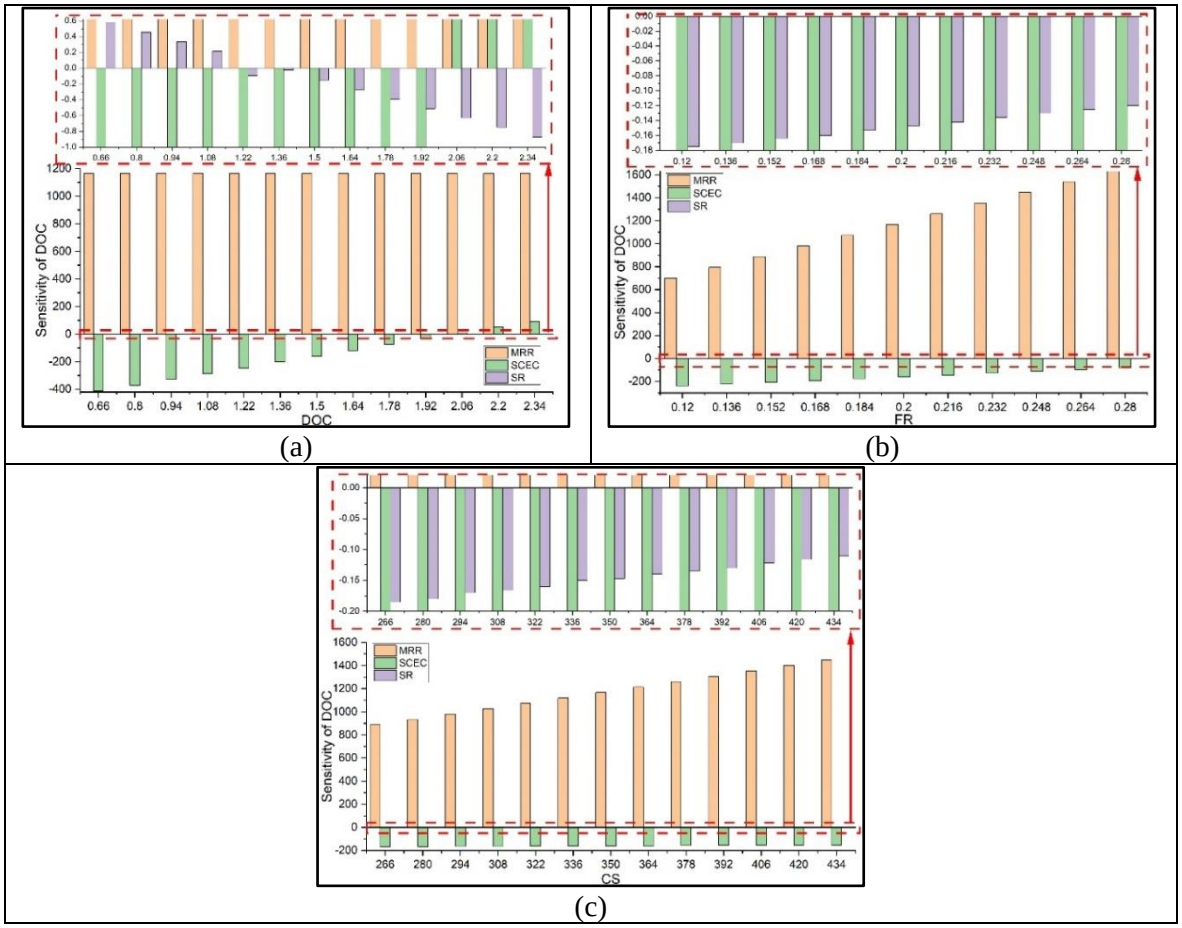
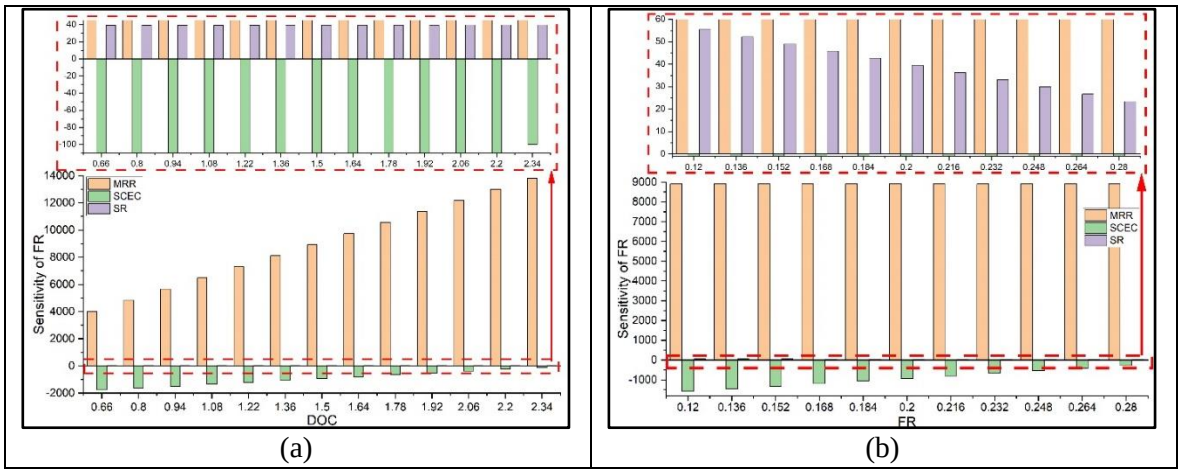
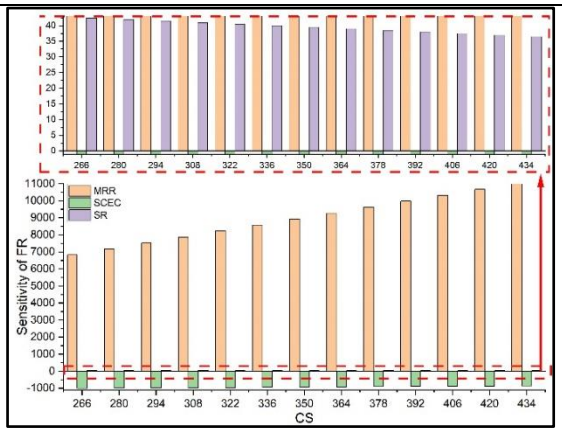


Figure 4.3 Results of sensitivity analysis with respect to DOC

The sensitivity of FR on MRR, SCEC and SR is illustrated in Figure 4.4. Figure 4.4(a) reveals that the sensitivity of FR on MRR and SCEC gradually increases, while the sensitivity on SR remains constant at higher values of DOC. In Figure 4.4(b), the sensitivity of FR on MRR is positive and remains unaffected by changes in DOC; whereas, the sensitivity on SCEC increases and that on SR decreases at higher FR levels. It is observed from Figure 4.4(c) that the sensitivity of FR on MRR increases, the sensitivity on SR decreases, and the sensitivity on SCEC remains constant as CS increases.

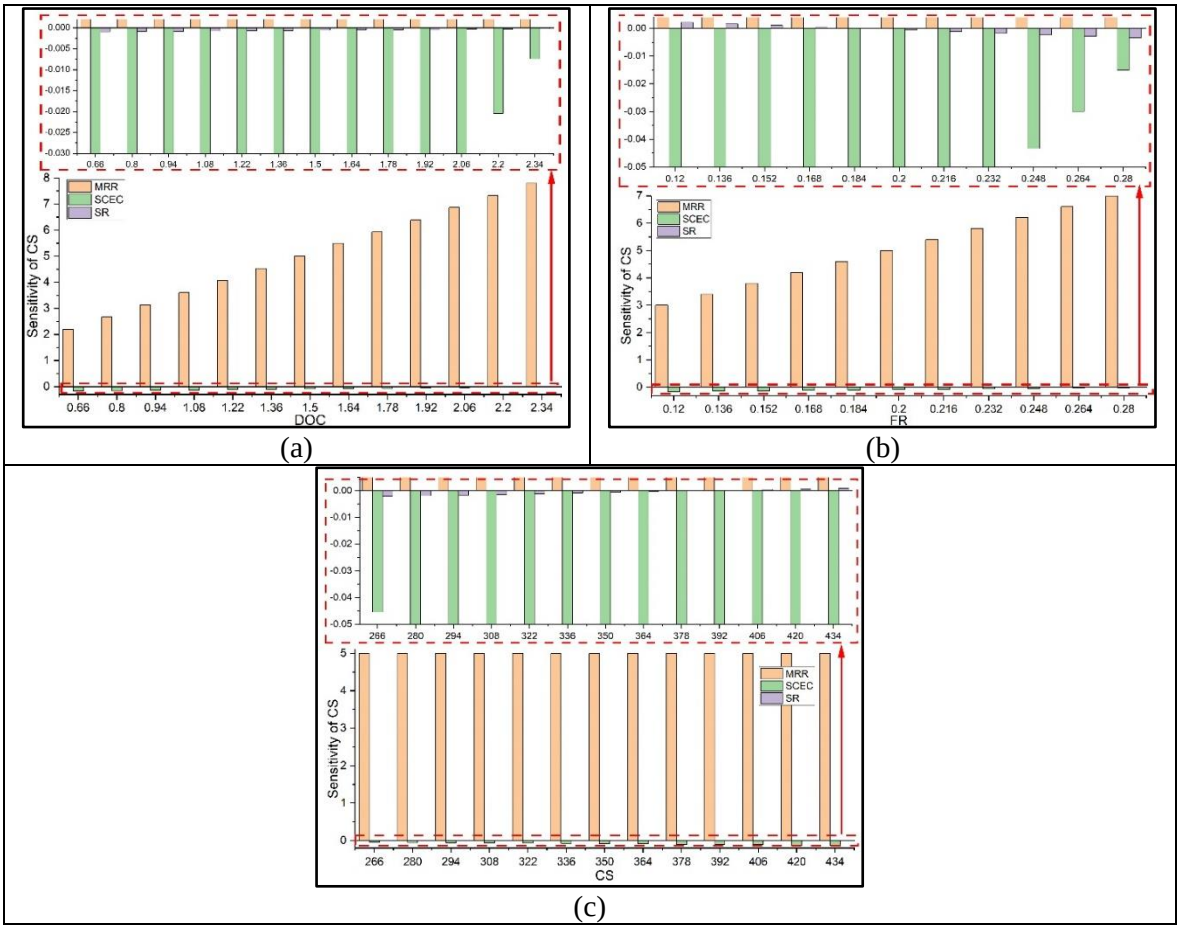




(c)

Figure 4.4 Results of sensitivity analysis with respect to FR

In Figure 4.5(a), the sensitivity of CS on MRR, SCEC and SR gradually increases as DOC increases. In Figure 4.5(b), the sensitivity of CS on MRR and SCEC increases with higher values of FR, while the sensitivity on SR decreases as FR increases. On the other hand, Figure 4.5(c) portrays that the sensitivity of CS on MRR is positive and remains unaffected by changes in CS; whereas, the sensitivity on SR increases and the sensitivity on SCEC decreases with increase in CS.



(a) (b)

(c)

Figure 4.5 Results of sensitivity analysis with respect to CS

4.1 Single-objective optimization of the turning process

Now, treating the aforementioned RSM-based equations as the objective functions, with lower and higher settings of the turning parameters as minimum and maximum limits of the design variables, respectively, each of the three responses is individually optimized using the five considered algorithms. The corresponding constraints are thereby treated as the working levels of the three turning parameters, i.e. 266 m/min $\leq x_1 \leq$ 434 m/min, 0.12 mm/rev $\leq x_2 \leq$ 0.28 mm/rev and 0.66 mm $\leq x_3 \leq$ 2.34 mm. Results of the single-objective optimization are exhibited in Table 4.1, which also contrasts the relative performance of the adopted algorithms.

Table 4.1 indicates that these metaheuristics provide considerable enhancements in the response values, like MRR (18.39-27.08%), SR (34.53-36.40%) and SCEC (28.52-31.16%), as compared to those recorded by Camposeco-Negrete [271]. TLBO, SAR and QSA derive the optimal values of SCEC and SR as 58.154 J/mm^3 and 0.547 µm, respectively, which are better than those obtained using TOA and HCO. The maximum MRR is achieved at the optimal combination of the turning parameters, specifically as CS = 434 m/min, FR = 0.28 mm/rev and DOC = 2.34 mm. TLBO, TOA, HCO and QSA can identify its global maximum value as 2959.80 mm^3/s. The results of single-objective optimization indicate that TLBO and QSA can provide the optimal values of all the considered responses.

Table 4.1 Solutions of the single-objective optimization problem

Optimizer	Response	Optimum	Turning parameter			Mean	SD	% improvement
			x_1	x_2	x_3			
Camposeco-Negrete [271]	MRR (mm^3/s)	2329.13	434	0.14	2.3	-	-	-
	SCEC (J/mm^3)	84.48						
	SR (µm)	0.86						
TLBO	MRR (mm^3/s)	2959.80	434	0.28	2.34	2943.17	51.454	27.08
	SCEC (J/mm^3)	58.154	266	0.28	1.8	58.34	0.534	31.16
	SR (µm)	0.547	266	0.12	2.34	0.55	0.007	36.40
SAR	MRR (mm^3/s)	2757.46	395	0.27	2.03	2740.00	44.690	18.39
	SCEC (J/mm^3)	58.154	266	0.28	1.8	58.46	0.755	31.16
	SR (µm)	0.547	266	0.12	2.34	0.55	0.008	36.40
TOA	MRR (mm^3/s)	2959.80	434	0.28	2.34	2933.00	71.370	27.08
	SCEC (J/mm^3)	60.370	311.2	0.27	1.83	60.52	0.494	28.54
	SR (µm)	0.559	266	0.12	0.66	0.565	0.016	35.00
HCO	MRR (mm^3/s)	2959.80	434	0.28	2.34	2934.62	65.200	27.08
	SCEC (J/mm^3)	60.390	307.8	0.28	1.78	60.56	0.510	28.52
	SR (µm)	0.563	317	0.12	1.97	0.57	0.017	34.53
QSA	MRR (mm^3/s)	2959.80	434	0.28	2.34	2936.60	55.126	27.08
	SCEC (J/mm^3)	58.154	266	0.28	1.8	58.34	0.540	31.16
	SR (µm)	0.547	266	0.12	2.34	0.55	0.007	36.40

Figure 4.6 plots the convergence diagram of each of the algorithms for the considered responses. Except SAR, TOA and HCO, the other algorithms (TLBO and QSA) show rapid progress towards the corresponding optimal solutions after only a few iterations. The convergence behaviors of TLBO and

QSA are almost same for all the responses. These convergence diagrams primarily prove the superiority of TLBO over the other algorithms in achieving the optimal solutions with least number of iterations. With respect to the speed of convergence, both HCO and TOA perform the worst, showing poor advancement towards the optimal solutions.

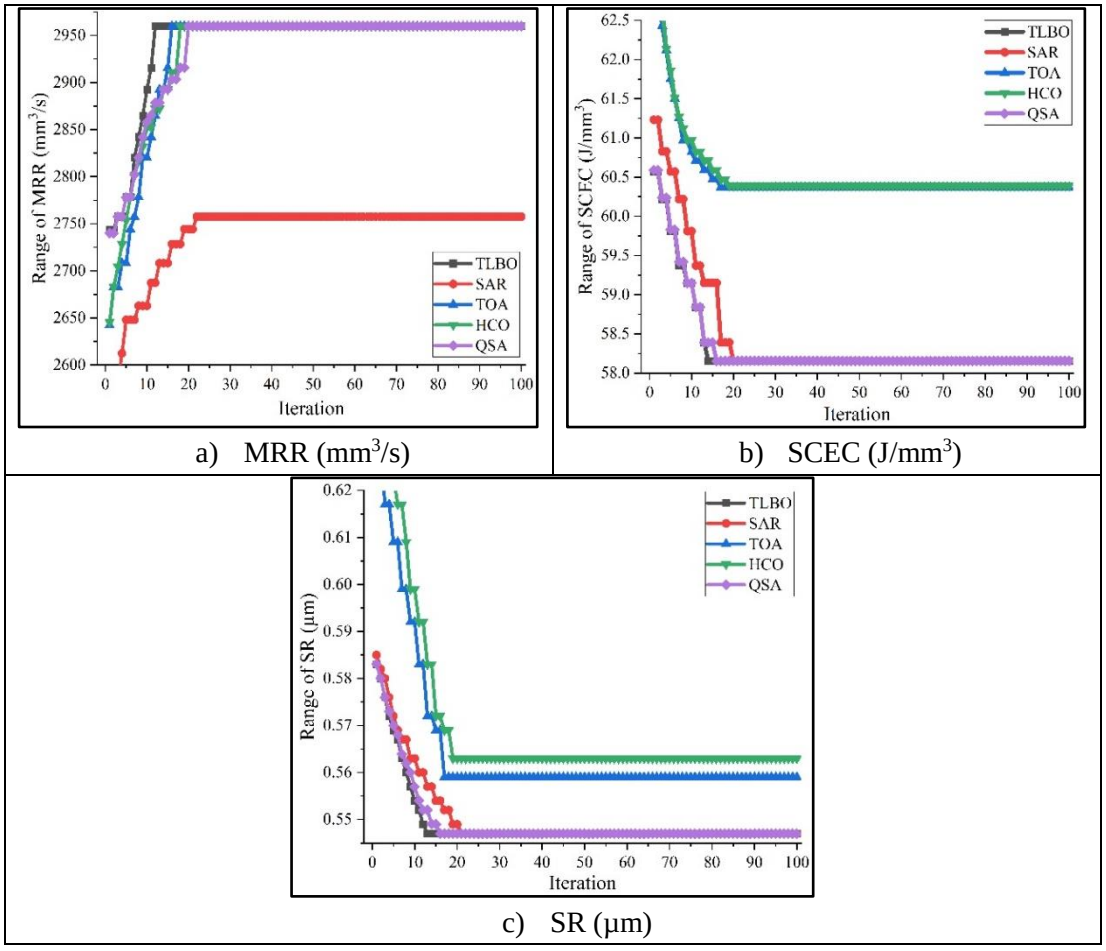


Figure 4.6 Convergence diagrams for the human-inspired metaheuristic algorithms

Figure 4.7 exhibits the box plots combined with data density diagrams illustrating the distribution and variability of the responses against the five algorithms, and also validating supremacy of TLBO across its competitors with reference to the achieved solutions. Box plots can effectively showcase reproducibility of the results attained from the optimization algorithms. In these box plots, data density estimation is contemplated to describe shape of the distribution for each response. The narrow portion of a box plot highlights less probability, while a thicker portion indicates high probability that an algorithm can identify the achieved value as the optimal, which is located nearer to the mean. In Figure 4.7, as compared to other algorithms, TLBO has a relatively compact distribution with small ranges and few outliers. Its corresponding box plots exhibit narrow interquartile ranges, indicating lower variability, while the data density estimations show denser concentration of data around the median, reflecting consistency in the derived solutions. Similarly, QSA also has narrow interquartile ranges with less outliers. However, its data density estimations show slightly more variation in the data spread compared to TLBO. Overall, both TLBO and QSA demonstrate consistent performance with minimal

variability, while TOA and HCO have maximum variability with large number of outliers.

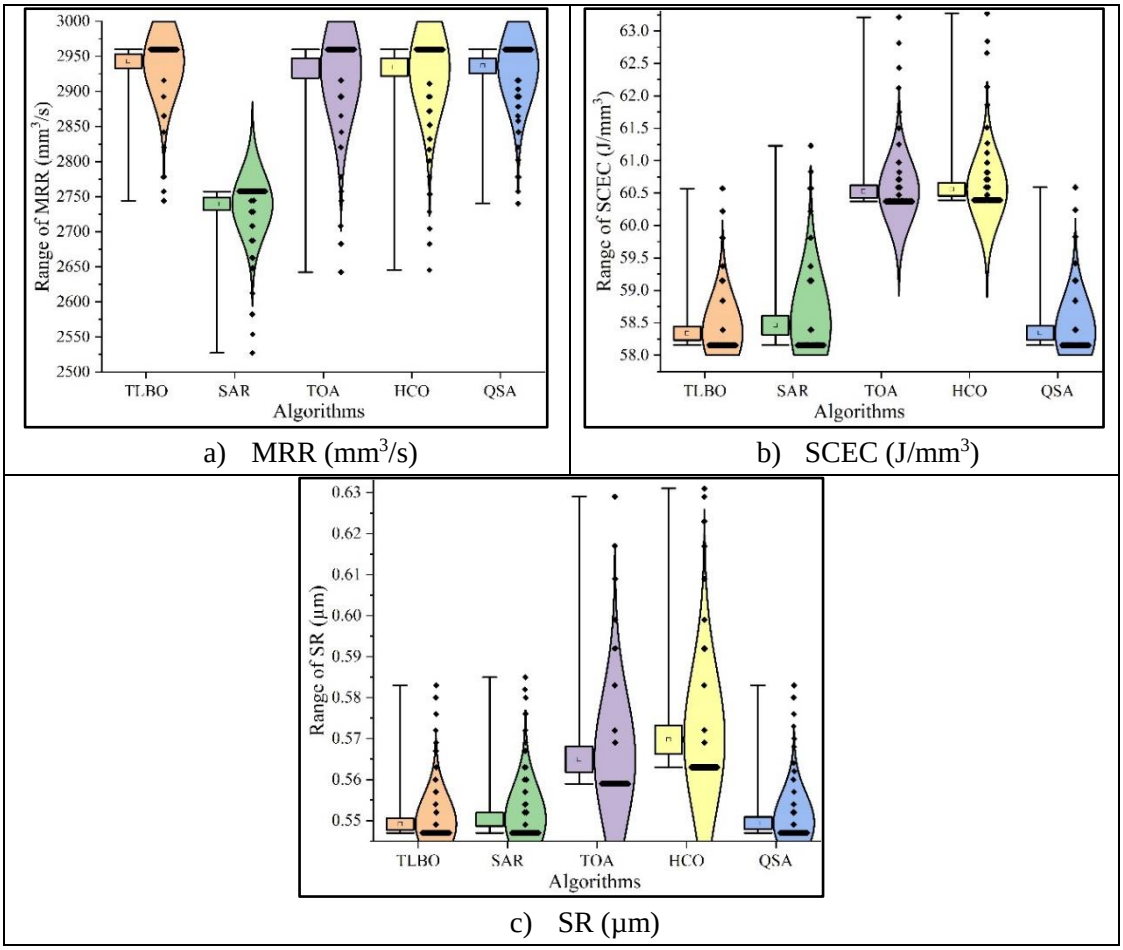


Figure 4.7 Developed box plots for the human-inspired metaheuristic algorithms

Table 4.1 shows that single-objective optimization derives entirely conflicting settings of the turning parameters in deriving the higher MRR, and lower SR and SCEC values, rendering it infeasible to attain in a single turning setup. It takes into account only one individual response at a time, while multi-objective optimization can search out a distinct combination of the turning parameters that can simultaneously optimize all the responses, fulfilling the objectives of both process economy and better surface quality. Figure 4.8 depicts the mean computation time for the three responses using five human-inspired metaheuristics, in the context of single-objective optimization. It reveals that TLBO, TOA, HCO and QSA are computationally efficient optimizers. Thus, on the basis of the solution quality, average computing time and effort of convergence, TLBO outperforms the other considered human-inspired algorithms.

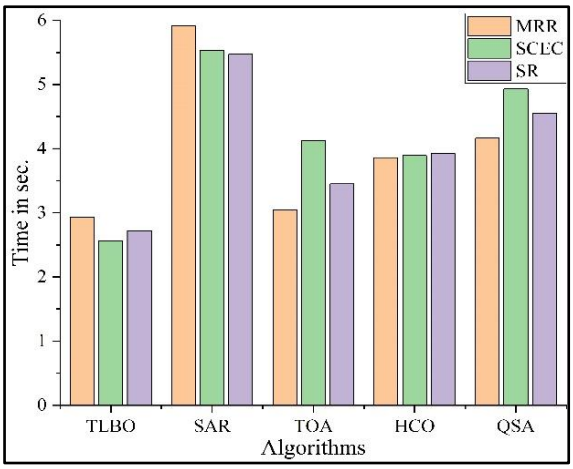


Figure 4.8 Mean computing time for single-objective optimization

4.2 Multi-objective optimization of the turning process

In single-objective optimization, each technique identifies different parametric settings for each of the responses, making it challenging to maintain in real scenarios. To overcome this problem, concurrent optimization of all the responses is necessary, ensuring a unique mixture of the turning parameters. In this research work, the WSMO model is initially applied, followed by framing of the related Pareto optimal front. The corresponding multi-objective function for WSMO is developed as follows:

$$Min\ (Z) = \left[-\frac{w_1 \times Y(MRR)}{MRR_{max}} + \frac{w_2 \times Y(SR)}{SR_{min}} + \frac{w_3 \times Y(SCEC)}{SCEC_{min}}\right] \quad (4.4)$$

where Y(*MRR*), Y(*SR*) and Y(*SCEC*) represent the RSM-based equations for MRR, SR and SCEC, respectively; MRR_{max}, SR_{min} and $SCEC_{min}$ are the results based on single-objective optimization for MRR, SR and SCEC, respectively; and w_1, w_2 and w_3 are the weights allocated to MRR, SR and SCEC, respectively (sum of these weights must be equal to one). However, in WSMO approach, values of the best solutions largely depend on the weights allocated to the responses. If the relative importance of the responses is unknown, treating them with equal importance may be reasonable. However, in practical applications, the importance of the responses may differ based on specific requirements. Table 4.2 outlines the results of multi-objective optimization of the said turning operation. Intriguingly, all the five human-inspired algorithms achieve better response values compared to those recorded by Camposeco-Negrete [271]. Among them, TLBO and QSA supersede SAR, TOA and HCO in maximizing MRR, and minimizing SR and SCEC. To concurrently optimize all the responses, it is advisable to set the ideal parametric combination as CS = 434 m/min, FR = 0.17 mm/rev and DOC = 2.16 mm. A comparative analysis reveals that almost all the algorithms result in similar improvement of 26.63% in SR (16.28% for HCO). For MRR, the performance improvement ranges from 6.03% to 15.03%, while for SCEC, TLBO and QSA show the maximum improvement of 23.27% against the observations of Camposeco-Negrete [271]. To address the dependency issue of response weights in WSMO approach, the related 3D Pareto optimal front is developed for TLBO technique, as depicted in Figure 4.9. Pareto optimality implies that no feasible solution is strictly superior in all objectives while

being at least as good in one objective compared to other solutions in the Pareto set. The shape and spread of the Pareto front demonstrate how efficiently TLBO explores the solution space, highlighting the necessary trade-offs while simultaneously optimizing all the responses. In this plot, red points in the lower portion and green points in the upper portion represent non-dominated solutions, forming the Pareto optimal front. These solutions reflect the best possible trade-offs between the three objectives, further validating effectiveness of TLBO. The 'Optimal point', marked in black on the Pareto front, highlights the best compromised solution.

Table 4.2 Multi-objective optimization results

Optimizer	Response	Optimum	Min z	Turning parameter			% improvement
				x_1	x_2	x_3	
Camposeco-Negrete [271]	MRR (mm^3/s)	2329.13	-	434	0.14	2.3	-
	SCEC (J/mm^3)	84.48					
	SR (μm)	0.86					
TLBO	MRR (mm^3/s)	2679.1	0.332	434	0.17	2.16	15.03
	SCEC (J/mm^3)	64.820					23.27
	SR (μm)	0.631					26.63
SAR	MRR (mm^3/s)	2590.98	0.333	434	0.2	2.1	11.24
	SCEC (J/mm^3)	65.510					22.46
	SR (μm)	0.631					26.63
TOA	MRR (mm^3/s)	2469.57	0.334	432	0.24	2.02	6.03
	SCEC (J/mm^3)	68.730					18.64
	SR (μm)	0.631					26.63
HCO	MRR (mm^3/s)	2504.35	0.336	429	0.23	2.01	7.52
	SCEC (J/mm^3)	70.190					16.92
	SR (μm)	0.720					16.28
QSA	MRR (mm^3/s)	2679.1	0.332	434	0.17	2.16	15.03
	SCEC (J/mm^3)	64.820					23.27
	SR (μm)	0.631					26.63

The parallel plot designed for TLBO algorithm, as shown in Figure 4.10, analyzes the impact of different parametric settings on the responses and their possible interactions. A parallel plot uses multiple lines, each line representing a data point moving across various dimensions (input parameters and responses). It visually demonstrates how different parameter values affect the optimization objectives, aiding in identification of the ideal settings leading to the optimal response values. It reiterates that to maximize MRR (2679.1 mm^3/s), and minimize SCEC (64.82 J/mm^3) and SR (0.631 μm), the specific combination of CS = 434 m/min, FR = 0.17 mm/rev and DOC = 2.16 mm should be the optimal choice, while performing turning operation on AISI 6061-T6 aluminium work material.

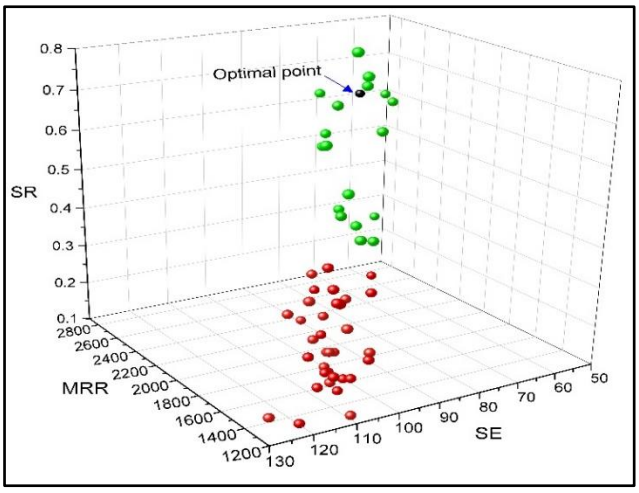


Figure 4.9 3D Pareto optimal front for TLBO algorithm

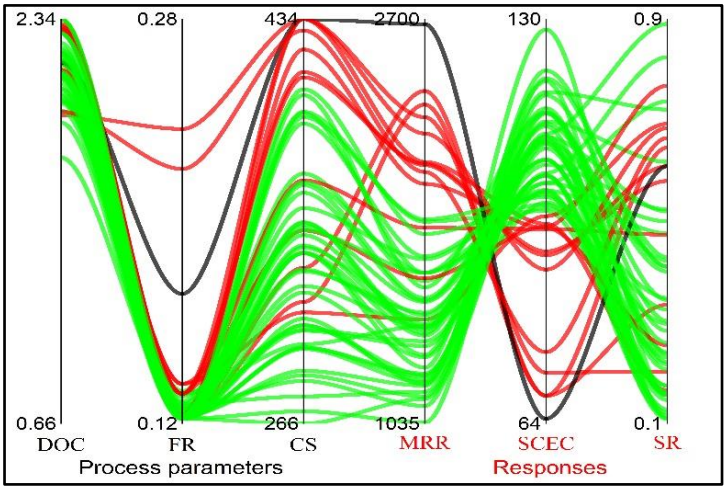


Figure 4.10 Parallel plot for TLBO algorithm

Figure 4.11 compares the average computation time based on 30 independent runs of the five human-inspired metaheuristics, while resolving the corresponding multi-objective optimization problem. According to the mean computing time, these algorithms can be rated as TLBO > HCO > TOA > QSA > SAR, with TLBO excelling in successfully picking out the optimal combination of the turning parameters with minimal computational effort.

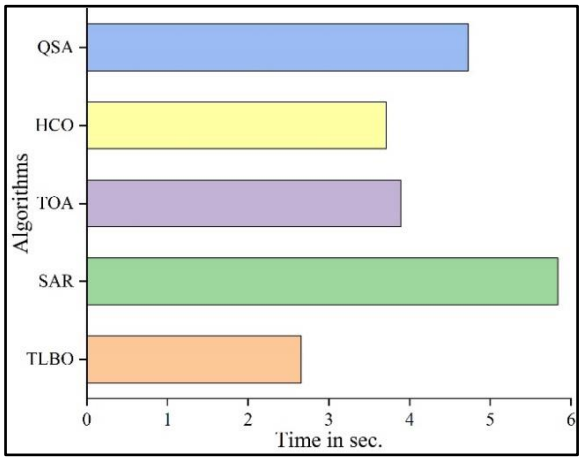


Figure 4.11 Mean computing time for multi-objective optimization

4.3 Performance analysis

As mentioned earlier, this study adopts HV and SP [273] as the two quality metrics to assess the performance of the five human-inspired metaheuristics. Table 4.3 provides their values. It is important to note that lower SP and higher HV values represent superior performance of an algorithm. Table 4.3 exhibits the best values of HV and SP, expressed in bold letters. Therefore, on the basis of these metrics, it can be concluded that TLBO has an imposing performance, excelling over its other competitors, and reveals its superb ability in resolving high-dimensional optimization problems.

Table 4.3 HV and SP values

Algorithm	TLBO	SAR	TOA	HCO	QSA
HV	**126.34**	125.72	124.59	123.90	125.98
SP	**22.24**	22.49	22.97	23.51	22.43

To critically assess performance of the five optimization algorithms, the Friedman's mean rank test [274] is conducted, as depicted in Table 4.4, with 1% significance level to appraise the statistically significant deviation between TLBO, and TOA, QSA, HCO and SAR optimizers. According to the results presented in Table 4.4, it is proved that TLBO outperforms others across all the considered responses, while HCO exhibits the worst performance.

Table 4.4 Friedman's mean rank test results

Algorithm	Response	TLBO	SAR	TOA	HCO	QSA
Average	MRR	2943.17	2739.63	2933.35	2934.62	2936.62
Rank sum		228	500	264	258.5	249.5
Average rank		2.28	5	2.64	2.585	2.495
Rank		**1**	**5**	**4**	**3**	**2**
Average	SR	0.5492	0.5503	0.5650	0.5698	0.5494
Rank sum		184	220	400	500	196
Average rank		1.84	2.2	4	5	1.96
Rank		**1**	**3**	**4**	**5**	**2**
Average	SCEC	58.3371	58.4630	60.5245	60.5577	58.3440
Rank sum		185.5	219	400.5	499.5	195.5
Average rank		1.855	2.19	4.005	4.995	1.955
Rank		**1**	**3**	**4**	**5**	**2**

Finally, to justify the meaningful difference between TLBO and other algorithms, Wilcoxon rank sum test [275] is also carried out with 1% significance level. The results, presented in Table 4.5, unveil that there is a notable difference between TLBO and its other competitors. The Wilcoxon rank sum test results are denoted as either significantly superior performance (+) or inferior performance (-) when comparing TLBO against the other algorithms. Table 4.5 provides the sum of negative ranks (R−), positive ranks (R+) and *p*-values calculated for the pair-wise comparisons with respect to TLBO algorithm.

Table 4.5 Results of the Wilcoxon rank-sum test

TLBO vs		SAR	TOA	HCO	QSA
R +	MRR	142	120	136	150
R –		11	24	18	3
p-value		1.00E-05	0.0032	0.0022	0.0025
R +	SR	210	280	156	78
R –		27	38	14	2
p-value		0.0004	0.0021	0.0016	0.00111
R +	SCEC	190	172	212	55
R –		34	46	28	13
p-value		0.0071	0.0019	0.0030	0.00256

4.4 Results and discussion

This study investigates application of five human-inspired algorithms for parametric optimization of turning operation on AISI 6061-T6 aluminium work material. They achieve improvements ranging from 18.39 to 27.08% for MRR, 34.53 to 36.40% for SR, and 28.52 to 31.16% for SCEC in single-objective optimization; and 6.03 to 15.03%, 16.28 to 26.63%, and 16.92 to 23.27% improvements for MRR, SR and SCEC, respectively in multi-objective optimization, compared to the results reported by Camposeco-Negrete [271]. Among these algorithms, TLBO consistently delivers the best results, proving to be most efficient in both single- and multi-objective optimization of the said turning process. In single-objective optimization, it attains the optimal values of 2943.17 mm^3/s for MRR, 0.55 μm for SR and 58.34 J/mm^3 for SCEC. Furthermore, in multi-objective optimization, the corresponding optimal values are 2679.1 mm^3/s, 0.631 μm and 64.82 J/mm^3 for MRR, SR and SCEC, respectively, at the best possible parametric combination of CS = 434 m/min, FR = 0.17 mm/rev and DOC = 2.16 mm. TLBO also excels over the others with reference to speed of convergence and computing time. For multi-objective optimization, it takes just 2.6 sec to find out the global optimal solution, reducing the runtime by 119.55, 46.24, 39.47 and 77.82% against SAR, TOA, HCO and QSA techniques, respectively. Its superiority is further confirmed using SP and HV values, and validation through Friedman's mean rank and Wilcoxon rank sum tests.

In order to validate supremacy of the human-inspired metaheuristic algorithms in solving complex parametric optimization problems, their performance is now contrasted across several benchmark functions, like Rosenbrock, Ackley, Griewank, Levy, Scahffer 2, Bohachevsky 1, six hump camel and Goldstein price, as exhibited in Table 4.6. It can be interestingly noticed that all the algorithms under consideration are able to provide satisfactory solutions for all the benchmark functions, while showcasing slightly superior performance of TLBO and QSA techniques.

Table 4.6 Comparison of performance of the human-inspired algorithms across different benchmark functions

Benchmark function	Dim	Algorithm	f_{min}	Avg. time
Rosenbrock: $F_1(x) = \sum_{i=1}^{n-1}\left[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2\right]$	30	TLBO	2.78E-06	3.6
		SAR	0	4.66
		TOA	0.00013	3.85
		HCO	934.35	4.23
		QSA	2.28E-05	3.9
Ackley: $F_2(x) = -20\,exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) - exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right) + 20 + e$	30	TLBO	3.55E-15	3.95
		SAR	0	6.17
		TOA	0	4.44
		HCO	0.00014	4.85
		QSA	0	4.92
Griewank: $F_3(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	30	TLBO	2.16E-05	2.9
		SAR	0	4.86
		TOA	0	3.3
		HCO	0.0011	4.97
		QSA	0	4.56
Levy: $F_4(x) = 0.1\{sin^2(3\pi x_1) + \sum_{i=1}^{n}(x_i - 1)^2[1 + sin^2(3\pi x_1 + 1)] + (x_n - 1)^2[1 + sin^2(2\pi x_n)]\} + \sum_{i=1}^{n} u(x_i, 5, 100, 4)$	30	TLBO	1.35E-31	4.2
		SAR	1.35E-31	5.37
		TOA	0.039	4.9
		HCO	0.029	4.8
		QSA	1.35E-31	4.64
Scahffer 2: $F_5(x) = 0.5 + ((\sin(\sum_{i=1}^{D} x_i^2))^2 - 0.5) \times (1 + 0.001\,(\sum_{i=1}^{D} x_i^2))^{-2}$	30	TLBO	3.90E-14	2.72
		SAR	0	5.48
		TOA	0	3.01
		HCO	1.01E-09	4.7
		QSA	0	4.59
Bohachevsky 1: $F_6(x) = (\sum_{i=1}^{D}[x_i^2 + 2x_{i+1}^2 - 0.3\cos(3\pi x_i) - 0.4\cos(4\pi x_{i+1}) + 0.7]$	30	TLBO	0	3.77
		SAR	0	5.36
		TOA	0	4.65
		HCO	2.1	4.7
		QSA	0	4.85
Six hump camel : $F_7(x) = \left\{\left[\sum_{i=1}^{n} sin^2(x_i)\right] - \exp(-\sum_{i=1}^{n} x_i^2)\right\} \times exp\left[-\sum_{i=1}^{n} sin^2\sqrt{|x_i|}\right]$	30	TLBO	-1.03	3.65
		SAR	-1.03	4.77
		TOA	-1.03	4.28
		HCO	11078.8	4.9
		QSA	-1.03	4.12

Table 4.6 Contd.

Goldstein price: $F_8(x) = [1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)] \times [30 + (2x_1 - 3x_2)^2 \times (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$	30	TLBO	3	3.67
		SAR	3	5.33
		TOA	4.88	4.54
		HCO	14.77	4.97
		QSA	3	4.2

To highlight the differences between TLBO and its other competitors, Table 4.7 contrasts its performance for both single- and multi-objective optimization, focusing on percentage improvements in the response values and mean computing time. It depicts superior or same results across all the responses. For single-objective optimization, it performs similarly to SAR and QSA for SCEC and SR, while TOA, HCO and QSA for MRR. In multi-objective optimization, QSA shows comparable performance to TLBO across all the responses. However, TLBO significantly outperforms its competitors in terms of average computation time. Overall, the findings suggest that higher CS, moderate FR and moderate-to-higher DOC values would result in concurrent optimization of all the responses, with TLBO emerging as the most efficient technique.

Table 4.7 Percentage improvements of the responses and computing time for TLBO across its competitors

Optimizer	SAR	TOA	HCO	QSA	SAR	TOA	HCO	QSA
Technique	Single-objective				Multi-objective			
MRR	6.84	0	0	0	3.29	7.82	6.52	0
SR	0	2.2	2.92	0	0	0	14.1	0
SCEC	0	3.81	3.84	0	1.06	6.032	8.28	0
Time	106.275	30.62	42.6	67.28	119.55	46.24	39.47	77.82

The considered turning operation is also optimized (single- and multi-objective) using several favored metaheuristic algorithms, like GA, PSO, GWO, SA and DE. The derived results, as depicted in Table 4.8, are compared against those obtained using TLBO algorithm. It can be noticed from Table 4.8 that TLBO renders significant improvements (in percentage) in the optimal values of all the responses as well as average computational time.

Table 4.8 Comparison of performance of TLBO against GA, PSO, GWO, SA and DE

Optimizer	GA	PSO	GWO	SA	DE	GA	PSO	GWO	SA	DE
Technique	Single-objective					Multi-objective				
MRR	4.55	3.38	1.49	2.39	3.38	2.97	2.79	1.11	1.49	2.23
SR	5.12	4.57	2.19	3.66	3.84	2.22	3.80	1.43	2.22	1.74
SCEC	3.97	3.09	2.14	4.72	3.86	1.80	1.97	1.33	2.24	2.11
Average time	37.03	12.55	10.57	22.78	30.09	18.42	10.53	13.53	15.79	10.53

To justify excellence of TLBO in accurately optimizing the said turning operation on AISI 6061-T6 aluminium, it performance is compared with three hybrid optimization techniques, i.e. HSTLBO [276], HFMOEA [277] and TPLPSO [278]. The results of this comparative analysis with respect to percentage improvements in the response values derived using TLBO against the considered hybrid

algorithms are provided in Table 4.9. It again reiterates its superior applicability in solving parametric optimization problem of the said turning operation.

Table 4.9 Comparison of performance of TLBO against the hybrid algorithms

Optimizer	HSTLBO	HFMOEA	TPLPSO	HSTLBO	HFMOEA	TPLPSO
Technique	Single-objective			Multi-objective		
MRR	1.37	1.91	1.29	1.04	1.31	0.89
SR	1.43	1.84	1.21	1.21	1.26	0.96
SCEC	1.29	2.42	1.37	1.17	1.19	1.07
Average time	5.21	8.87	6.29	4.13	6.5	5.72

Finally, a trade-off analysis between computational complexity and solution quality for TLBO algorithm is conducted in Figure 4.12 to showcase the associationship between the number of agents, average computational time and number of Pareto solutions obtained. It can be noticed from Figure 4.12 that as the number of agents increases, the number of Pareto solutions also rises significantly, indicating that larger agent populations would contribute to exploring the solution space more effectively, while enhancing the diversity of the Pareto solutions. Additionally, the computational time also increases, suggesting that higher agent counts demand for greater computational resources due to complexity of managing and evaluating more agents during the optimization process. Increasing the number of agents improves solution quality (as measured by the number of Pareto solutions), but it comes at the cost of higher computational complexity and time. It thereby suggests a trade-off, where the number of agents and computational time must be carefully balanced to achieve an optimal number of high quality Pareto solutions without incurring excessive computational overhead.

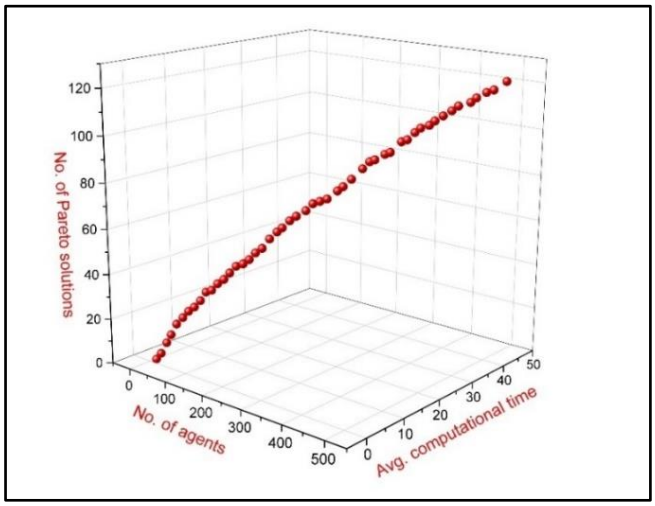


Figure 4.12 Trade-off between computational complexity and solution quality

5. OPTIMIZING A GREEN MACHINING PROCESS BASED ON MATING BEHAVIOR-BASED METAHEURISTICS

The growing emphasis on sustainable manufacturing has led to increased research in optimizing green machining processes to reduce energy consumption and environmental impact. This research work employs five mating behavior-based metaheuristic algorithms, i.e. FMA, NMR, BOA, BMOA and AZOA to optimize a dry milling process based on the past experimental dataset. The relative performance is later compared in terms of convergence speed, computation time, accuracy and variability of the derived solutions. Although all of these algorithms have provided acceptable solutions for many of the well-known benchmark functions with increased complexity, their effectiveness in optimizing any of the machining processes has not yet been explored. The application steps for parametric optimization of the said dry milling operation are depicted in Figure 5.1, through a flow diagram. To strengthen this comparison, two statistical non-parametric tests, i.e. Wilcoxon rank sum test and Friedman's mean rank test are applied to evaluate the performance difference among those algorithms. To derive the optimal multi-objective solutions and unique parametric settings for the green machining process, the corresponding Pareto optimal fronts are subsequently developed. The parallel plots would help in determining the best possible combinations of the green machining parameters and responses. To identify the best performing mating behavior-based algorithm, the corresponding values of SP and HV metrics are also evaluated.

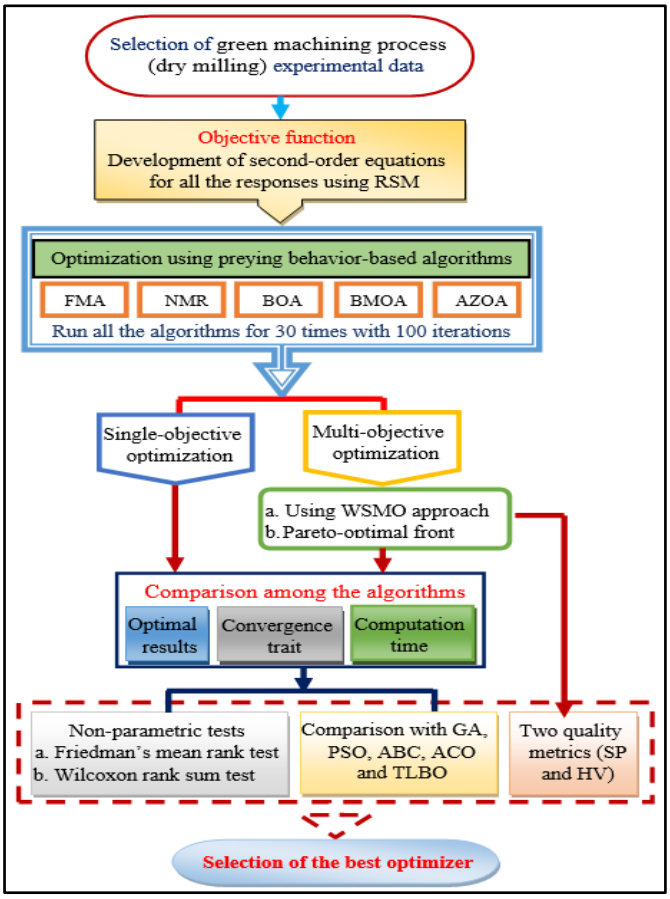


Figure 5.1 Flow diagram for optimization of a green machining process (dry milling)

To further validate the optimization capabilities of the selected algorithms, an illustrative example based on the past experimental data from a dry milling operation is considered, with the primary objective of identifying the optimal combinations of the green machining parameters to minimize energy consumption and promote a sustainable machining environment. This work presents a comprehensive evaluation of the mating behavior-based metaheuristic algorithms for the green machining process optimization, offering insights into their applicability and effectiveness in enhancing sustainability in manufacturing. As mentioned earlier, the related codes are developed in Python on Spyder (Anacoda-3) and run in Windows-10 Pro with an Intel® CoreTM i3-6006U CPU operating at 2.00 GHz processor, and 4.00 GB of RAM environment. Each algorithm is separately run 30 times with a population size of 50 and number of iterations 100. Thus, 30×100 = 3000 trials are used to carry out the analyses.

Treating CS (x_1) (m/min), DOC (x_2) (mm), FR (x_3) (mm/z) and Nr (x_4) (mm) as the input parameters, and PF, EC (kJ) and SR (μm) as the responses, and based on a BBD plan, Nguyen et al. [279] conducted 29 experiments on a Spinner U620 center milling setup for dry machining of SS 304 material with an objective to reduce power consumption, while providing green, safer and environment-friendly machining operation. The values of all the milling parameters were varied at three different levels having equal intervals, i.e. CS {60-160 m/min}, DOC {0.2-1 mm}, FR {0.04-0.12 mm/z} and Nr {0.2-0.8 mm}. During the experiments, power was measured utilizing a power meter KEW6305 attached with three sensors, while a surface roughness tester (Mitutoyo SJ-301) was employed to record the corresponding SR values. Table 5.1 exhibits the adopted experimental design plan and measured response values during dry milling of SS 304 work material. It was noticed that the most desired values of all the responses could be achieved at moderate-to-higher settings of all the four process parameters as CS = 160 m/min, DOC = 0.42 mm, FR = 0.09 mm/z and Nr = 0.8 mm. It was experimentally proved that the value of PF would increase with higher values of CS, DOC, FR and Nr. On the other hand, for EC, higher value of CS and FR would result in reduced EC, and FR was identified as the most influential input parameter with respect to consumption of energy during dry milling of SS 304 work material. In case of SR, higher CS and Nr would provide better surface quality of the machined components, and higher FR and DOC had also been correlated for better surface quality [279].

Using RSM and experimental data of Table 5.1, the following second order equations for the considered responses were developed using Design Expert software (Ver. 13), which would later be treated as the objective functions for solving both single- and multi-objective optimization problems for the said dry milling process using five mating behavior-based metaheuristic algorithms.

$$Y(PF) = 0.495827 - 0.0011826\times x_1 + 0.0304298\times x_2 + 0.511898\times x_3 - 0.313639\times x_4 + 0.000074017\times x_1\times x_2 - 0.00613776\times x_1\times x_3 + 0.00112641\times x_1\times x_4 + 0.626472\times x_2\times x_3 + 0.00133421\times x_2\times x_4 + 0.712156\times x_3\times x_4 + 0.0000114009\times x_1^2 + 0.0874663\times x_2^2 + 11.917\times x_3^2 + 0.299616\times x_4^2 \quad (5.1)$$

$$Y(EC) = 184.205 - 1.28054\times x_1 + 29.261\times x_2 - 1580.25\times x_3 + 15.4049\times x_4 - 0.0970274\times x_1\times x_2 + 3.22594\times x_1\times x_3 - 0.00967865\times x_1\times x_4 - 129.946\times x_2\times x_3 - 3.95919\times x_2\times x_4 - 113.762\times x_3\times x_4 +$$

$0.00342579\times x_1^2 + 3.16239\times x_2^2 + 5723.12\times x_3^2 + 4.83362\times x_4^2$ (5.2)

$Y(SR) = 1.53864 - 0.000562835\times x_1 + 0.775299\times x_2 - 7.5505\times x_3 - 3.3899\times x_4 - 0.00125938\times x_1\times x_2 - 0.03176\times x_1\times x_3 + 0.000255552\times x_1\times x_4 - 6.61321\times x_2\times x_3 - 0.442603\times x_2\times x_4 + 2.91797\times x_3\times x_4 - 0.00000131084\times x_1^2 + 0.575952\times x_2^2 + 129.81\times x_3^2 + 2.56996\times x_4^2$ (5.3)

Table 5.1 Experimental design plan and measured responses during dry milling operation [279]

Exp. No.	x_1 (m/min)	x_2 (mm)	x_3 (mm/z)	x_4 (mm)	EC (kJ)	PF	SR (μm)
1	110	0.2	0.04	0.4	50.33	0.518	0.45
2	110	0.6	0.12	0.8	25.46	0.867	1.08
3	110	0.6	0.08	0.4	31.56	0.652	0.85
4	60	0.6	0.08	0.2	53.66	0.611	1.34
5	160	0.6	0.12	0.4	18.42	0.851	0.95
6	60	0.6	0.12	0.4	42.6	0.736	1.47
7	110	0.2	0.12	0.4	21.99	0.69	1.14
8	60	0.6	0.08	0.8	59.13	0.685	0.78
9	60	1	0.08	0.4	61.68	0.703	1.31
10	110	1	0.12	0.4	26.72	0.868	1.49
11	110	1	0.08	0.2	35.41	0.732	1.42
12	160	0.6	0.08	0.2	22.84	0.719	0.89
13	160	1	0.08	0.4	27.26	0.835	0.79
14	60	0.2	0.08	0.4	48.96	0.547	0.82
15	160	0.6	0.04	0.4	44.62	0.69	0.47
16	110	0.6	0.04	0.2	54.03	0.566	0.98
17	60	0.6	0.04	0.4	94.95	0.529	0.82
18	110	1	0.04	0.4	63.82	0.659	1.06
19	160	0.2	0.08	0.4	22.07	0.671	0.41
20	110	0.6	0.12	0.2	23.74	0.752	1.55
21	110	0.6	0.04	0.8	62.35	0.648	0.52
22	160	0.6	0.08	0.8	26.68	0.862	0.36
23	110	0.2	0.08	0.2	28.23	0.576	0.91
24	110	0.2	0.08	0.8	32.95	0.681	0.48
25	110	1	0.08	0.8	39.02	0.843	0.89
26	80	0.4	0.06	0.4	53.5	0.533	0.45
27	100	0.8	0.02	0.4	86.83	0.563	1.08
28	130	0.5	0.1	0.8	22.98	0.817	0.85
29	150	0.9	0.05	0.2	39.61	0.712	1.34

5.1 Single-objective optimization of the dry milling process

In single-objective optimization, considering the above-mentioned RSM-based equations as the objective functions, the three responses, i.e. PF, EC and SR are separately optimized using the five mating behavior-based metaheuristic algorithms, i.e. FMA, NMR, BOA, BMOA and AZOA. The corresponding constraints are considered as the operating levels of the four dry milling parameters, i.e. 60 m/min $\le x_1 \le$ 160 m/min, 0.2 mm $\le x_2 \le$ 1 mm, 0.04 mm/z $\le x_3 \le$ 0.12 mm/z, and 0.2 mm $\le x_4 \le$ 0.8 mm. The single-objective optimization results for the said dry milling process are provided in Table

5.2. It can be noticed from Table 5.2 that FMA, NMR, BOA and AZOA are able to identify the maximum value of PF as 1.08 which is 28.57% better than that obtained by Nguyen et al. [279]. Among the considered mating behavior-based algorithms, only BMOA cannot find out the optimal value of PF. Thus, the maximum PF can be obtained at the optimal intermix of different milling parameters as CS = 160 m/min, DOC = 1 mm, FR = 0.12 mm/z and Nr = 0.8 mm. A detailed analysis of the results shown in Table 5.2 also unveils the superior performance of FMA against its competitors, i.e. NMR, BOA, BMOA and AZOA, while achieving minimum values of both EC and SR as well as mean and SD of the derived solutions. Thus, using FMA, for attaining minimum EC as 17.88 kJ (with mean 18.07 kJ and SD 0.528 kJ), it is recommended to set the corresponding milling parameters as CS = 137 m/min, DOC = 0.21 mm, FR = 0.1 mm/z and Nr = 0.2 mm, providing 13.04% improvement against that obtained by Nguyen et al. [279]. On the other hand, an optimal combination of dry milling parameters as CS = 160 m/min, DOC = 0.2 mm, FR = 0.04 mm/z and Nr = 0.665 mm would result in attaining the minimum value of SR as 0.27 μm with 25% improvement against its value as noted by Nguyen et al. [279].

Table 5.2 Single objective optimization results for the dry milling operation

Optimizer	Response	Mean	SD	Optimal	Milling parameters				% improvement
					x_1 (m/min)	x_2 (mm)	x_3 (mm/z)	x_4 (mm)	
Nguyen et al. [279]	PF	-	-	0.84	160	0.42	0.09	0.8	-
	EC (kJ)	-	-	20.56					
	SR (μm)	-	-	0.36					
FMA	PF	1.06	0.053	1.08	160	1	0.12	0.8	28.57
	EC (kJ)	18.07	0.528	17.88	137	0.21	0.1	0.2	13.04
	SR (μm)	0.28	0.031	0.27	160	0.2	0.04	0.665	25.00
NMR	PF	1.06	0.053	1.08	160	1	0.12	0.8	28.57
	EC (kJ)	18.52	0.602	18.28	142	0.2	0.1	0.2	11.09
	SR (μm)	0.29	0.033	0.28	160	0.2	0.046	0.65	22.22
BOA	PF	1.06	0.055	1.08	160	1	0.12	0.8	28.57
	EC (kJ)	18.22	0.578	18.00	140	0.2	0.1	0.2	12.45
	SR (μm)	0.30	0.035	0.29	160	0.2	0.046	0.64	19.44
BMOA	PF	0.95	0.037	0.97	160	0.63	0.09	0.65	15.48
	EC (kJ)	18.64	0.604	18.37	145	0.29	0.11	0.2	10.65
	SR (μm)	0.31	0.034	0.30	147	0.27	0.056	0.75	16.67
AZOA	PF	1.06	0.590	1.08	160	1	0.12	0.8	28.57
	EC (kJ)	18.34	0.529	18.12	140	0.21	0.1	0.2	11.87
	SR (μm)	0.30	0.038	0.28	160	0.2	0.046	0.65	22.22

Based on the results, the corresponding convergence diagrams are developed in Figure 5.2 for the three responses under consideration. These convergence diagrams prove the superiority of FMA over the other metaheuristics in arriving at the optimal solutions with minimum number of iterations. With respect to convergence speed, the performance of BMOA is the worst, showing poor progress towards the optimal response values.

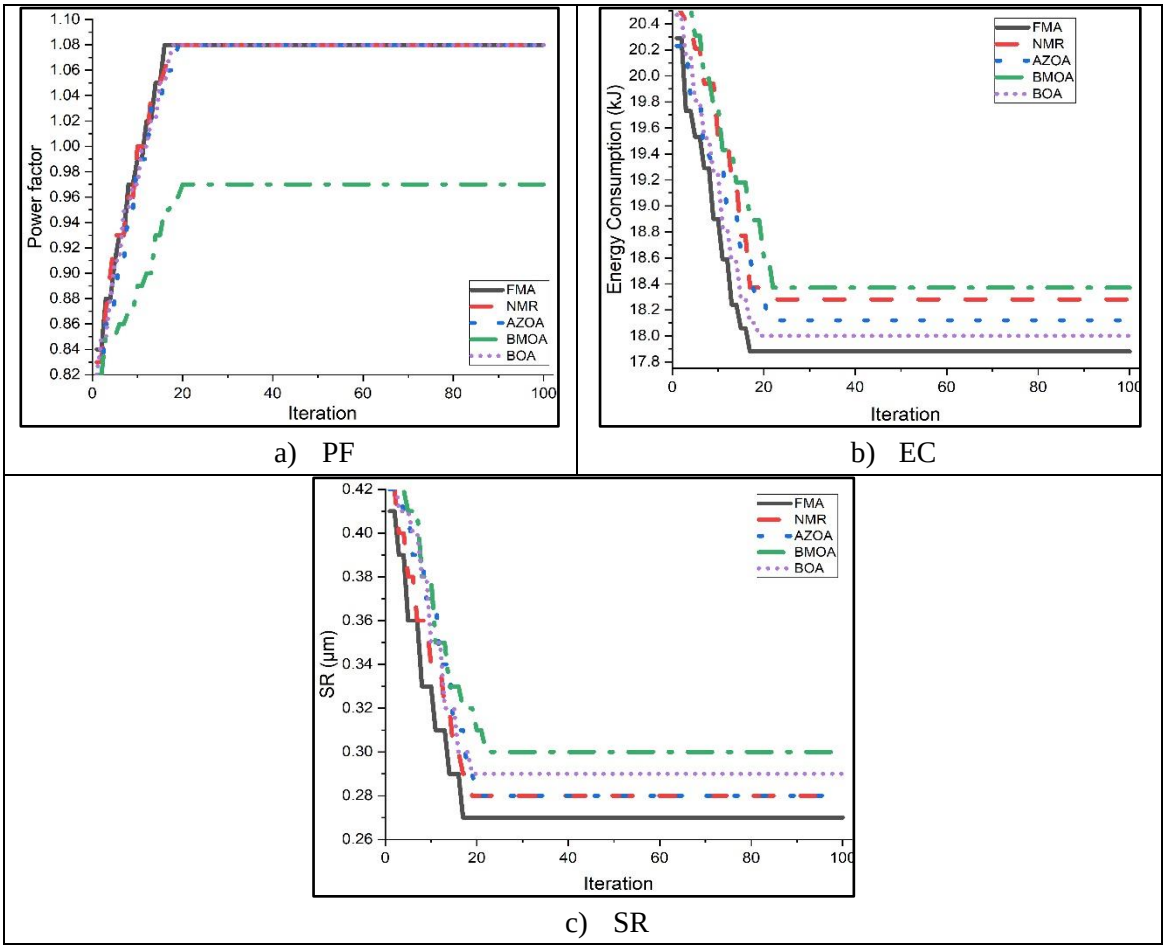


Figure 5.2 Convergence diagrams for the mating behavior-based metaheuristic algorithms

The box plots, portrayed in Figure 5.3 for the three responses against the five algorithms, also validate supremacy of FMA over its competitors in regard of variability of the derived solutions. Among these algorithms, FMA and AZOA have the minimum outlier spread, while NMR, BOA and BMOA have maximum outlier spread for all the responses under consideration. It indicates that FMA and AZOA require least number of iterations for obtaining the optimal solutions. The box plot also depicts the exact distribution of the solutions for each algorithm along with the corresponding skewness values.

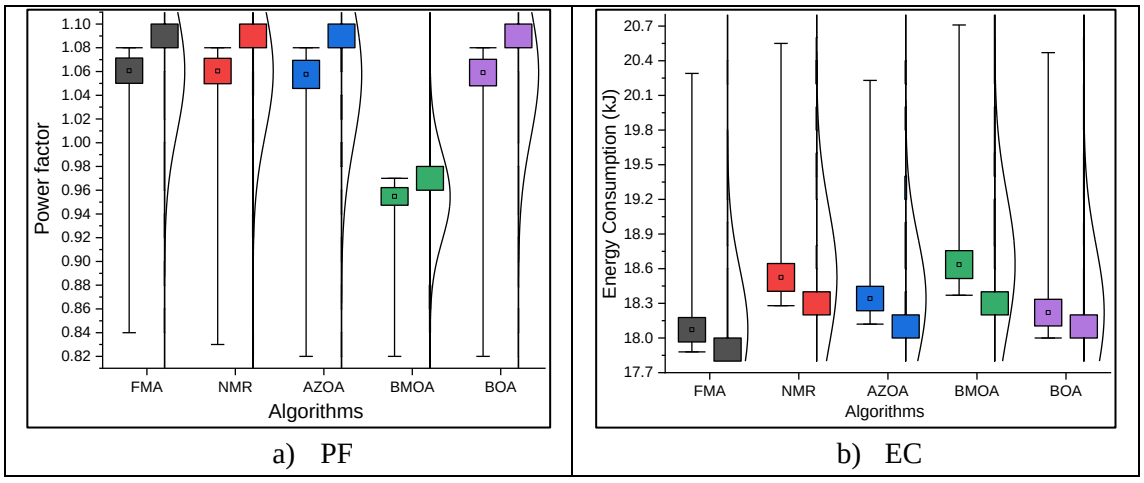
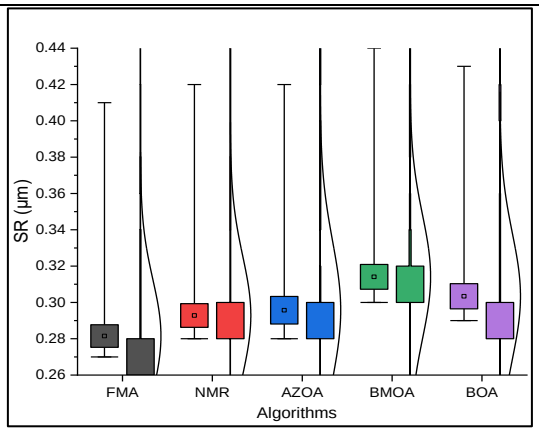


Figure 5.3 Box plots for the mating behavior-based metaheuristic algorithms



c) SR

Figure 5.3 Contd.

It can be noticed from Table 5.2 that single-objective optimization provides completely contradictory settings of the dry milling parameters in attaining maximum PF, and minimum EC and SR values, which would be quite impractical to maintain in a single dry milling setup. Single-objective optimization only concentrates on an individual quality characteristic (response) at a time, but multi-objective optimization can identify a unique combination of the process parameters leading to concurrent optimization of all the responses, fulfilling the productivity requirements of the machining processes.

Figure 5.4 exhibits the average computational time taken by each of the mating behavior-based metaheuristics for each of the responses while solving the corresponding single-objective optimization problems. It reveals that both FMA and BMOA are computationally efficient excelling over NMR, BOA and AZAO. Thus, based on quality of the derived solutions, convergence effort and average computational time, it can be concluded that FMA performs best against the other mating behavior-based algorithms.

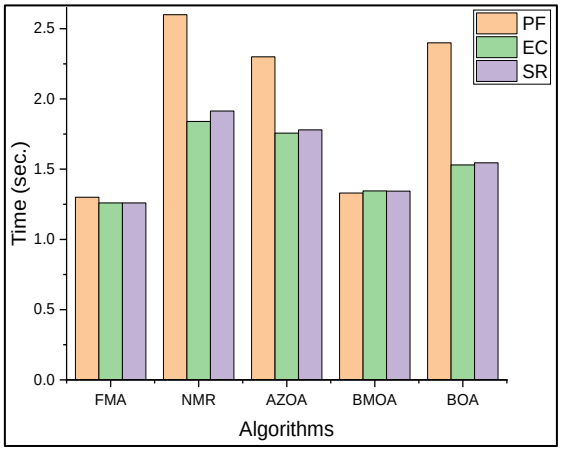


Figure 5.4 Average computation time for single-objective optimization

5.2 Multi-objective optimization of the dry milling process

To solve the corresponding multi-objective optimization problem, the traditional WSMO approach is deployed, followed by development of the Pareto optimal front. For WSMO, the following

composite objective function is framed while assigning equal importance to all the responses.

$$\text{Min}(Z)=\left[-\frac{w_1\times Y(PF)}{PF_{\max}}+\frac{w_2\times Y(EC)}{EC_{\min}}+\frac{w_3\times Y(SR)}{SR_{\min}}\right] \tag{5.4}$$

where $Y(PF)$, $Y(EC)$ and $Y(SR)$ are the RSM-based equations for PF, EC and SR respectively; $PF_{\max}$, $EC_{\min}$ and $SR_{\min}$ are the single-objective optimization-based results for PF, EC and SR respectively; and w_1, w_2 and w_3 are the weights allocated to PF, EC and SR respectively. However, in this method, values of the best solutions largely depend on the weights assigned to the responses. If their relative importance is not known beforehand, it would be a wise decision to treat all the responses having equal significance. But, it may not be the real situation. Depending on the end use requirements, those responses may have dissimilar importance. Table 5.3 provides the results of multi-objective optimization of the considered dry milling operation. It can be interestingly noticed that FMA supersedes NMR, BOA, BMOA and AZOA with respect to the minimum value of the composite objective function. Thus, to simultaneously optimize all the responses, CS = 160 m/min, DOC = 0.2 mm, FR = 0.085 mm/z and Nr = 0.6 mm would provide the optimal parametric combination. A comparative analysis reveals that application of almost all the considered optimizers results in similar improvement in PF as 3.57% (2.38% for BOA and 1.19% for BMOA). On the other hand, for EC, the performance improvement ranges between 2.33% and 7.05%, and for SR, all the optimizers, except AZOA (2.78%), provide 5.56% improvement in its value, against that obtained by Nguyen et al. [279].

Table 5.3 Multi-objective optimization results

Optimizer	Response	Optimal	Min z	Milling parameters				% improvement
				x_1 (m/min)	x_2 (mm)	x_3 (mm/z)	x_4 (mm)	
Nguyen et al. [279]	PF	0.84	-	160	0.42	0.09	0.8	-
	EC (kJ)	20.56						
	SR (μm)	0.36						
FMA	PF	0.87	1.36	160	0.2	0.085	0.6	3.57
	EC (kJ)	19.11						7.05
	SR (μm)	0.34						5.56
NMR	PF	0.87	1.41	160	0.2	0.085	0.5	3.57
	EC (kJ)	19.53						5.01
	SR (μm)	0.34						5.56
BOA	PF	0.86	1.45	160	0.2	0.09	0.6	2.38
	EC (kJ)	20.08						2.33
	SR (μm)	0.34						5.56
BMOA	PF	0.85	1.47	160	0.2	0.085	0.45	1.19
	EC (kJ)	19.65						4.43
	SR (μm)	0.34						5.56
AZOA	PF	0.87	1.37	160	0.2	0.08	0.5	3.57
	EC (kJ)	19.11						7.05
	SR (μm)	0.35						2.78

To resolve the problem of response weight dependency in WSMO approach affecting the derived optimal solutions, the corresponding Pareto optimal front consisting of a set of non-dominated solutions, as shown in Figure 5.5, is developed for FMA technique. It thus provides a compromise between PF, EC and SR, resulting in maximum performance of the said dry milling process.

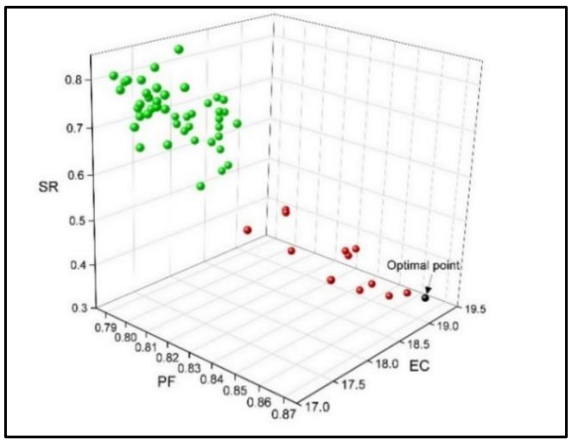


Figure 5.5 3D Pareto optimal front for FMA

To understand influences of the dry milling parameters on the responses and interactions between them, the related parallel plot is developed in Figure 5.6 for FMA. In Figure 5.6, the solutions represented by the red and green colors validate superiority of FMA in the lower and upper portions of the Pareto front while the optimal solutions are present at the lower portion of the Pareto front. The black point in the Pareto front and black line in the parallel plot show the optimal performance of the said dry milling process with respect to maximum PF, and minimum EC and SR. Thus, a specific combination of CS =160 m/min, DOC = 0.2 mm, FR = 0.085 mm/z and Nr = 0.6 mm would result in achieving the corresponding values of PF, EC and SR as 0.87, 19.11kJ and 0.34 μm respectively for FMA.

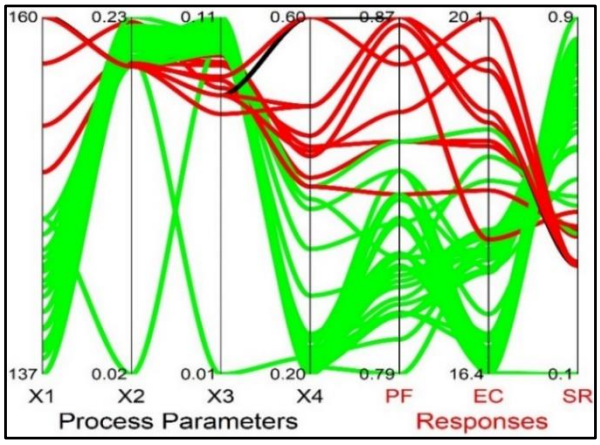


Figure 5.6 Parallel plot for FMA

In Figure 5.7, the average computational time is compared for the adopted metaheuristics based on 30 independent runs for each algorithm while solving the corresponding multi-objective optimization problem. It can be observed that both NMR and AZOA consume maximum computational time, while FMA is the most computationally efficient technique. Thus, based on average computational time, the

considered mating behavior-based algorithms can be ranked as FMA > BMOA > BOA > AZOA > NMR.

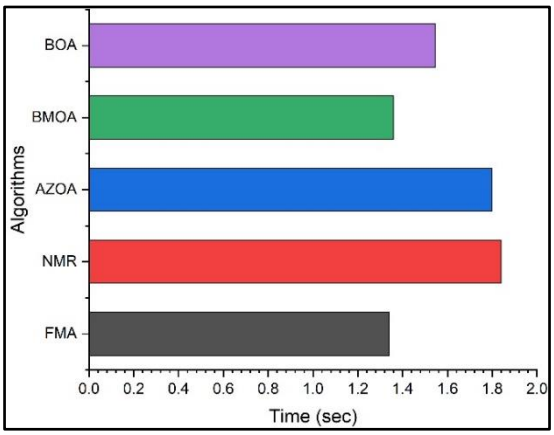


Figure 5.7 Average computational time for the multi-objective optimization problem

5.3 Performance analysis

Like the earlier section, the corresponding HV and SP values are also computed, as shown in Table 5.4. Among the mating behavior-based metaheuristics, FMA has the maximum HV value of 0.2468 and minimum SP value of 0.066. Thus, considering these two metrics, it can be unveiled that FMA has an impressive performance, outperforming its other competitors, and demonstrates its excellent capability in solving high-dimensional optimization problems.

Table 5.4 HV and SP values

Algorithm	FMA	NMR	BOA	BMOA	AZOA
HV	**0.2468**	0.2413	0.2349	0.231	0.24
SP	**0.066**	0.068	0.071	0.074	0.069

Similarly, the results of Friedman's mean rank test, as exhibited in Table 5.5, also prove the excellence of FMA over the other mating behavior-based algorithms, in optimizing the said dry milling operation of SS 304 work material. For all the responses under consideration, BMOA performs worst.

Table 5.5 Results of Friedman's mean rank test

Algorithm	Response	FMA	NMR	BOA	BMOA	AZOA
Average	PF	1.06	1.06	1.06	0.95	1.06
Rank sum		238	240	255	498	269
Average rank		2.38	2.4	2.55	4.98	2.69
Rank		**1**	**2**	**3**	**5**	**4**
Average	EC	18.07	18.52	18.22	18.64	18.34
Rank sum		102	402	205	496	295
Average rank		1.02	4.02	2.05	4.96	2.95
Rank		**1**	**4**	**2**	**5**	**3**
Average	SR	0.28	0.29	0.30	0.31	0.30
Rank sum		100.5	243.5	391.5	493.5	271
Average rank		1.005	2.435	3.915	4.935	2.71
Rank		**1**	**2**	**4**	**5**	**3**

Finally, the results of Wilcoxon rank sum test, as provided in Table 5.6, reassure the superiority for FMA over its competitors in solving this multi-objective optimization problem at 1% significance level.

Table 5.6 Wilcoxon rank sum test results

FMA vs		NMR	BOA	BMOA	AZOA
R–	PF	10.5	1	1	10.5
R+		47.5	135	135	47.5
p-value		2.51E-03	5.40E-04	5.40E-04	2.51E-03
R–	EC	55	3.5	3.5	55
R+		130	121.5	121.5	130
p-value		9.72E-21	1.76E-20	1.76E-20	9.72E-21
R–	SR	34	86	86	34
R+		126	104	104	126
p-value		4.36E-21	7.51E-21	7.51E-21	4.36E-21

5.4 Results and discussion

This research work investigates application of five mating behavior-based metaheuristic algorithms for parametric optimization of dry milling machining of SS 304 work material. They achieve improvements ranging from 15.48 to 28.57% for PF, 10.65 to 13.04% for EC, and 16.67 to 25% for SR in single-objective optimization; and 1.19 to 3.57%, 2.33 to 7.05%, and 2.78 to 5.56% improvements for PF, EC and SR, respectively in multi-objective optimization, compared to the results reported by Nguyen et al. [279]. Among these algorithms, FMA consistently delivers the best results, proving to be most efficient in both single- and multi-objective optimization of the said milling process. In single-objective optimization, it attains the optimal values of 1.08 for PF, 17.88 kJ for EC and 0.27 μm for SR. Furthermore, in multi-objective optimization, the corresponding optimal values are 0.87, 19.11 kJ and 0.34 μm for PF, EC and SR, respectively, at the best possible parametric combination of CS = 160 m/min, DOC = 0.2 mm, FR = 0.085 mm/z and Nr = 0.6 mm. FMA also excels over the others with reference to speed of convergence and computing time. For multi-objective optimization, it takes just 1.34 sec to find out the global optimal solution, reducing the runtime by 37.31, 34.33, 1.49 and 15.3% against NMR, AZOA, BMOA and BOA techniques, respectively. Its superiority is further confirmed using SP and HV values, and validated through Friedman's mean rank and Wilcoxon rank sum tests.

Finally, the said green machining process (dry milling) is optimized (single- and multi-objective) using other state-of-the-art metaheuristics, i.e. GA, PSO, ABC, ACO and TLBO. The improvements in responses (in percentage values), along with the average computation time for those algorithms are depicted in Table 5.7. It can be revealed from Table 5.7 that all the considered mating behavior-based techniques, especially FMA, outperform those algorithms while solving both the single- and multi-objective optimization problems under consideration.

Table 5.7 GA, PSO, ABC, ACO and TLBO-based optimization results

Optimizer	Technique	PF	EC	SR	Average time (s)
GA	Single-objective	27.38	11.72	25.00	1.28
PSO		28.57	9.00	25.00	1.14
ABC		28.57	6.13	19.44	1.45
ACO		28.57	10.46	22.22	1.65
TLBO		28.57	12.60	22.22	1.33
GA	Multi-objective	3.57	6.61	5.56	1.48
PSO		3.57	6.61	5.56	1.25
ABC		2.38	4.43	2.78	1.92
ACO		1.19	2.33	2.78	2.10
TLBO		2.38	5.01	5.56	1.79

6. OPTIMIZATION OF A LASER BEAM DRILLING PROCESS USING PREYING BEHAVIOR-BASED METAHEURISTIC ALGORITHMS

Selecting the most appropriate algorithm from a set of preying behavior-based metaheuristics can help the process engineers achieve global or near-global optimization of LBD processes. The procedural steps for parametric optimization of an LBM process are depicted in Figure 6.1. Focusing on the identified research gap and attempting to bridge it, this work focuses on the application of five preying behavior-based metaheuristic algorithms in the form of SFO, HHO, AO, BA and GWO to solve both single- and multi-objective optimization problems for an LBD process while assigning equal importance to all the responses under consideration. This work aims to identify the optimal combinations of the considered LBD parameters to achieve the desired micro-hole characteristics, compare the optimization performance of the five algorithms with respect to quality and consistency of the derived solutions, convergence speed and computational effort, and apply Friedman's mean rank and Wilcoxon rank sum tests to determine the best performing algorithm. To derive the optimal multi-objective solutions and unique parametric settings for the LBD process, the corresponding Pareto optimal fronts are subsequently developed, while parallel plots aid in visualizing the best parameter and response combinations. Additionally, SP and HV metrics are used to evaluate effectiveness of the preying behavior-based algorithms for the considered LBD process.

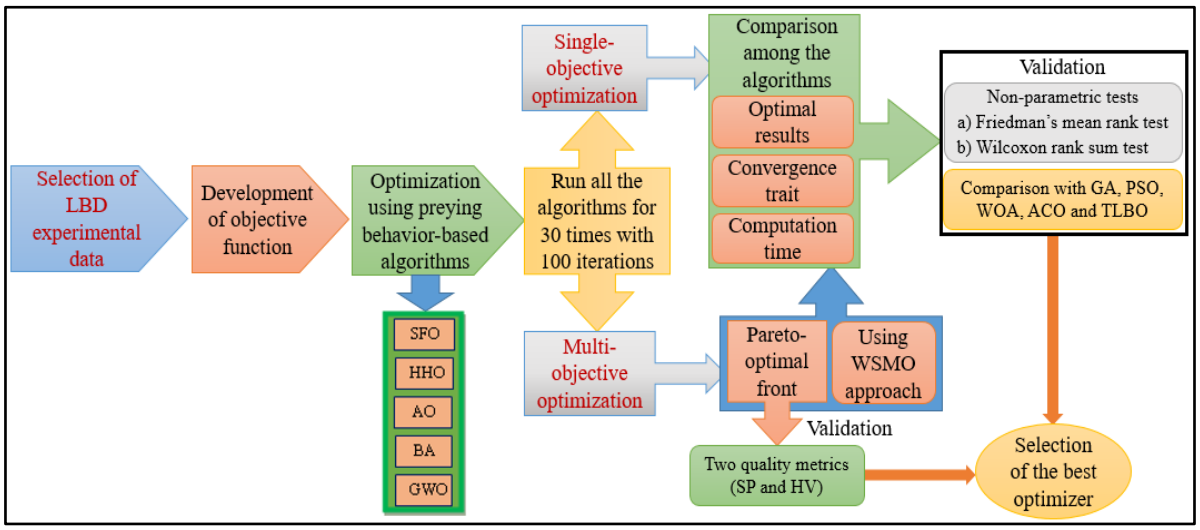


Figure 6.1 Flow diagram for optimization of the LBD process

To assess the relative efficiency of all the five preying behavior-based metaheuristics, an illustrative example, i.e. CO_2-based micro-drilling on polycarbonate work material is considered here. The corresponding codes are written in Python Spyder (Anaconda 3) and are run in a Windows 10 Pro, Intel(R) Core(TM) i5-7500 processor, 8.00 GB RAM platform. For all the algorithms, the population size and number of iterations are maintained as 50 and 100 respectively, (i.e. each of them is run 30 times independently, total number of trials = 30 runs×100 iterations = 3000) to allow uniformity during the comparative performance analysis.

Treating Pl (x_1) (in W), ET (x_2) (in sec) and FPP (x_3) (in mm) as the input parameters, and Hh (in mm), Hd (in µm) and HAZ (in µm) as the responses, Abdulwahab et al. [280] performed 17 laser micro-drilling experiments on polycarbonate work material using a CNC-based CO_2 LBD setup. During those experiments, values of the input parameters were varied at three different levels, i.e. Pl = {2, 3, 4 W}, ET = {0.1, 0.15, 0.2 sec} and FPP = {-4, -2, 0 mm}. The FPP was treated as 0 at the position of the workpiece's upper area, and the lower and upper FPP surfaces were considered as negative and positive respectively. After each micro-drilling operation, the corresponding values of Hh, Hd and HAZ were measured using a microscope at 50X magnification. To avoid chances of any measurement error, all the readings were noted three times, and their average was calculated for each of the responses under consideration. The dimensions of the polycarbonate workpiece were 7 cm×3 cm×1 mm, and the wavelength of CO_2 laser was set at 10.6 µm from JK series. It is worthwhile to mention here that Hh is a larger-the-better characteristic, while Hd and HAZ are small-the-better responses. The employed BBD plan along with the measured response values is exhibited in Table 6.1. The 3-factor, 3-level BBD plan is based on 3^3 factorial design, consisting of 17 experiments, including 12 factorial points and 5 replicates at the center point for estimation of pure error sum of squares.

Table 6.1 Experimental design plan and response values [280]

Exp. No.	Pl (x_1)	ET (x_2)	FPP (x_3)	Hd (µm)	HAZ (µm)	Hh (mm)
1	2	0.15	−4	500	155	0.4
2	2	0.2	−2	350	102	0.7
3	2	0.1	−2	320	80	0.4
4	2	0.15	0	320	80	0.8
5	3	0.1	−4	565	118	0.4
6	3	0.2	−4	618	206	0.82
7	3	0.15	−2	340	100	0.65
8	3	0.15	−2	341	100.2	0.651
9	3	0.15	−2	341	100	0.65
10	3	0.15	−2	340.7	100.05	0.65
11	3	0.15	−2	340	100	0.6502
12	3	0.1	0	331	80	0.8
13	3	0.2	0	400	101	1
14	4	0.15	−4	616	170	0.8
15	4	0.1	−2	352	101	1
16	4	0.2	−2	500	202	1
17	4	0.15	0	351	102	1

In order to optimize the considered micro-drilling operation on polycarbonate material using SFO, HHO, AO, BA and GWO techniques, the corresponding polynomial equations were developed by Abdulwahab et al. [280] using Design-Expert software (Ver. 13).

$$Y(Hh) = 0.6 + 0.003x_1 - 2.08x_2 + 0.2x_3 - 1.5x_1x_2 - 0.0025x_1x_3 - 0.55x_2x_3 + 0.05x_1^2 + 25.9x_2^2 + 0.009x_3^2 \quad (6.1)$$

$$Y(Hd) = 771.5 - 93.2x_1 - 5242.6x_2 + 71.9x_3 + 590x_1x_2 - 10.6x_1x_3 + 40x_2x_3 + 4.1x_1^2 + 14342x_2^2 + 25.5x_3^2 \quad (6.2)$$

$$Y(HAZ) = 347.45 - 102.85x_1 - 2182x_2 + 20.475x_3 + 395x_1x_2 + 0.875x_1x_3 - 167x_2x_3 + 10.85x_1^2 + 4140x_2^2 + 3.9x_3^2 \quad (6.3)$$

6.1 Single-objective optimization of the LBD process

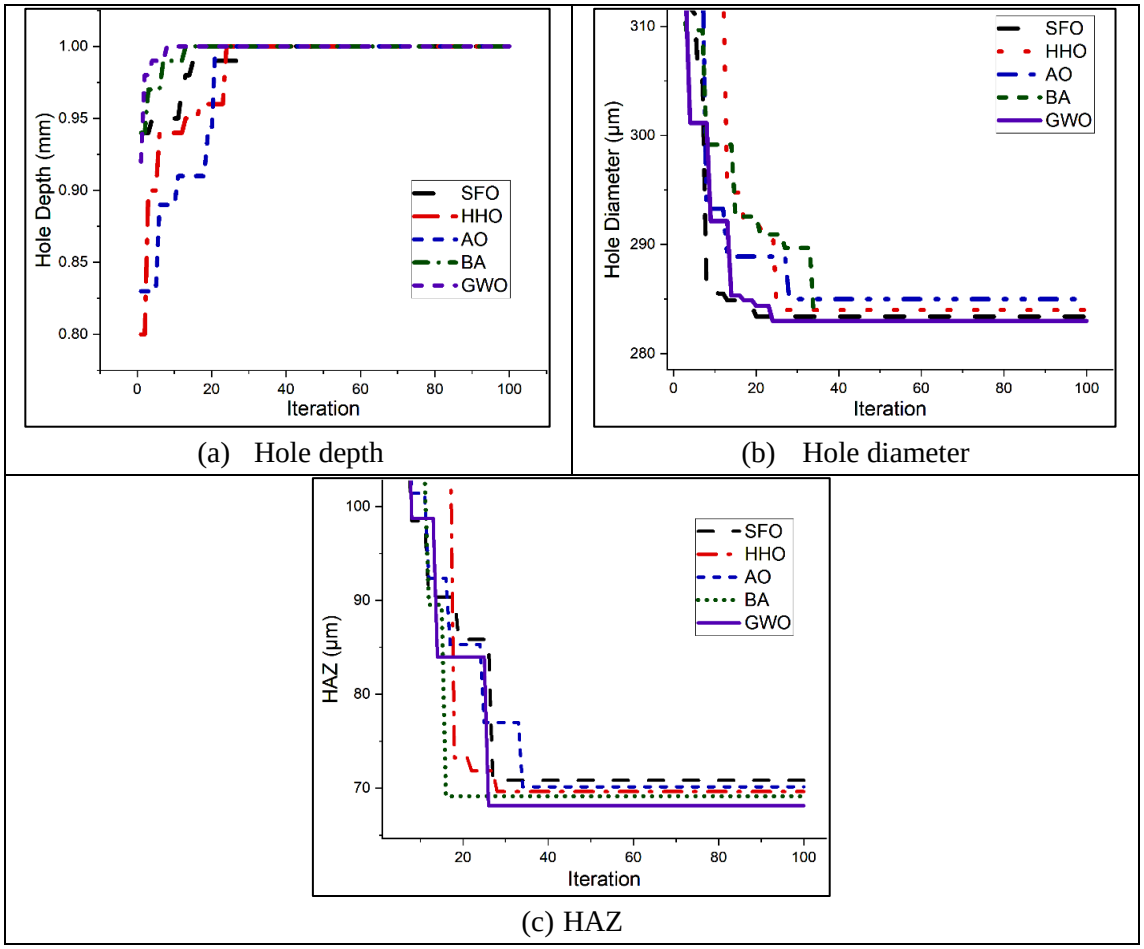
In single-objective optimization, based on the above-mentioned polynomial equations, all the considered responses are individually optimized within the given constraints of the process variables, i.e. 2 W $\leq x_1 \leq$ 4 W, 0.1 sec $\leq x_2 \leq$ 0.2 sec and –4 mm $\leq x_3 \leq$ 0 mm. The responses derived using the five metaheuristics along with the values of micro-drilling parameters are exhibited in Table 6.2. It can be interestingly noticed that during micro-drilling operation on polycarbonate material with 1 mm thickness, full length of the Hh is achieved in SFO, HHO, AO, BA and GWO techniques which exactly corroborate with the observations of Abdulwahab et al. [280]. It can also be revealed that there are 18.8-19.37% improvements in the value of Hd response as compared to that attained by Abdulwahab et al. [280]. With respect to Hd response, AO is unable to determine its optimal value. On the other hand, the HAZ value is reduced by 33.33% irrespective of the optimization technique employed against that measured by Abdulwahab et al. [280].

Table 6.2 Results of single-objective optimization

Optimizer	Response	Mean	SD	Optimal	Process parameters			% improvement
					x_1	x_2	x_3	
Abdulwahab et al. [280]	Hh (mm)	-	-	1	4	0.15	0	-
	Hd (µm)			351				
	HAZ (µm)			102				
SFO	Hh (mm)	0.99	0.0167	1	4	0.2	0	0
	Hd (µm)	284.06	3.264	283	2	0.14	-1.1	19.37
	HAZ (µm)	69.33	2.493	68.14	2	0.168	0	33.33
HHO	Hh (mm)	0.98	0.0358	1	4	0.2	0	0
	Hd (µm)	290.8	4.227	283	2	0.14	-1.1	19.37
	HAZ (µm)	77.7	3.100	68.14	2	0.168	-0.01	33.33
AO	Hh (mm)	0.96	0.0455	1	4	0.2	0	0
	Hd (µm)	288	3.659	285	2	0.14	-0.8	18.8
	HAZ (µm)	70.1	2.802	68.14	2	0.168	0	33.33
BA	Hh (mm)	0.997	0.0218	1	4	0.2	0	0
	Hd (µm)	288	5.287	283	2	0.14	-1.1	19.37
	HAZ (µm)	68.8	3.073	68.14	2	0.168	0	33.33
GWO	Hh (mm)	0.99	0.0086	1	4	0.2	0	0
	Hd (µm)	284	3.099	283	2	0.14	-1.1	19.37
	HAZ (µm)	70.27	2.155	68.14	2	0.168	0	33.33

Figure 6.2 shows the convergence diagrams for all the three responses for SFO, HHO, AO, BA and GWO techniques which depict that except AO, the remaining algorithms take only a few iterations to reach the optimal solutions. But, GWO supersedes others with respect to the rate of convergence. To

study variations in the derived solutions in single-objective optimization, the corresponding box plots of all the five metaheuristics are portrayed in Figure 6.3 based on the results of 30 independent runs of each algorithm. It can be noticed from Table 6.2 that HHO has the highest variations in the predicted values of Hd and HAZ, whereas, SFO performs comparatively better for the three responses. From the developed box plots, it can be revealed that HHO has the highest mean and SD values for Hd and HAZ. It is also interesting to note that among all the algorithms, GWO performs exceptionally well with respect to variations (SD values) in the predicted values of the responses under consideration. It implies that the relative performance of the five algorithms, specially GWO should be analyzed in depth while solving the consequent multi-objective optimization problem. Figure 6.4 shows the average computation time required by each of the algorithms for single-objective optimization of the responses. With respect to relative computation time, HHO, BA and GWO are more efficient than SFO and AO techniques.



(a) Hole depth

(b) Hole diameter

(c) HAZ

Figure 6.2 Convergence diagrams for the preying behavior-based metaheuristic algorithms

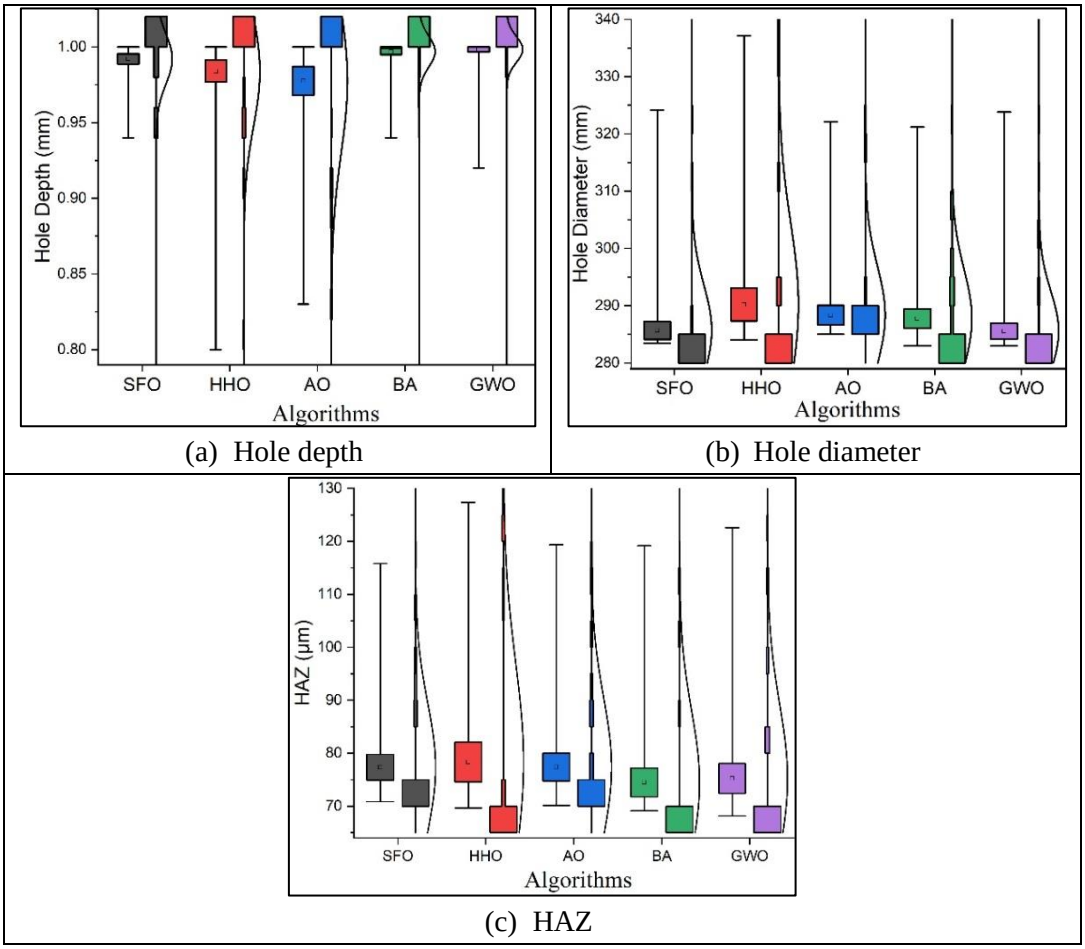


Figure 6.3 Box plots for all the algorithms in single-objective optimization

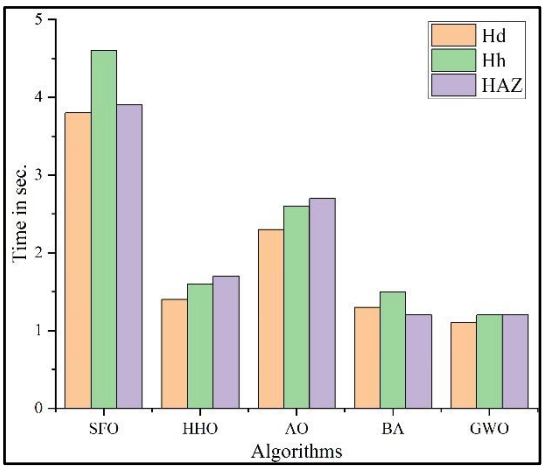


Figure 6.4 Average computation time for each algorithm in single-objective optimization problem

6.2 Multi-objective optimization of the LBD process

It can be noted that in single-objective optimization, all the algorithms identify dissimilar settings of the LBD parameters for each of the responses which are quite impractical to maintain in a real-time setup. It is thus required to derive a unique intermix of all the LBD parameters leading to simultaneous optimization of the responses. In this work, the WSMO model is initially applied, followed by framing of the related Pareto optimal front. The corresponding multi-objective function for WSMO is developed

as follows:

$$\text{Min}(Z) = -w_1 \times \frac{Y(Hh)}{Hh_{\max}} + w_2 \times \frac{Y(Hd)}{Hd_{\min}} + w_3 \times \frac{Y(HAZ)}{HAZ_{\min}} \tag{6.4}$$

where $Y(Hh)$, $Y(Hd)$ and $Y(HAZ)$ are the RSM-based equations formulated for Hh, Hd and HAZ respectively; $Hh_{\max}$, $Hd_{\min}$ and $HAZ_{\min}$ are the optimal response values derived based on single-objective optimization results; and w_1, w_2 and w_3 are the relative weights assigned to Hh, Hd and HAZ respectively. Table 6.3 provides the multi-objective optimization results which reveal that all the five algorithms are capable of achieving better response values against those attained by Abdulwahab et al. [280]. It is worthwhile to mention here that for multi-objective optimization, equal importance is assigned to all three responses. The performance of these algorithms is almost similar providing 11.78% (12.25% in case of GWO) and 15.24% improvements in Hd and HAZ respectively. Figure 6.7 exhibits the average computational time required by each of the algorithms in solving the considered multi-objective optimization problem. It reveals that like single-objective optimization problem, HHO, BA and GWO excel over SFO and AO with respect to average computational time. Thus, based on average computational time, these algorithms can be ranked as GWO > BA > HHO > AO > SFO. It can be concluded from Table 6.3 that an optimal combination of Pl = 3.95 W, ET = 0.1 sec and FPP = -0.91 mm would lead to simultaneous achievement of the maximum value of Hh, and minimum values of Hd and HAZ for the said LBD process.

Table 6.3 Results of multi-objective optimization

Optimizer	Response	Optimal	Min z	Process parameter			% improvement
				x_1	x_2	x_3	
Abdulwahab et al. [280]	Hh (mm)	1	-	4	0.15	0	-
	Hd (µm)	351					
	HAZ (µm)	102					
SFO	Hh (mm)	1	2.59	3.95	0.1	-0.9	-
	Hd (µm)	309.66					11.78
	HAZ (µm)	86.45					15.24
HHO	Hh (mm)	1	2.59	3.95	0.1	-0.9	-
	Hd (µm)	309.66					11.78
	HAZ (µm)	86.45					15.24
AO	Hh (mm)	1	2.59	3.95	0.1	-0.9	-
	Hd (µm)	309.66					11.78
	HAZ (µm)	86.45					15.24
BA	Hh (mm)	1	2.59	3.95	0.1	-0.9	-
	Hd (µm)	309.66					11.78
	HAZ (µm)	86.45					15.24
GWO	Hh (mm)	1	2.59	3.95	0.1	-0.91	-
	Hd (µm)	308					12.25
	HAZ (µm)	86.45					15.24

To address the limitations of response weight dependency in the WSMO approach, a 3D Pareto optimal front is developed for GWO technique, as shown in Figure 6.5. The distribution and spread of the Pareto front highlight GWO's effectiveness in exploring the solution space and identifying meaningful trade-offs among the three objectives for simultaneous optimization. In this plot, green points at the upper portion and red points at the lower portion represent non-dominated solutions that form the Pareto optimal front, offering a range of optimal trade-offs. The 'Optimal point', marked in black, represents the best compromised solution, further validating GWO's robustness in multi-objective optimization.

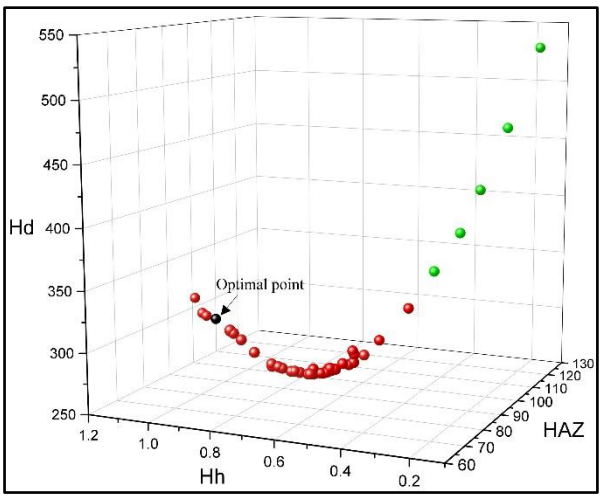


Figure 6.5 3D Pareto optimal front for GWO

Figure 6.6 exhibits the parallel plot developed for this example which would help in investigating influences of the LBD parameters on the responses, and interaction effects between them. The considered LBD parameters, i.e. Pl, ET and FPP have varying settings to achieve the desired responses. But, it can be ensured that considering application of GWO algorithm for multi-objective optimization of the said LBD process, a unique combination of Pl = 3.95 W, ET = 0.1 sec and FPP = -0.91 mm would result in obtaining the corresponding Hh, Hd and HAZ values as 1 mm, 308 µm and 86.45 µm, respectively.

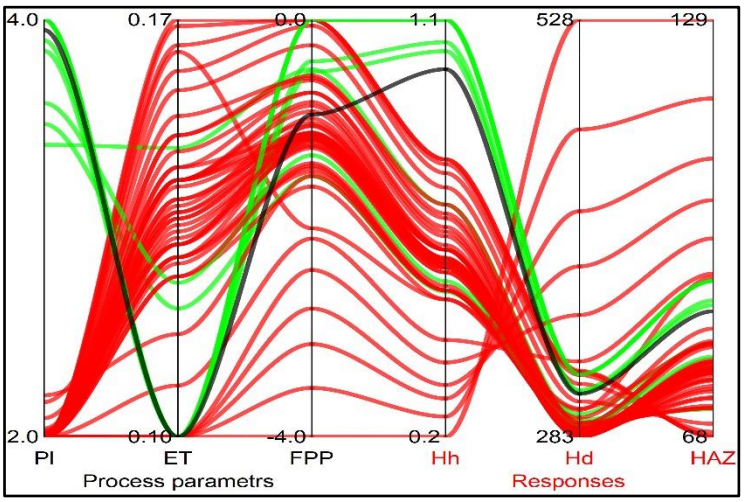


Figure 6.6 Parallel plot for GWO

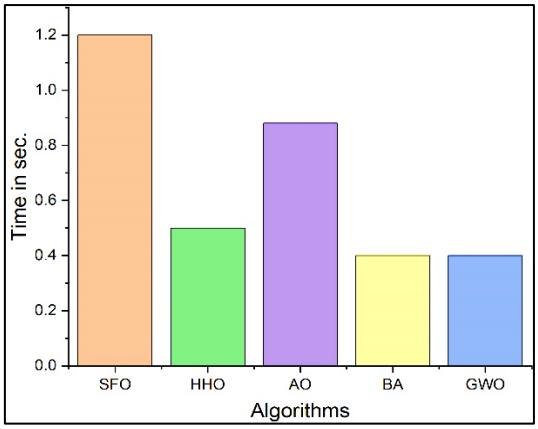


Figure 6.7 Average computational time for multi-objective optimization problem

6.3 Performance analysis

Like the earlier sections, the corresponding HV and SP values are also computed, as shown in Table 6.4. Among the mating behavior-based metaheuristics, GWO has the maximum HV value of 14.64 and minimum SP value of 5.06. Thus, considering these two metrics, it can be unveiled that GWO has an impressive performance, outperforming its other competitors, and demonstrates its excellent capability in solving high-dimensional optimization problems.

Table 6.4 HV and SP values

Algorithm	SFO	HHO	AO	BA	GWO
HV	14.61	14.55	14.58	14.59	**14.64**
SP	5.09	5.12	5.13	5.08	**5.06**

To have a critical analysis of the optimization performance of the five metaheuristics, Friedman's mean rank test is performed at 1% level of significance, as provided in Table 6.5. It is a nonparametric test, assuming that the input data would not follow normal distribution. It can be concluded from Table 6.5 that GWO supersedes the other algorithms with respect to average ranks for all the responses and is the best performing algorithm among the preying behavior-based metaheuristics under consideration while optimizing the said LBD process.

Table 6.5 Results of Friedman's mean rank test

Algorithm	Response	SFO	HHO	AO	BA	GWO
Average	Hh	0.9921	0.9815	0.9984	0.997	0.9621
Rank sum		326.5	274.5	370	361	168
Average rank		3.265	2.745	3.7	3.61	1.68
Rank		**3**	**2**	**5**	**4**	**1**
Average	Hd	284.4446	290.8578	288.0437	287.8985	284.0642
Rank sum		184.5	329	425	382	179.5
Average rank		1.845	3.29	4.25	3.82	1.795
Rank		**2**	**3**	**5**	**4**	**1**
Average	HAZ	69.33076	77.78031	70.1685	70.27636	68.8411
Rank sum		234.5	377.5	358	329.5	200.5
Average rank		2.345	3.775	3.58	3.295	2.005
Rank		**2**	**5**	**4**	**3**	**1**

Finally, the results of Wilcoxon rank sum test, as provided in Table 6.6, reassure the superiority for GWO over its competitors in solving this multi-objective optimization problem at 1% significance level.

Table 6.6 Wilcoxon rank sum test results

GWO vs		SFO	HHO	AO	BA
R–	Hh	390	741	278	57.5
R+		16	7	22	8.5
p-value		2.00E-04	3.70E-04	1.20E-04	0.02926
R–	Hd	400	214	547	352
R+		266	5	3	29
p-value		2.94E-01	1.80E-04	8.60E-04	1.40E-04
R–	HAZ	1066	473	1394	1128
R+		62	238	884	29
p-value		2.70E-04	1.00E-04	1.12E-01	1.79E-03

It has been experimented that all the LBD parameters, i.e. Pl, ET and FPP have significant effects on the responses under consideration. During LBD operation, due to Gaussian laser beam energy distribution, the energy input would vary at different areas within the beam diameter. The highest energy density would be at the center of the beam, whereas, the outer end would have the lowest energy. Increase in the laser exposure time with the material would lead to hole enlargement and deformation since the material would be exposed to greater heat for a longer time period. The Hd at focus would be lower than that at defocus due to decrease in laser beam spot diameter. At higher Pl and ET, HAZ would significantly increase due to having more energy from the laser source. Thus, higher Pl, lower ET and low-to-moderate FPP are recommended to achieve the most preferred values of the considered responses.

6.4 Results and discussion

It has already been mentioned that this work deals with application of five potential preying behavior-based metaheuristic algorithms for parametric optimization of an LBD process, i.e. CO_2 laser-based micro-drilling of polycarbonate material. For single-objective optimization, all the algorithms provide 18.8-19.37% improvements in Hd, and 33.33% improvement in HAZ as compared to those derived by Abdulwahab et al. [280]. All of them are able to identify the optimal Hh of exactly 1 mm. For multi-objective optimization, the corresponding improvements are 0%, 11.78% and 15.24% for all the algorithms for Hh, Hd and HAZ, respectively. But, GWO supersedes others with respect to faster convergence speed and computation time. It only takes 0.4 sec to arrive at the near global optimal solution, and accounts for 33.33%, 66.67%, 45.45% and 80% of the runtimes of SFO, HHO, AO and BA, respectively. Thus, based on the observations, it can be stated that higher Pl, lower ET and moderate-to-high FPP would result in simultaneous maximization of Hh, and minimization of Hd and HAZ during CO_2 laser-based micro-drilling of polycarbonate material. GWO also excels over the others with reference to speed of convergence and computing time. For multi-objective optimization, it takes

just 0.4 sec to find out the optimal solution, reducing the runtime by 200, 50, 120 and 25% against SFO, HHO, AO and BA techniques, respectively. Its superiority is further confirmed using SP and HV values, and validation through Friedman's mean rank and Wilcoxon rank sum tests.

The considered LBD operation is also optimized (single- and multi-objective) using several favored metaheuristic algorithms, like GA, PSO, WOA, ACO and TLBO. The derived results, as depicted in Table 6.7, are compared against those obtained using GWO algorithm. It can be noticed from Table 6.7 that GWO renders significant improvements (in percentage) in the optimal values of all the responses as well as average computational time.

Table 6.7 Comparison of performance of GWO against GA, PSO, WOA, ACO and TLBO

Optimizer	GA	PSO	WOA	ACO	TLBO	GA	PSO	WOA	ACO	TLBO
Technique	Single-objective					Multi-objective				
Hh	0	0	0	0	0	0	0	0	0	0
Hd	1.06	1.06	1.77	2.12	2.47	1.30	0.97	1.62	2.60	1.30
HAZ	3.08	2.33	3.74	4.49	2.29	1.93	1.91	3.23	3.38	3.99
Average time	87.1	61.3	106.4	112.9	65.2	187.5	135	155	170	135

7. OPTIMIZING AN ABRASIVE WATER JET MACHINING PROCESS BASED ON FORAGING BEHAVIOR-BASED METAHEURISTICS

This research work optimizes an AWJM process using five foraging behavior-based metaheuristic algorithms, i.e. DOA, AVOA, GOA, FOA and BSA, and their performance is contrasted with respect to accuracy and variability of the solutions, convergence speed and average computation time. Although all these metaheuristics have provided satisfactory solutions to many of the well-known benchmark functions having higher complexity, their performance has not been validated till date while optimizing any of the machining processes. To strengthen this comparison, Friedman's mean rank and Wilcoxon rank sum tests are also conducted to single out the best performing algorithm. The developed parallel plot would help in identifying the most suitable combinations of the input parameters and responses leading to process optimization. To execute these algorithms, the corresponding Python codes have been developed, and subsequently run on Spyder (Anacoda-3) in Windows-10 Pro, Intel® Core™ i3-6006U CPU @ 2.00 GHz processor and 4.00 GB RAM platform. All of them have been independently run 30 times and a maximum number of iterations has been taken as 100. Hence, the total number of trials is 3000 (i.e. 30×100), and the population size is maintained as 50.

Based on a CCD plan (with 6 replications at the central position) and using an AWJM setup (Model DIP 6D-2230), Balamurugan et al. [281] conducted 20 experiments on Lanthanum phosphate/Yttria ($LaPO_4/Y_2O_3$) composite to investigate effects of WJP, SOD and St on MRR, KA and SR (in terms of average roughness Ra) of the machined components. During the said AWJM operation, diameters of the orifice and tungsten carbide nozzle were kept as 0.25 mm and 0.67 mm respectively. Silicon carbide with 80-mesh size was considered as the abrasive particles. Based on the developed response surface plots, Balamurugan et al. [281] noticed significant influences of all the AWJM parameters on the responses under consideration. It was observed that MRR would increase with higher value of WJP, and lower values of SOD and St. On the other hand, lower values of WJP and TS, and moderate SOD value would lead to attainment of lower KA. Minimum SR values could be obtained at lower settings of WJP and St, and moderate setting of SOD. Different AWJM parameters along with their corresponding levels are depicted in Table 7.1. Based on the experimental dataset and using Design Expert software (Ver. 13), Balamurugan et al. [281] developed the following RSM-based equations interlinking different AWJM parameters and responses:

$$Y(MRR) = 0.099 - 0.00087\times A - 0.02502\times B - 0.000875\times C + 0.000003\times A^2 - 0.000343\times B^2 + 0.000002\times C^2 + 0.000123\times A\times B + 0.000001\times A\times C + 0.000084\times B\times C \quad (7.1)$$

$$Y(KA) = 2.58 - 0.0200\times A - 0.519\times B - 0.0012\times C + 0.000039\times A^2 + 0.0398\times B^2 + 0.000127\times C^2 + 0.001850\times A\times B + 0.000073\times A\times C - 0.001050\times B\times C \quad (7.2)$$

$$Y(SR) = -2.04 + 0.0190\times A - 0.369\times B + 0.0292\times C - 0.000027\times A^2 + 0.0513\times B^2 - 0.000172\times C^2 + 0.000963\times A\times B - 0.000023\times A\times C + 0.00110\times B\times C \quad (7.3)$$

Table 7.1 AWJM parameters and their levels [281]

AWJM parameter	Symbol	Level		
		-1	0	+1
WJP (Bar)	*A*	220	240	260
SOD (mm)	*B*	1	2	3
St (mm/min)	*C*	20	30	40

7.1 Single objective optimization of the AWJM process

Now, considering the above-mentioned RSM-based equations as the objective functions, and minimum and maximum values of the AWJM parameters as the lower and upper bounds of the design variables, all the three responses are individually optimized using the five considered foraging behavior-based algorithms. It is worthwhile to mention here that MRR is a beneficial response (its higher value is desired), and KA and SR are non-beneficial responses (their lower values are preferred). Table 7.2 shows the single-objective optimization results which also compare the relative performance of the adopted algorithms. It can be noticed from Table 7.2 that all the adopted metaheuristics result in substantial improvements in the considered response values, e.g. MRR (70.07-71.47%), KA (42.28-43.73%) and SR (9.22-10%), as compared to those recorded by Balamurugan et al. [281]. AVOA, GOA and BSA derive the optimal values of MRR and KA as 0.0369 g/s and 0.202° respectively, which are better than those obtained using DOA and FOA. For SR, AVOA is only able to identify its global minimum value as 1.152 µm. Based on the single-objective optimization results, it can be unveiled that AVOA provides the optimal values of all the responses under consideration.

Table 7.2 Single objective optimization results

Optimizer	Response	AWJM parameter			Mean	SD	Optimal	% improvement
		A (Bar)	*B* (mm)	*C* (mm/min)				
Balamurugan et al. [281]	MRR	220	1	40			0.02152	-
	KA	220	1.68	20			0.359	-
	SR	220	1.32	20			1.28	-
AVOA	MRR	260	3	20	0.0368	0.0003	0.0369	71.47
	KA	220	1.317	20	0.203	0.0022	0.202	43.73
	SR	220	1.67	20	1.15	0.0059	1.152	10.00
DOA	MRR	260	3	28.31	0.0366	0.0003	0.0367	70.54
	KA	234.6	1.293	20.16	0.2037	0.0022	0.2028	43.51
	SR	226.2	1.314	22.05	1.159	0.0065	1.158	9.53
FOA	MRR	260	2	25	0.0365	0.0003	0.0366	70.07
	KA	220	1.678	20	0.2087	0.0039	0.2072	42.28
	SR	220	1.319	20	1.166	0.0136	1.162	9.22
GOA	MRR	260	3	20	0.0366	0.0003	0.0369	71.47
	KA	220	1.67	20	0.2032	0.0018	0.202	43.73
	SR	220	1.317	20	1.161	0.0082	1.161	9.30
BSA	MRR	260	3	20	0.0366	0.0003	0.0369	71.47
	KA	220	1.67	20	0.2036	0.0023	0.202	43.73
	SR	220	1.318	20	1.16	0.0079	1.1583	9.51

Figure 7.1 plots the convergence diagrams of all the algorithms for the three responses. Except DOA and FOA, the remaining algorithms (AVOA, GOA and BSA) show rapid progress towards the corresponding optimal solutions after only a few iterations. The convergence patterns of AVOA, GOA and BSA are noticed to be almost similar for MRR and KA responses. On the other hand, while minimizing SR, AVOA, DOA, GOA and BSA have almost similar trends of convergence.

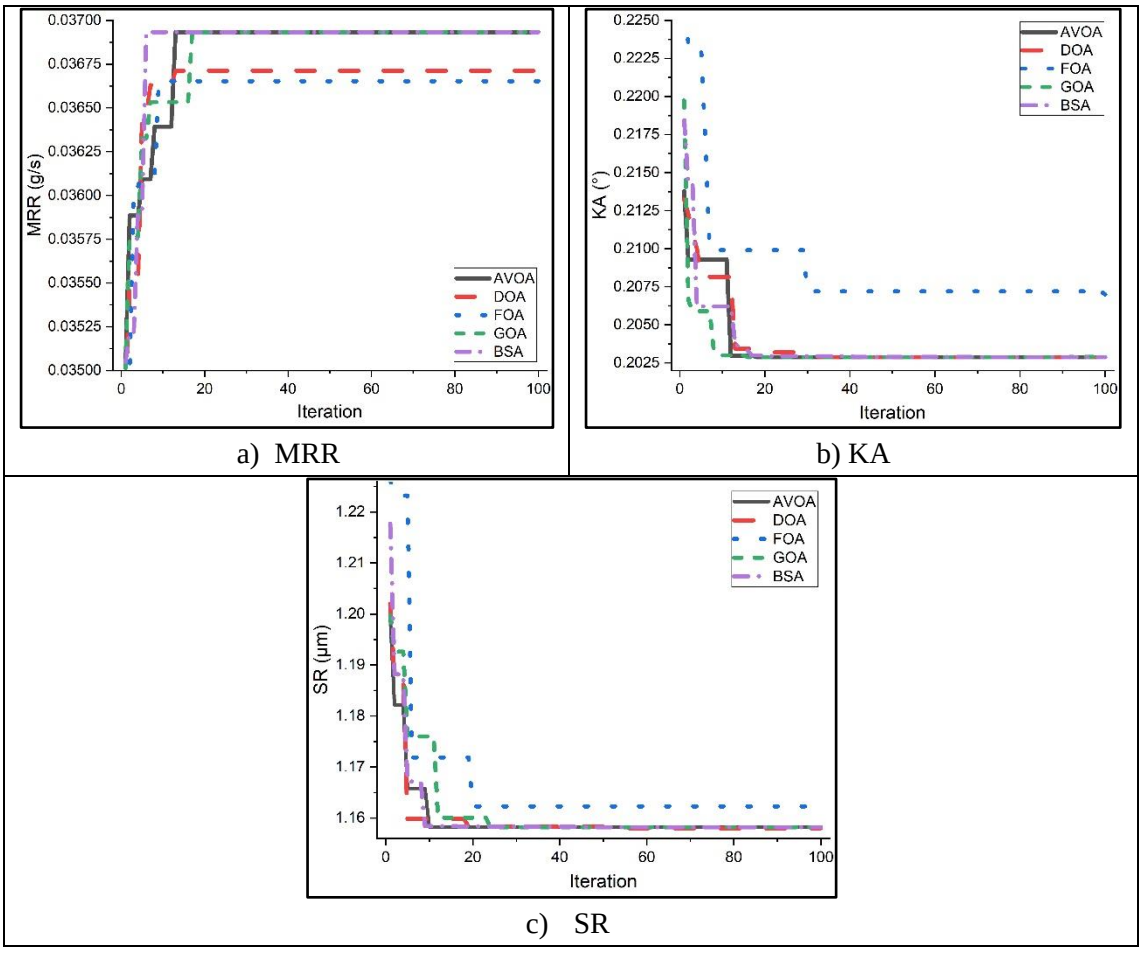


Figure 7.1 Convergence diagrams for the foraging behavior-based metaheuristic algorithms

Figures 7.2 exhibits the box plots of the three responses for AVOA, DOA, FOA, GOA and BSA techniques. Box plots are powerful tools to show repeatability of the solutions derived by any of the metaheuristic algorithms. For each optimizer, there are two boxes, one showing the corresponding minimum, maximum and quartiles of the solutions, while the other correlates with data distribution. In the box plots, if the data is skewed, the corresponding median would not be closer to the mean. In Figure 7.2, exact distribution of the solutions for each optimizer can be clearly noticed along with the skewness. For MRR, higher variations are observed in AVOA and BSA, while the solutions obtained using FOA are more consistent. As compared to other algorithms, AVOA and BSA require fewer function evaluations to derive the optimal solutions, validating their fast convergence speed [282]. Figures 7.2(b) and 7.2(c) have almost similar distributions of the solutions for KA and SR. For both these responses, the optimization performance of FOA with respect to variability of solutions is worse than GOA.

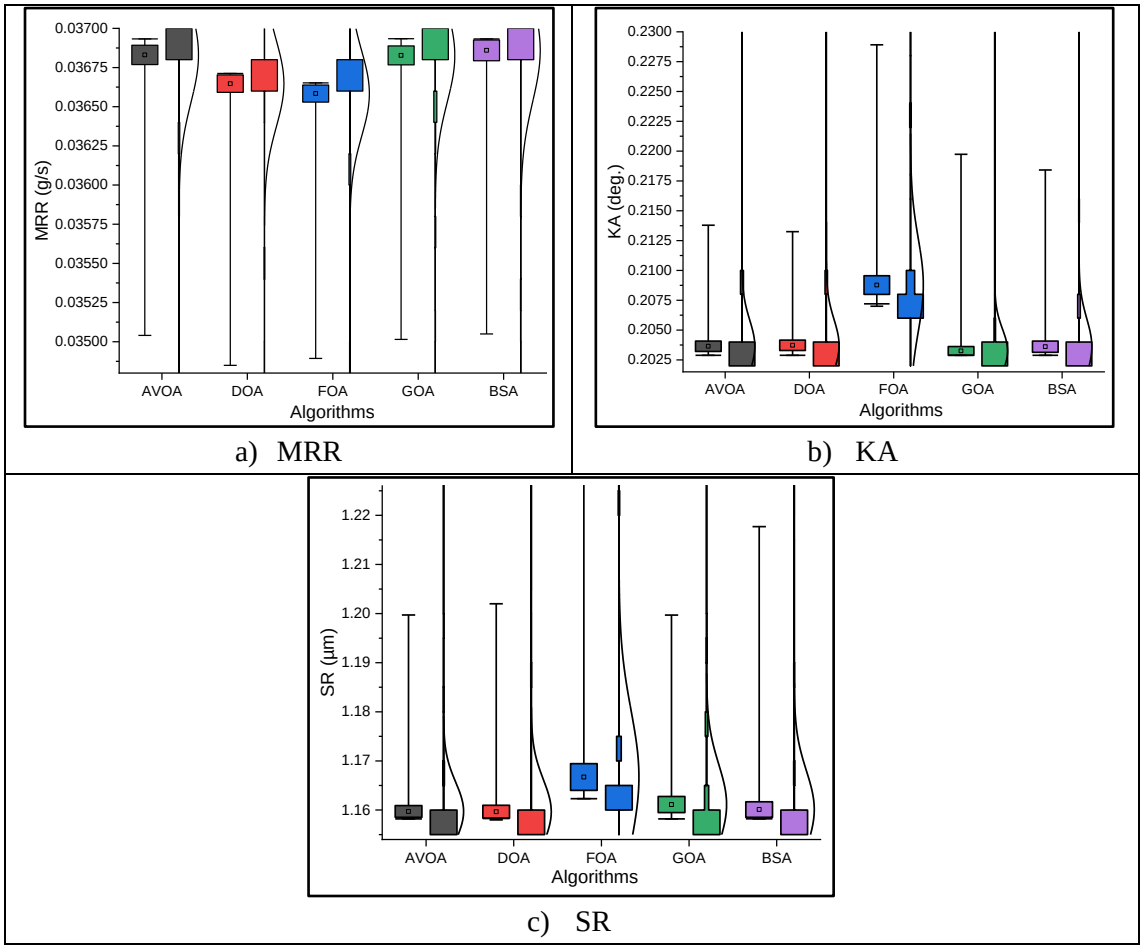


Figure 7.2 Box plots for the foraging behavior-based metaheuristic algorithms

Figure 7.3 shows the average computation time taken by each of the foraging behavior-based metaheuristic algorithms for each of the responses. It reveals that AVOA, FOA and BSA are the most computationally efficient optimizers, followed by GOA and DOA. When the performance of these algorithms is finally compared with respect to optimal values of the responses and computation time, the following ranking results can be derived:

MRR: AVOA = GOA = BSA > DOA > FOA

KA: AVOA = GOA = BSA > FOA > DOA

SR: AVOA > DOA > BSA > GOA > FOA

Computational time: AVOA > FOA > BSA > GOA > DOA

Thus, based on the single-objective optimization results, AVOA outperforms the other algorithms in regard of accuracy and variability of the derived solutions, and computation effort. As compared to the response values (MRR = 0.02152 g/s, KA = 0.359° and SR = 1.28 µm) reported by Balamurugan et al. [281], AVOA is able to achieve their corresponding values as 0.0369 g/s, 0.202° and 1.152 µm respectively at the identified optimal intermix of the AWJM parameters. It is interesting to note that all the algorithms under consideration provide dissimilar settings of the AWJM parameters for the three different responses which are practically impossible to maintain during a single machining

operation. Therefore, it is recommended to extract a unique combination of the AWJM parameters for simultaneous optimization of all the responses through multi-objective optimization of the said process.

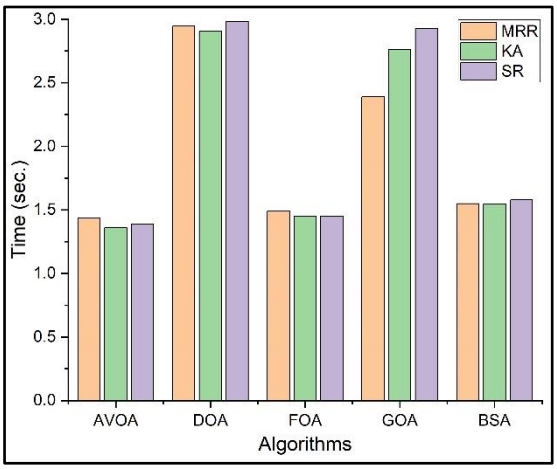


Figure 7.3 Average computation time for single objective optimization

7.2 Multi-objective optimization of the AWJM process

In an attempt to obtain a unique intermix of the AWJM parameters for simultaneous optimization of all the three responses, the WSMO model is applied, followed by development of the corresponding Pareto optimal front. The corresponding multi-objective function for WSMO is formulated as below:

$$\text{Min } Z = \left[-\frac{w_1 \times Y(MRR)}{MRR_{\max}} + \frac{w_2 \times Y(KA)}{KA_{\min}} + \frac{w_3 \times Y(SR)}{SR_{\min}} \right] \tag{7.4}$$

where $Y(MRR)$, $Y(KA)$ and $Y(SR)$ are the RSM-based models for MRR, KA and SR respectively; $MRR_{\max}$ is the maximum value of MRR, and $KA_{\min}$ and $SR_{\min}$ are the minimum KA and SR values (based on the single-objective optimization results); and w_1, w_2 and w_3 are the weights allocated to MRR, KA and SR, respectively. In this example, equal importance is assigned to all the responses. The multi-objective optimization results are presented in Table 7.3, which demonstrates that all the five foraging behavior-based metaheuristics are capable of providing better response values against those reported by Balamurugan et al. [281]. AVOA and FOA offer the best values of MRR in multi-objective optimization as 0.0362 g/s and 0.0354 g/s, respectively. It can be noticed from Table 7.3 that MRR is increased by 68.37, 57.21, 64.65, 62.79 and 62.79% for AVOA, DOA, FOA, GOA and BSA, respectively against that obtained by Balamurugan et al. [281]. Similarly, there are 43.45, 38.72, 42.34, 42.90 and 42.90% improvements in KA for AVOA, DOA, FOA, GOA and BSA, respectively. On the other hand, AVOA, DOA, FOA, GOA and BSA identify the corresponding SR values as 1.158, 1.163, 1.185, 1.16 and 1.159 µm, respectively which are much better than the recorded values of Balamurugan et al. [281]. To better comprehend the relationships between the responses, the corresponding 3D Pareto optimal front, consisting of a collection of non-dominated solutions, is developed for AVOA algorithm in Figure 7.4 to address the response weight dependency in WSMO approach affecting the optimal solutions. The optimal profile of the Pareto front for AVOA is smoother than the other algorithms, indicating its superior convergence properties. The Pareto optimal set of solutions would provide

flexibility to the process planners to increase performance of the said AWJM process, thereby offering a balance between MRR, KA and SR, depending on the end requirements.

Table 7.3 Multi-objective optimization results

Optimizer	Responses	AWJM parameters			Optimal	% improvement	Min. z
		A (bar)	*B* (mm)	*C* (mm/min)			
Balamurugan et al. [281]	MRR	220	1	40	0.0215	-	-
	KA	220	1.68	20	0.359		
	SR	220	1.32	20	1.28		
AVOA	MRR	220	2	20	0.0362	68.37	0.462
	KA				0.203	43.45	
	SR				1.158	9.53	
DOA	MRR	220	1.49	20.1	0.0338	57.21	0.48
	KA				0.22	38.72	
	SR				1.163	9.14	
FOA	MRR	220	1.421	20	0.0354	64.65	0.483
	KA				0.207	42.34	
	SR				1.185	7.42	
GOA	MRR	220	1.418	20	0.035	62.79	0.468
	KA				0.205	42.90	
	SR				1.16	9.38	
BSA	MRR	220	1.418	20	0.035	62.79	0.468
	KA				0.205	42.90	
	SR				1.159	9.45	

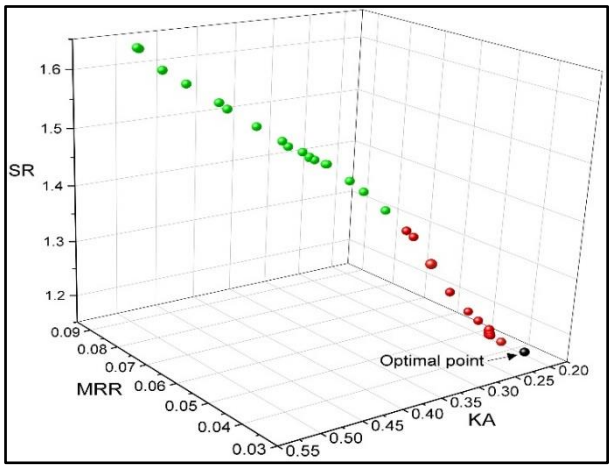


Figure 7.4 3D Pareto optimal front for AVOA

To investigate influences of the AWJM parameters on the responses and interactions between them, the corresponding parallel plot for AVOA algorithm is developed in Figure 7.5. However, it can be guaranteed that when AVOA algorithm is employed for multi-objective optimization of the AWJM operation on Lanthanum phosphate/Yttria composite, a specific combination of WJP = 220 bar, SOD = 2 mm and St = 20 mm/min would lead to achievement of the corresponding MRR, KA and SR values as 0.0362 g/s, 0.203° and 1.158 µm, respectively.

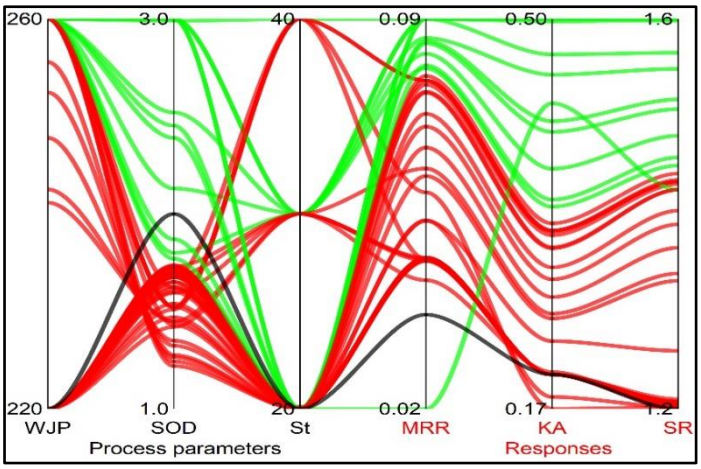


Figure 7.5 Parallel plot for AVOA

For this multi-objective optimization problem, convergence of *Z* value towards its minimum is portrayed in Figure 7.6 which again reiterates superiority of AVOA over the other algorithms. On the other hand, DOA and FOA require more number of iterations to arrive at their corresponding minimum *Z* values.

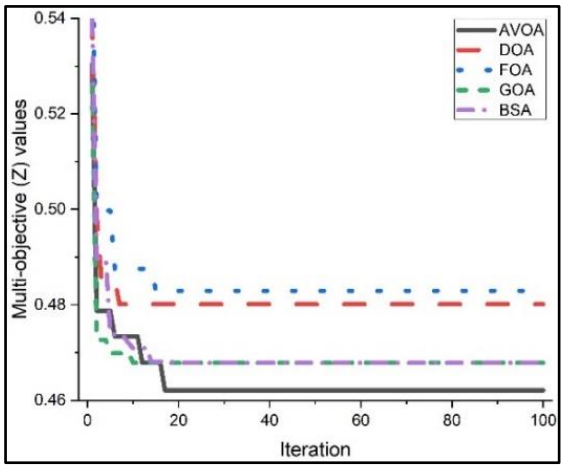


Figure 7.6 Convergence diagram for multi-objective optimization problem

Figure 7.7 shows the average computation time for the five foraging behavior-based metaheuristcs while solving the said multi-objective optimization problem. Based on the average computation time, these algorithms can be ranked as AVOA > FOA > BSA > DOA > GOA. Thus, AVOA excels over the others in successfully identifying the optimal combination of AWJM parameters, providing maximum MRR, and minimum KA and SR values with least computation effort.

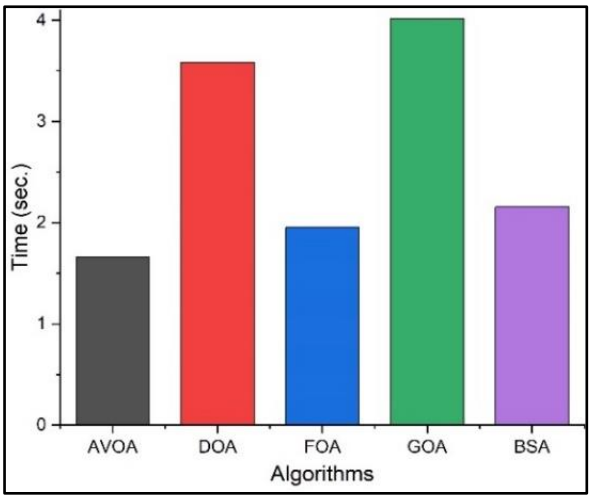


Figure 7.7 Average computational time for multi-objective optimization problem

7.3 Performance analysis

Like the earlier sections, the corresponding HV and SP values are computed to evaluate the optimization performance of the five foraging behavior-based algorithms. In Table 7.4, the best achieved values of HV and SP are denoted in bold letters. Thus, considering these two metrics, it can be unveiled that AVOA has an impressive performance, outperforming its competitors, and demonstrates excellent capability in solving high-dimensional optimization problems.

Table 7.4 HV and SP values

Algorithm	AVOA	DOA	FOA	GOA	BSA
HV	**0.0239**	0.0237	0.0231	0.0236	0.0234
SP	**0.0042**	0.0044	0.0048	0.0046	0.0043

The Friedman's mean rank test, as shown in Table 7.5, is carried out at 1% level of significance to evaluate the statistically significant difference between AVOA, DOA, FOA, GOA and BSA optimizers. Based on the results of Table 7.5, it can be inferred that for all the responses under consideration, AVOA is the best performing optimizer and the performance of FOA is the worst.

Table 7.5 Results of Friedman's mean rank test

Optimizer	Response	AVOA	DOA	FOA	GOA	BSA
Average	MRR	0.03683	0.036647	0.0365841	0.036827	0.036801
Rank sum		213	377	464	220	226
Avg. rank		2.13	3.77	4.64	2.2	2.26
Rank		**1**	**4**	**5**	**2**	**3**
Average	KA	0.203644	0.203726	0.2087755	0.203256	0.20361
Rank sum		179.5	213	500	248.5	359
Avg. rank		1.795	2.13	5	2.485	3.59
Rank		**1**	**2**	**5**	**3**	**4**
Average	SR	1.159714	1.159664	1.1667184	1.16112	1.160107
Rank sum		165.5	203	494	286.5	351
Avg. rank		1.655	2.03	4.94	2.865	3.51
Rank		**1**	**2**	**5**	**3**	**4**

To further validate the significant difference between AVOA and other algorithms, the Wilcoxon rank sum test is also conducted at 1% significance level. The results, presented in Table 7.6, reveal a notable distinction between AVOA and its competitors. Table 7.6 provides the sum of negative ranks (R−), positive ranks (R+) and *p*-values for pair-wise comparisons with respect to AVOA algorithm, reinforcing its superior performance.

Table 7.6 Wilcoxon rank sum test results

AVOA vs		DOA	FOA	GOA	BSA
R +	MRR	4299	4999	4314	56
R –		751	51	736	22
p-value		1.20E-04	1.70E-04	1.30E-04	0.18352
R +	KA	2806	5005	3770	4310
R –		2244	45	1180	740
p-value		3.32E-01	2.30E-04	3.80E-04	3.40E-04
R +	SR	3298	4473	1852	4951
R –		1752	328	39	99
p-value		7.82E-03	1.00E-04	2.70E-04	1.90E-04

7.4 Results and discussion

This sub-section investigates application of the five foraging behavior-based algorithm for parametric optimization of AWJM operation on Lanthanum phosphate/Yttria ($LaPO_4/Y_2O_3$) composite material. They achieve improvements ranging from 70.07 to 71.47% for MRR, 42.28 to 43.73% for KA, and 9.22 to 10% for SR in single-objective optimization; and 57.21 to 68.37%, 38.72 to 43.45%, and 7.42 to 9.53% improvements for MRR, KA and SR, respectively in multi-objective optimization, compared to the results reported by Balamurugan et al. [281]. Among these algorithms, AVOA consistently delivers the best results, proving to be most efficient in both single- and multi-objective optimization of the said AWJM process. In single-objective optimization, it attains the optimal values of 0.0369 g/s for MRR, 0.202° for KA and 1.152 μm for SR. Furthermore, in multi-objective optimization, the corresponding optimal values are 0.0362 g/s, 0.203° and 1.158 μm for MRR, KA and SR, respectively, at the best possible parametric combination of WJP = 220 bar, SOD = 2 mm and St = 20 mm/min. AVOA also excels over the others with reference to speed of convergence and computing time. For multi-objective optimization, it takes just 1.657 sec to find out the near global solution, reducing the runtime by 116.28, 17.85, 142.28 and 30.024% against DOA, FOA, GOA and BSA techniques, respectively. Its superiority is further confirmed using two quality matrics (SP and HV), and validated through Friedman's mean rank and Wilcoxon rank sum tests.

Finally, the said AWJM process is optimized (single-objective and multi-objective) using four other popular algorithms, i.e. GA, PSO, ABC and ACO, and the corresponding improvements (in percentage) in the response values against those obtained by the past researchers [281] are reported in Table 7.7. It can be interestingly noticed that none of them can provide better solutions than the five foraging behavior-based metaheuristic algorithms.

Table 7.7 Percentage improvements in response values based on GA, PSO, ABC and ACO algorithms

Optimizer	Problem	MRR	SR	KA	Av. time (sec)
GA	Single objective optimization	23.61	9.53	43.45	2.87
PSO		71.47	9.53	43.45	2.03
ABC		62.17	9.53	43.45	1.79
ACO		62.17	9.53	43.45	1.62
GA	Multi-objective optimization	64.65	9.14	42.34	2.86
PSO		64.65	9.14	42.34	2.1
ABC		60.47	9.38	33.15	1.73
ACO		62.71	9.23	37.15	1.68

8. OPTIMIZATION OF A PLASMA ARC CUTTING PROCESS EMPLOYING PHYSICS-BASED METAHEURISTIC ALGORITHMS

This section focuses on parametric optimization of a PAC process using five physics-based metaheuristic algorithms, i.e. AOA, ASO, NRO, EFO and GSA, considering a past experimental dataset. Subsequently, comparative studies are also conducted among their performance with respect to accuracy, convergence speed, computation time and variability of the derived solutions. Although all of them are quite competent in solving many of the popular benchmark functions with increasing complexity, there is no current research study on their applications in machining process optimization and comparison against other algorithms developed on the same principle. To strengthen the comparison, two statistical non-parametric tests, i.e. Friedman's mean rank test and Wilcoxon rank sum test are applied. The Pareto optimal front is subsequently developed to obtain unique parametric settings and optimal multi-objective solution for the PAC process. Finding out the optimal combination of the PAC parameters and achievable response values is made easier with the aid of a parallel plot. Additionally, the related values of SP and HV metrics are analyzed in order to identify the best performing physics-based algorithm. The procedural steps involved for parametric optimization of the considered PAC process are portrayed in Figure 8.1. This block diagram demonstrates a comprehensive approach for single- and multi-objective optimization of a PAC process using five physics-based algorithms, and comparison of their consistency, convergence trait and computation time based on different statistical and quality metrics, leading to selection of the best optimization strategy.

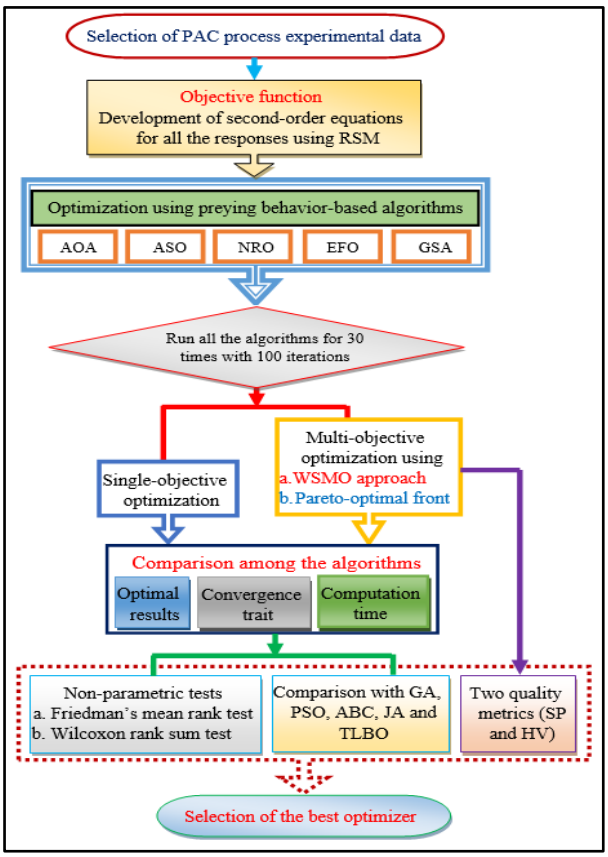


Figure 8.1 Block diagram for optimization of a PAC process

To execute the all considered algorithms, the relevant Python routines are developed and run on Spyder (Anacoda-3) in Windows 10 Pro with an Intel® Core™ i3-6006U CPU operating at 2.00 GHz and 4.00 GB of RAM. Each algorithm is individually run 30 times, with a maximum of 100 iterations. This means that 30 × 100 = 3000 trials would be used to carry out the analyses, and the population size is kept at 50.

Based on a BBD plan, Devaraj et al. [38] conducted 30 experiments on Monel 400 utilizing a CAM CNC-based PAC setup (Pro Arc Welding and Cutting System, India). During those experiments, CS (x_1) (mm/min), Pg (x_2) (bar), AC (x_3) (A) and SOD (x_4) (mm) were the input parameters, and MRR (g/min), KT (°) and HAZ (mm) were the responses. To study the effects of those PAC parameters on the responses, their values were varied at three different levels having equal intervals, i.e. CS {2200–2600 mm/min}, Pg {3–4 bar}, AC {45–55 A} and SOD {2–3 mm}. After each experiment, an Infra IN210 electronic weight scale (accuracy 0.0001 g) was employed to measure MRR. Similarly, an automated profile projector was used to measure KT, and a tool maker microscope BX53 with a magnification of 40X was utilized to record HAZ values. It is important to note that KT and HAZ are the small-the-better responses, while MRR is a larger-the-better response. It has been experimentally proved that maximum value of MRR can be obtained at lower CS and SOD values. On the other hand, higher settings of all the PAC parameters would lead to wider KT. A lower KT can only be attained at lower values of CS and Pg. It has also been noticed that HAZ would decrease with increase in CS and AC [38]. Using RSM technique and experimental data, the following second-order equations were developed for the responses under consideration using Design Expert software (Ver. 13), which would be subsequently treated as the objective functions for solving both the single- and multi-objective optimization problems for the said PAC process employing five physics-based metaheuristic algorithms.

$$Y(MRR) = 894.842 + 0.0623\times x_1 + 3.757\times x_2 - 29.547\times x_3 - 137.812\times x_4 - 0.01205\times x_1\times x_2 + 0.005028\times x_1\times x_3 - 4.327\times x_2\times x_4 + 2.041\times x_3\times x_4 - 0.00005779\times x_1^2 + 4.092\times x_2^2 + 0.1218\times x_3^2 + 8.604\times x_4^2 \quad (8.1)$$

$$Y(KT) = -542.946 + 0.107\times x_1 + 45.884\times x_2 + 12.156\times x_3 + 25.781\times x_4 - 0.0151\times x_1\times x_2 - 0.001018\times x_1\times x_3 + 0.204\times x_2\times x_3 + 3.968\times x_2\times x_4 - 0.776\times x_3\times x_4 - 4.335\times x_2^2 - 0.0843\times x_3^2 \quad (8.2)$$

$$Y(HAZ) = 82.091 + 0.0341\times x_1 + 5.167\times x_2 - 4.350\times x_3 - 10.578\times x_4 - 0.00198\times x_1\times x_2 + 0.0006337\times x_1\times x_3 - 0.002287\times x_1\times x_4 - 0.795\times x_2\times x_4 + 0.274\times x_3\times x_4 - 0.00001128\times x_1^2 + 0.022\times x_3^2 + 0.819\times x_4^2 \quad (8.3)$$

8.1 Single-objective optimization of the PAC process

Now, considering the above-mentioned RSM-based equations as the objective functions, and minimum and maximum values of the PAC parameters as the lower and upper bounds of the design variables, all the three responses, i.e. MRR, KT and HAZ are individually optimized using the five considered physics-based algorithms, i.e. AOA, ASO, NRO, EFO and GSA for single-objective optimization. The corresponding constraints are considered as the operating levels of the four PAC parameters, i.e. 2200 mm/min $\le x_1 \le$ 2600 m/min, 3 bar $\le x_2 \le$ 4 bar, 45 A $\le x_3 \le$ 55 A and 2 mm $\le x_4 \le$

3 mm. Table 8.1 shows the single-objective optimization results which also compare the relative performance of the adopted algorithms. To take into account the stochastic nature of these algorithms, each of them is independently executed 30 times with the maximum number of iterations and search agents. It can be noticed from Table 8.1 that all the metaheuristics result in substantial improvements in the considered response values, e.g. MRR (31.19-42.50%), KT (29.92-35.85%) and HAZ (18.49-25.55%), as compared to those recorded by Devaraj et al. [38]. Interestingly, NRO, EFO and GSA derive the optimal values of MRR as 44.46 g/min, which is better than that obtained using AOA and ASO. Thus, the maximum MRR can be obtained at the optimal intermix of different PAC parameters as CS = 2214 mm/min, Pg = 3 bar, AC = 45 A and SOD = 2 mm. In case of KT, there is superior performance of AOA, EFO and GSA against its competitors, achieving minimum KT value as 4.76°. Thus, to achieve minimum KT, it is recommended to set the corresponding PAC parameters as CS = 2200 mm/min, Pg = 4 bar, AC = 45 A and SOD = 2 mm, providing 35.85% improvement against that obtained by Devaraj et al. [38]. For HAZ, AOA, NRO, EFO and GSA can identify its minimum value of 3.06 mm with 25.55% improvement against the observation of Devaraj et al. [38]. Based on single-objective optimization results, it can be unveiled that EFO supersedes all its competitors, i.e. AOA, ASO, NRO and GSA, while achieving maximum MRR, and minimum of KT and HAZ values, along with the least SD of the derived solutions.

Table 8.1 Results of single-objective optimization

Optimizer	Response	Mean	SD	Optimal	PAC parameters				% improvement
					x_1 (mm/min)	x_2 (bar)	x_3 (A)	x_4 (mm)	
Devaraj et al. [38]	MRR (g/min)			31.2	2400	3.5	50	2.5	-
	KT (°)	-	-	7.42	2400	3.5	50	2.5	
	HAZ (mm)			4.11	2400	3.5	50	2.5	
AOA	MRR (g/min)	39.74	2.93	40.93	2263	3	45	2	31.19
	KT (°)	5.02	0.63	4.76	2200	4	45	2	35.85
	HAZ (mm)	3.25	0.48	3.06	2600	4	45	3	25.55
ASO	MRR (g/min)	41.38	2.96	42.64	2228	3	45	2	36.67
	KT (°)	5.29	0.53	5.07	2209	4	45	2	31.67
	HAZ (mm)	3.55	0.52	3.35	2576	3.98	46.23	2.94	18.49
NRO	MRR (g/min)	43.1	3.16	44.46	2214	3	45	2	42.5
	KT (°)	5.40	0.55	5.20	2228	4	45	2	29.92
	HAZ (mm)	3.23	0.45	3.06	2600	4	45	3	25.55
EFO	MRR (g/min)	43.34	2.73	44.46	2214	3	45	2	42.5
	KT (°)	4.98	0.52	4.76	2200	4	45	2	35.85
	HAZ (mm)	3.20	0.36	3.06	2600	4	45	3	25.55
GSA	MRR (g/min)	43.25	3.02	44.46	2214	3	45	2	42.5
	KT (°)	5.00	0.59	4.76	2200	4	45	2	35.85
	HAZ (mm)	3.23	0.43	3.06	2600	4	45	3	25.55

The corresponding convergence diagrams are developed in Figure 8.2 for the three responses under consideration. They prove superiority of EFO, showing its rapid progress towards achieving the

optimal response values with minimum iterations. With reference to convergence speed, the performance of ASO is the worst. The convergence patterns of NRO, EFO and GSA are noticed to be almost similar for MRR. For KT, the convergence patterns of AOA, EFO and GSA are almost the same. On the other hand, while minimizing HAZ, AOA, NRO, EFO and GSA have almost similar trends of convergence.

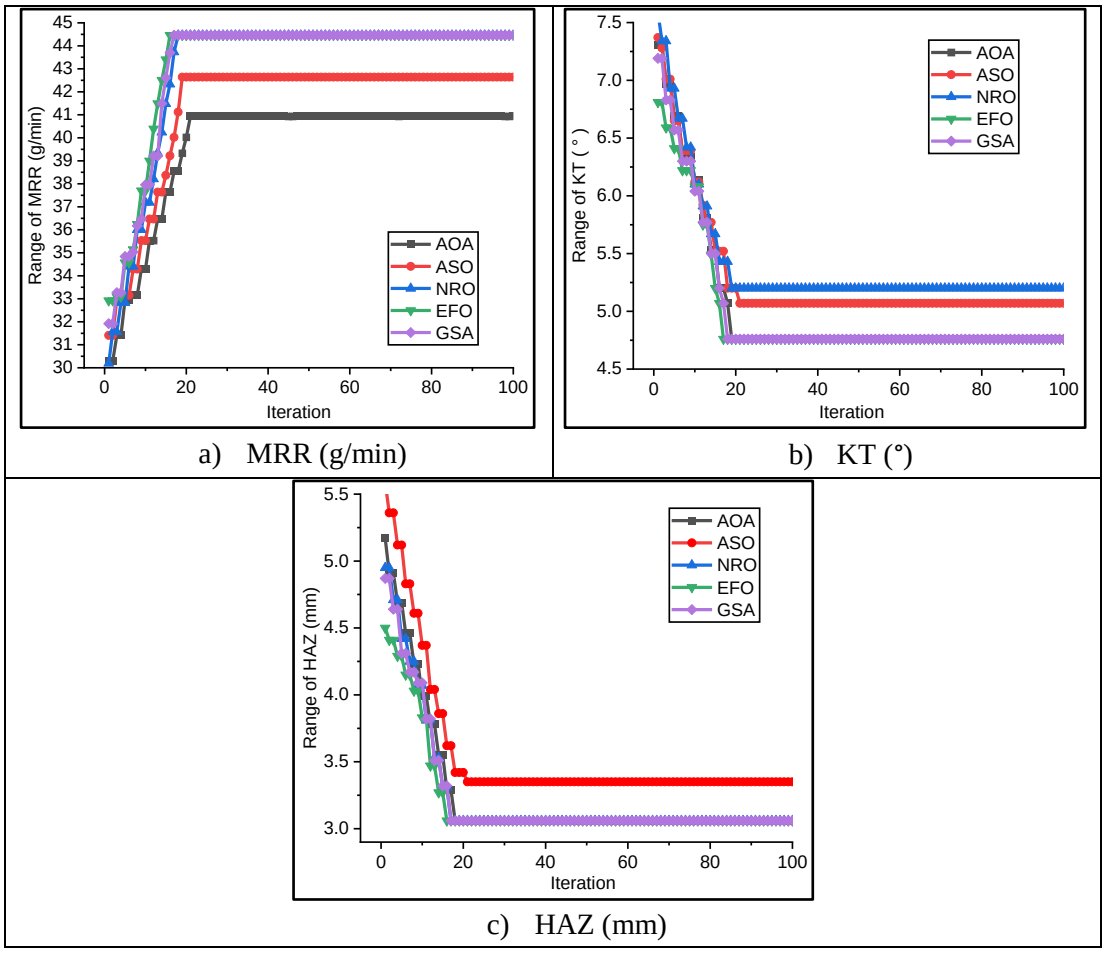


Figure 8.2 Convergence diagrams for the physics-based metaheuristic algorithms

Figure 8.3 depicts the box plots of the three responses against AOA, ASO, NRO, EFO and GSA techniques, also validating supremacy of EFO over its competitors in regard of variability of the derived solutions. In a box plot, the corresponding minimum, maximum and quartiles of the solutions, along with distribution of the data are portrayed. If the associated median is not closer to the mean, the data is skewed. In Figure 8.3, as compared to other algorithms, EFO has the minimum outlier spread and requires fewer function evaluations to derive the optimal solutions, validating its fast convergence speed, while AOA, ASO, NRO and GSA have maximum outlier spread for all the responses under consideration. Data density estimation is considered to display the distribution shape of the data for all the responses; the thinner section of a box plot indicates lower probability, while a wider section symbolizes a higher probability that the algorithm would provide the given value as optimal, which is highly concentrated around the mean value as observed in EFO algorithm. Thus, compared to other algorithms, the results produced by EFO are more consistent.

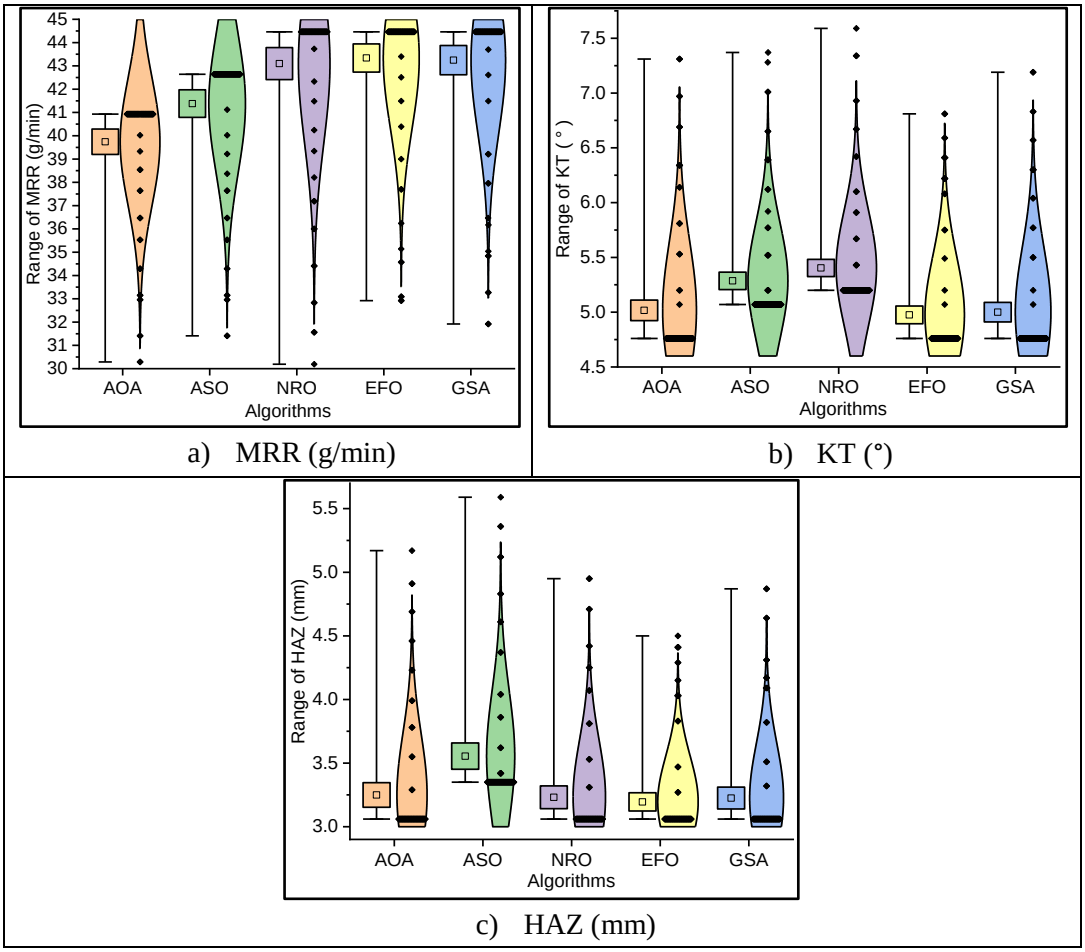


Figure 8.3 Box plots for the physics-based metaheuristic algorithms

It is noticed from Table 8.1 that single-objective optimization provides completely contradictory settings of PAC parameters in attaining maximum MRR, and minimum KT and HAZ values, which would be quite impractical to maintain in a single PAC setup. Single-objective optimization only concentrates on an individual quality characteristic (response) at a time, but multi-objective optimization can identify a unique combination of PAC parameters leading to concurrent optimization of all the responses, fulfilling productivity and surface quality requirements of a PAC process.

Figure 8.4 exhibits the average computational time for each of the physics-based metaheuristics across each of the responses while solving the corresponding single-objective optimization problems. It reveals that EFO is computationally efficient excelling over AOA, ASO, NRO and GSA. Thus, based on quality of the derived solutions, convergence effort and average computational time, it can be concluded that EFO performs best among the other physics-based algorithms.

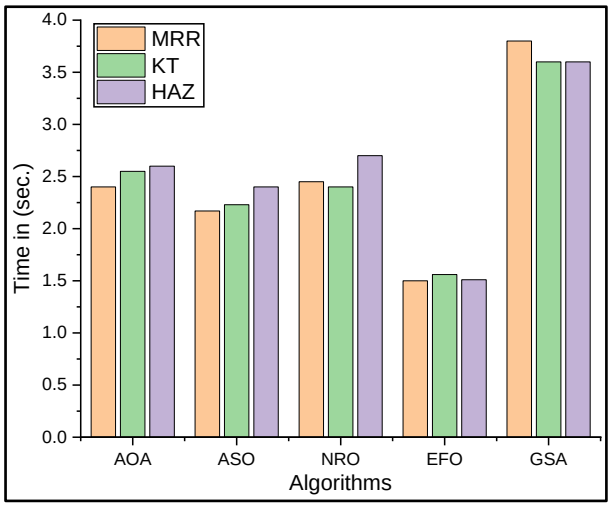


Figure 8.4 Average computation time for single-objective optimization

8.2 Multi-objective optimization of the PAC process

It should be highlighted that each algorithm in single-objective optimization finds out different PAC parameter settings for each response, making it difficult to manage in a real-time scenario. To simultaneously optimize all the three responses while providing a unique intermix of PAC settings, the WSMO model is adopted, followed by development of the corresponding Pareto optimal front. The multi-objective function for WSMO is formulated as shown follow:

$$Min\ (Z) = \left[-\frac{w_1 \times Y(MRR)}{MRR_{max}} + \frac{w_2 \times Y(KT)}{KT_{min}} + \frac{w_3 \times Y(HAZ)}{HAZ_{min}}\right] \tag{8.4}$$

where Y(MRR), Y(KT) and Y(HAZ) are the RSM-based equations for MRR, KT and HAZ respectively; MRR_{max}, KT_{min} and HAZ_{min} are the single-objective optimization-based results for maximum MRR, and minimum KT and HAZ respectively; and w_1, w_2 and w_3 are the weights allocated to MRR, KT and HAZ respectively. However, in this approach, the weights assigned to the responses have significant impact on the optimization results. It would be prudent to allocate equal importance to each of the responses if its relative significance is unknown in advance. But, it may not be the real situation. Depending on the end requirements, these responses may have dissimilar importance. The multi-objective optimization results of the considered PAC operation are presented in Table 8.2, demonstrating that all the five physics-based algorithms are capable of providing better response values against those reported by Devaraj et al. [38]. Hence, CS = 2554 mm/min, Pg = 4 bar, AC = 45.2 A and SOD = 2.44 mm would provide the optimal parametric combination for simultaneous maximization of MRR, and minimization of KT and HAZ for the said PAC process. It can be interestingly noted that EFO offers the best value of MRR in multi-objective optimization as 42.34 g/min. Table 8.2 also reveals that MRR is increased by 25.32, 28.81, 29.17, 35.71 and 34.26% for AOA, ASO, NRO, EFO and GSA, respectively. Similarly, for KT, the performance improvement ranges between 13.75% and 18.19%. On the other hand, AOA, ASO, NRO, EFO and GSA identify the corresponding HAZ values as 3.84, 3.9, 3.85, 3.74 and 3.74 mm respectively, which are much better than the recorded values of Devaraj et al. [38]. To better comprehend the relationships between the responses, the corresponding 3D Pareto optimal front,

consisting of a collection of non-dominated solutions, is developed for EFO algorithm in Figure 8.5 to address the response weight dependency in WSMO approach affecting the optimal solutions. The optimal profile of the Pareto front for EFO is smoother than the other algorithms, indicating its superior convergence properties. The Pareto optimal set of solutions would provide flexibility to the process planners to increase performance of the said PAC process, thereby offering a balance between MRR, KT and HAZ, depending on the end requirements.

Table 8.2 Results of multi-objective optimization

Optimizer	Response	Optimal	Min z	PAC parameters				% improvement
				x_1 (mm/min)	x_2 (bar)	x_3 (A)	x_4 (mm)	
Devaraj et al. [38]	MRR (g/min)	31.2	-	2400	3.5	50	2.5	-
	KT (°)	7.42						
	HAZ (mm)	4.11						
AOA	MRR (g/min)	39.1	0.78	2545	4	45	2.92	25.32
	KT (°)	6.17						16.85
	HAZ (mm)	3.84						6.57
ASO	MRR (g/min)	40.19	0.72	2598.6	3.95	47.95	2.79	28.81
	KT (°)	6.28						15.36
	HAZ (mm)	3.9						5.11
NRO	MRR (g/min)	40.3	0.69	2573	4	45.4	2.91	29.17
	KT (°)	6.4						13.75
	HAZ (mm)	3.85						6.33
EFO	MRR (g/min)	42.34	0.66	2554	4	45.2	2.44	35.71
	KT (°)	6.07						18.19
	HAZ (mm)	3.74						9.00
GSA	MRR (g/min)	41.89	0.667	2598	4	45.2	2.52	34.26
	KT (°)	6.1						17.79
	HAZ (mm)	3.74						9.00

To understand impacts of PAC parameters on the responses and interactions between them, the related parallel plot is designed in Figure 8.6 for EFO algorithm. It provides several parametric intermixes to increase MRR (42.34 g/min) with lower KT (6.07°) and HAZ (3.74 mm) with a specific combination of CS = 2554 mm/min, Pg = 4 bar, AC = 45.2 A and SOD = 2.44 mm, resulting in achieving the corresponding response values during multi-objective optimization based on EFO algorithm. The solutions represented by red and green colors validate superiority of EFO in the lower and upper portions of the Pareto front. The black point in the Pareto front and black line in the parallel plot show the PAC process's optimal performance with respect to maximum MRR, and minimum KT and HAZ. Figure 8.7 shows the average computation time for the five physics-based metaheuristics based on 30 independent runs for each algorithm while solving the said multi-objective optimization problem. Based on the average computation time, they can be ranked as EFO > ASO > NRO > AOA > GSA. Thus, EFO excels over the others in successfully identifying the optimal combination of PAC parameters, providing maximum MRR, and minimum KT and HAZ values with minimum computation effort.

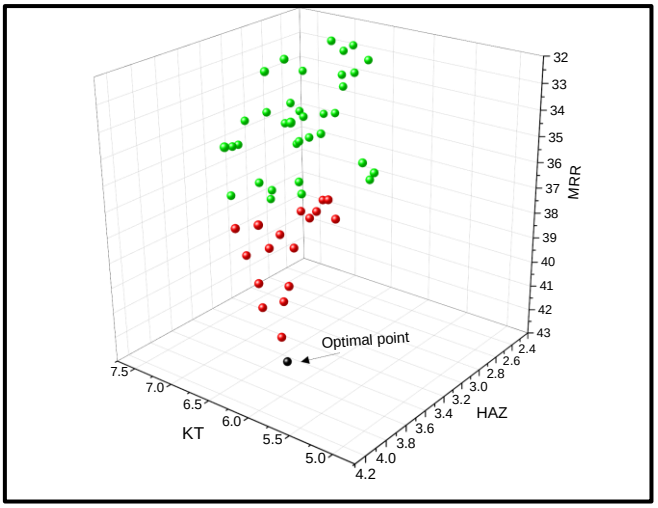


Figure 8.5 3D Pareto optimal front for EFO

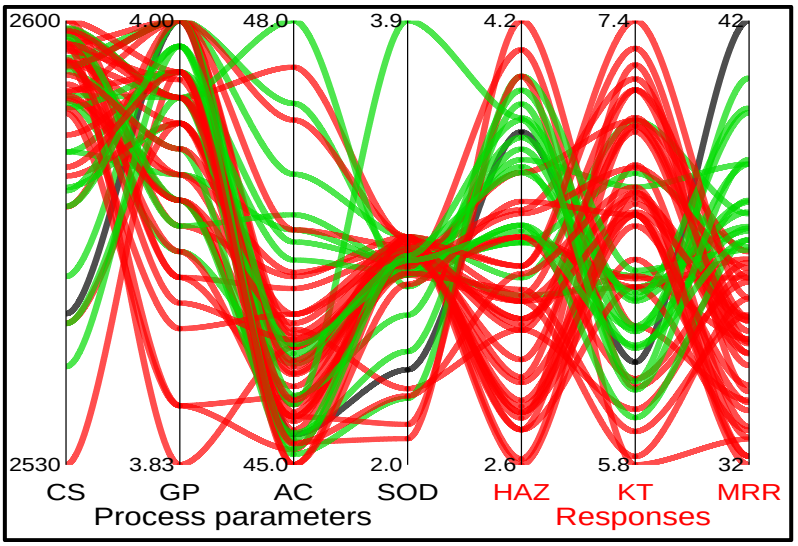


Figure 8.6 Parallel plot for EFO

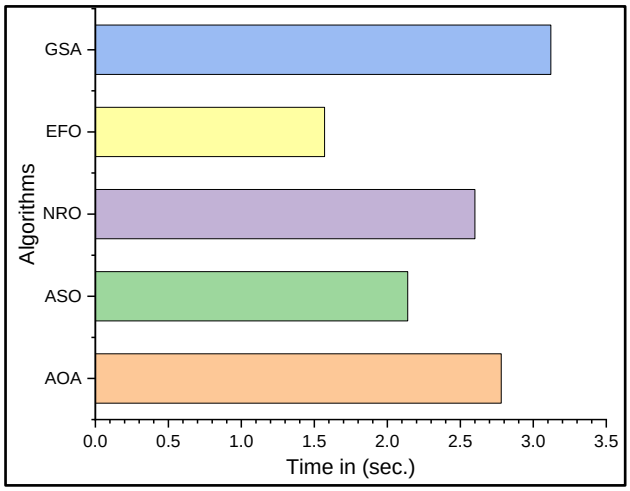


Figure 8.7 Average computational time for multi-objective optimization

8.3 Performance analysis

Like the earlier sections, the corresponding HV and SP values are also computed to evaluate the optimization performance of the five physics-based algorithms. In Table 8.3, the best achieved values

of HV and SP are denoted in bold letters. Thus, considering these two metrics, it can be unveiled that EFO has an impressive performance, outperforming its competitors, and demonstrates excellent capability in solving high-dimensional optimization problems.

Table 8.3 HV and SP values

Algorithm	AOA	ASO	NRO	EFO	GSA
HV	1.772	1.767	1.775	**1.778**	1.77
SP	0.44	0.45	0.43	**0.41**	0.42

Similarly, the results of Friedman's mean rank test for all the responses are provided in Table 8.4, to evaluate the statistically significant difference between AOA, ASO, NRO, EFO and GSA optimizers. Based on the results of Table 8.4, it can be inferred that EFO is the best performing optimizer, while the performance of ASO is the worst while optimizing the said PAC process.

Table 8.4 Results of Friedman's mean rank test

Algorithm	Response	AOA	ASO	NRO	EFO	GSA
Average	MRR	39.7431	41.3811	43.0968	43.3383	43.2452
Rank sum		498	396	222	188.5	195.5
Average rank		4.98	3.96	2.22	1.885	1.955
Rank		**5**	**4**	**3**	**1**	**2**
Average	KT	5.0171	5.286	5.4036	4.9754	5.0005
Rank sum		228.5	400.5	488	184.5	198.5
Average rank		2.285	4.005	4.88	1.845	1.985
Rank		**3**	**4**	**5**	**1**	**2**
Average	HAZ	3.2495	3.5547	3.2314	3.195	3.225
Rank sum		266.5	500	256.5	227.5	249.5
Average rank		2.665	5	2.565	2.275	2.495
Rank		**4**	**5**	**3**	**1**	**2**

Finally, the results of Wilcoxon rank sum test, as provided in Table 8.5, prove the presence of significant difference between EFO over its competitors in solving the considered multi-objective optimization problem while optimizing the said PAC operation on Monel 400 work material.

Table 8.5 Wilcoxon rank sum test results

EFO vs		AOA	ASO	NRO	GSA
R +	MRR	4215	38	148	121
R –		125	7	5	15
p-value		1.00E-05	9.73E-05	1.50E-04	0.0031
R +	KT	162	3524	159	144
R –		9	84	8	9
p-value		1.90E-04	1.58E-05	4.86E-05	3.40E-04
R +	HAZ	145	4167	135	135
R –		8	103	1	1
p-value		1.40E-04	1.00E-05	2.70E-04	2.70E-04

8.4 Results and discussion

As previously stated, this work focuses on application of five potential physics-based algorithms for parametric optimization of a PAC process. For single-objective optimization of the PAC process, all the algorithms provide the corresponding improvements of 31.19-42.50% in MRR, 29.92-35.85% in KT and 18.49-25.55% in HAZ, as compared to those derived by Devaraj et al. [38]. For multi-objective optimization, these improvements are 25.32-35.71%, 13.75-18.19% and 5.11-9% for MRR, KT and HAZ, respectively.

For single- and multi-objective optimization problems, EFO always provides the best results, appearing as the most efficient algorithm. It achieves the optimal values of 44.46 g/min, 4.76° and 3.06 mm for single-objective optimization; and 42.34 g/min, 6.07° and 3.74 mm for multi-objective optimization, for MRR, KT and HAZ, respectively with a specific combination of CS = 2554 mm/min, Pg = 4 bar, AC = 45.2 A and SOD = 2.44 mm.

However, EFO supersedes the others with respect to faster convergence speed and computation time. In multi-objective optimization, it only takes 1.57 sec to obtain the global optimal solution; and saves 77.1%, 36.3%, 65.6% and 98.73% of the runtime against AOA, ASO, NRO and GSA techniques, respectively. Results of two quality metrics (SP and HV) and two non-parametric statistical tests (Friedman's mean rank test and Wilcoxon rank-sum test) also confirm superiority of EFO over the other algorithms under consideration.

To verify the significant difference between EFO and other considered algorithms, its performance is contrasted for single- and multi-objective optimization with respect to percentage improvements in the predicted response values and average computational time, as depicted in Table 8.6. It exhibits that EFO provides better or similar values for all the responses under consideration as compared to other metaheuristics. In case of single-objective optimization, its performance is similar to that of AOA for KT and HAZ; NRO for MRR and HAZ; and GSA for MRR, KT and HAZ. For multi-objective optimization, it performs same as GSA for HAZ. Thus, it can be concluded that the optimization performance of EFO is only comparable to that of GSA. With respect to average computational time, EFO significantly outperforms its competitors for the optimization problem. Thus, based on the observations, it can be concluded that increasing CS enhances MRR allowing more material to be removed in a shorter time. Faster tool movement also limits heat absorption by the material, reducing size of the HAZ and risk of thermal deformation. It also helps in maintaining a straighter, narrower kerf, thereby minimizing KT. Higher Pg effectively removes the molten material from the cut area, improving cut quality while preserving the plasma jet's shape and intensity. It cools the material more efficiently, further reducing HAZ. Lowering AC decreases heat generation, minimizes HAZ and improves surface quality by reducing unwanted thermal effects. While lower AC may reduce energy input, optimizing other parameters, like CS can help maintaining or even increasing MRR. A proper SOD ensures that the plasma arc remains focused, allowing efficient cutting without excessive heat dissipation, helping maintain a consistent arc and minimizing KT by preventing

widening of the cut.

Table 8.6 Percentage improvements of the responses and computational time for EFO against its competitors

EFO vs	Technique	MRR	KT	HAZ	Time
AOA	Single-objective	7.94	0	0	65.21
ASO		4.09	6.51	9.48	48.80
NRO		0	9.24	0	65.21
GSA		0	0	0	140.70
AOA	Multi-objective	7.65	1.65	2.67	77.07
ASO		5.08	3.46	4.28	36.31
NRO		4.82	5.44	2.94	65.61
GSA		1.06	0.49	0	98.73

Five popular state-of-the-art metaheuristic algorithms, i.e. GA, PSO, ABC, TLBO and JA are finally utilized to optimize the PAC process. Table 8.7 reports the improvements in the response values (in percentage) that EFO algorithm can achieve [38] as well as average computation time for those algorithms. It can be interestingly noticed from Table 8.7 that all the considered physics-based algorithms outperform them while addressing both the single- and multi-objective optimization problems.

Table 8.7 GA, PSO, ABC, TLBO and JA-based optimization results

Optimizer	Technique	MRR	KT	HAZ	Average time (sec)
GA	Single-objective	32.82	33.96	24.57	3.43
PSO		32.82	33.69	24.57	1.84
ABC		29.81	31.67	22.38	4.3
TLBO		29.81	34.37	23.60	3.83
JA		32.69	33.96	21.41	1.64
GA	Multi-objective	34.26	17.79	9.00	3.16
PSO		34.26	17.79	9.00	1.97
ABC		33.04	16.04	3.41	4.25
TLBO		34.94	16.58	7.30	3.92
JA		30.77	16.98	7.79	1.72

In this section, all the five physics-based algorithms demonstrate superior performance as compared to other methods, like RSM, fuzzy logic and DFA, adopted by the past researchers [38], making them highly suitable for solving complex optimization problems, particularly multi-objective and high-dimensional cases. Against the conventional optimization methods, they show higher efficiency, adaptability and scalability, and can be deployed to optimize any of the machining processes in continuous solution space.

9. OPTIMIZING AN END MILLING PROCESS USING THE BEST-PERFORMING METAHEURISTIC ALGORITHMS

In the previous sections, five machining processes, i.e. turning, green machining, LBD, AWJM and PAC are optimized, employing different nature-inspired metaheuristic algorithms to determine the optimal combinations of the process parameters. Five distinct categories of nature-inspired algorithms are utilized for optimization of several machining operations. In case of the turning process, five human-inspired metaheuristic algorithms are applied, with TLBO emerging as the most efficient technique for optimizing the said turning operation. For the green machining process, five mating behavior-based metaheuristic algorithms are used, where FMA demonstrate its superiority in solving the said parametric optimization problem. Among five preying behavior-based techniques applied for optimization of the LBD process, GWO outperforms the others. For the PAC process, five physics-based metaheuristic algorithms are employed, where EFO consistently yields the best results in both the single- and multi-objective optimization problems. Finally, for the AWJM process, five foraging behavior-based metaheuristics are adopted, with AVOA emerges as the best performing optimizer. To assess the algorithm performance, Table 9.1 presents the best performing algorithms from five different categories of the nature-inspired metaheuristics. A total of 25 metaheuristic algorithms are tested across five machining processes, from which five best performing algorithms, i.e. TLBO, FMA, GWO, EFO and AVOA, are selected for further analysis in terms of accuracy, convergence speed, computation time and variability of the derived solutions. These top-performing algorithms are considered to optimize an end milling operation using the past experimental dataset. Comparative studies are conducted to assess their performance across some key evaluation criteria. To strengthen the comparison, two statistical non-parametric tests, i.e. Friedman's mean rank test and Wilcoxon rank sum test are also applied. The Pareto optimal front is subsequently developed to obtain the unique parametric settings and optimal multi-objective solution for the end milling process. Finding out the optimal combination of the end milling parameters and achievable response values is made easier with the aid of the parallel plot. Additionally, the related values of SP and HV metrics are also analyzed in order to identify the best performing algorithm.

Table 9.1 Best-performing algorithms from five types of nature-inspired metaheuristics

Types of nature-inspired algorithm	Algorithms combined	Machining process	Best-performing algorithm
Human-inspired	TLBO, SAR, TOA, HCO, QSA	Turning	TLBO
Mating behavior-based	FMA, NMR, BMA, BMOA, AZOA	Green machining (dry milling)	FMA
Preying behavior-based	SFO, AO, HHO, BA, GWO	LBD	GWO
Physics-based	AOA, ASO, NRO, EFO, GSA	PAC	EFO
Foraging behavior-based	DOA, AVOA, GOA, FOA, BSA	AWJM	AVOA

Shihab et al. [283] conducted 30 milling experiments on hybrid Al7075/SiC/Gr composite based on a CCD plan, considering four input parameters, i.e. SS (x_1) (in rpm), FR (x_2) (in mm/rev), DOC (x_3) (in mm) and concentration of SiC/Gr (x_4) (in %), at three distinct levels as SS {500, 750, 1000 rpm}, FR {0.02, 0.06, 0.1 mm/rev}, DOC {0.5, 1, 1.5 mm} and concentration of SiC/Gr {9, 12, 15 %}. The three responses measured had included MRR (mm^3/min), SR (μm) and MH (HV). It is important to note that MRR and MH are considered as the 'larger-the-better' responses, while SR is a 'smaller-the-better' attribute. After every milling experiment, the MH values were monitored using a digital micro Vickers hardness tester, while SR values were measured using MAHR FEDERAL POCKET SURF 44100 tester. By integrating RSM with DFA, Shihab et al. [283] observed that the optimal parameters as SS = 1000 rpm, FR = 0.0677 mm/rev, DOC = 1.1869 mm and concentration of SiC/Gr = 12.697% would provide the maximum values of MRR = 2043.35 mm^3/min and MH = 142.368 HV, and minimum value of SR = 1.2942 μm. The following second order equations for the responses were formulated employing Design Expert software (Ver. 13.0), which would be later employed as the objective functions while working out the single- and multi-objective optimization problems related to the said milling process using five best-performing metaheuristic algorithms.

$$Y(MRR) = -3986 + 0.611\times x_1 + 1694\times x_2 + 306\times x_3 + 593\times x_4 + 7\times x_1\times x_2 + 11500\times x_2x_3 - 24.69\times x_4^2 \quad (9.1)$$

$$Y(SR) = -0.949 + 0.000608\times x_1 + 33.66\times x_2 - 1.638\times x_3 + 0.1485\times x_4 - 0.001035\times x_1x_3 - 1.339\times x_2\times x_4 + 1.274\times x_3^2 \quad (9.2)$$

$$Y(MH) = 118.62 - 0.1367\times x_1 + 57.78\times x_2 + 27.47\times x_3 + 2.701\times x_4 - 0.01115\times x_1x_3 - 0.002208\times x_1\times x_4 + 1.346\times x_3\times x_4 + 0.000136\times x_1^2 - 17.99\times x_3^2 \quad (9.3)$$

9.1 Single-objective optimization of the end milling process

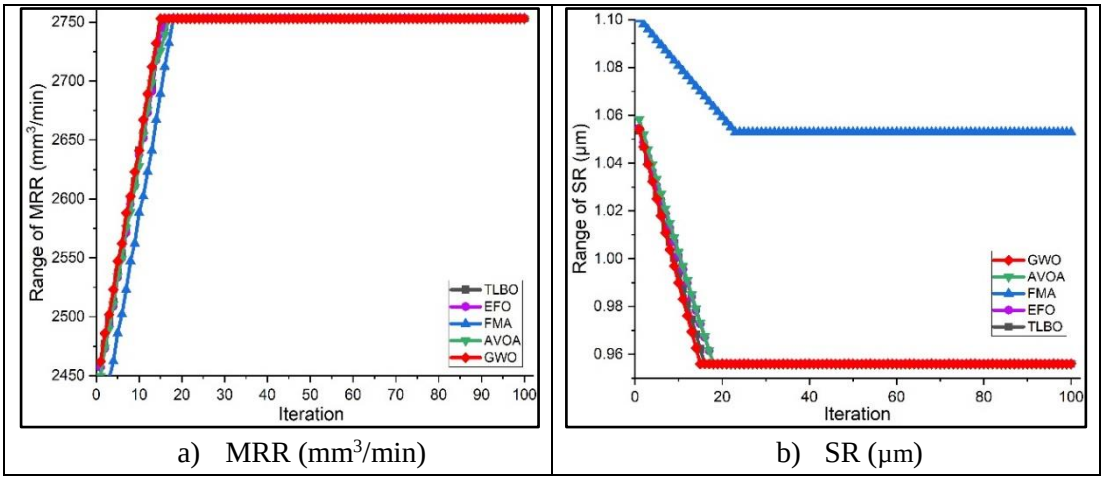
Now, using the RSM-based equations as the objective functions, and lower and higher limits of the considered milling parameters as the minimum and maximum values of the design variables, respectively, each of the three responses is individually optimized using the five considered algorithms. The corresponding constraints are thereby treated as the working levels of the four milling parameters, i.e. 500 rpm $\leq x_1 \leq$ 1000 rpm, 0.02 mm/rev $\leq x_2 \leq$ 0.1 mm/rev, 0.5 mm $\leq x_3 \leq$ 1.5 mm and 9% $\leq x_4 \leq$ 15%. Results of the single-objective optimization are exhibited in Table 9.2, which also contrasts relative performance of the adopted algorithms.

Table 9.2 indicates that these metaheuristics provide considerable enhancements in the response values, like MRR (34.74%), SR (18.64-26.13%) and MH (5.07%), as compared to those recorded by Shihab et al. [283]. All the considered algorithms derive the optimal values of MRR and MH as 2753.18 mm^3/min and 149.58 HV, respectively. The minimum SR is achieved at the optimal combination of the milling parameters, specifically as SS = 1000 rpm, FR = 0.02 mm/rev, DOC = 1.05 mm and concentration of SiC/Gr = 9%. GWO, AVOA, EFO and TLBO can identify its global maximum value as 0.956 μm. The results of single-objective optimization indicate that GWO, AVOA, EFO and TLBO can provide the optimal values of all the considered responses.

Table 9.2 Solutions of the single-objective optimization problem

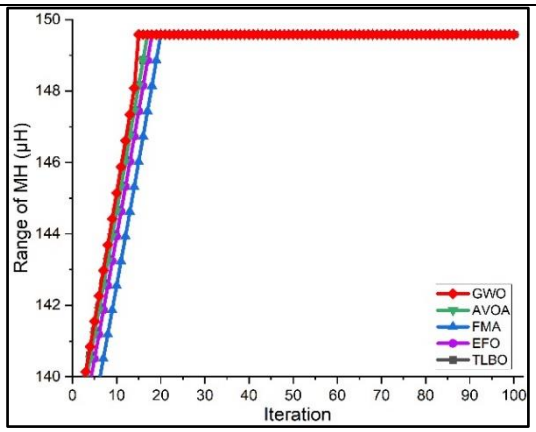
Optimizer	Response	Optimal	Process Parameters				SD	Mean	% improvement
			x_1	x_2	x_3	x_4			
Shihab et al. [283]	MRR (mm^3/min)	2043.35	1000	0.0677	1.1869	12.697	-	-	-
	SR (μm)	1.2942							
	MH (HV)	142.368							
GWO	MRR (mm^3/min)	2753.18	1000	0.1	1.5	12	2731.09	63.237	34.74
	SR (μm)	0.956	1000	0.02	1.05	9	0.021	0.963	26.13
	MH (HV)	149.58	1000	0.1	1.01	15	2.43	148.710	5.07
AVOA	MRR (mm^3/min)	2753.18	1000	0.1	1.5	12	2728.87	67.060	34.74
	SR (μm)	0.956	1000	0.02	1.05	9	0.024	0.965	26.13
	MH (HV)	149.58	1000	0.1	1.01	15	2.54	148.630	5.07
FMA	MRR (mm^3/min)	2753.18	1000	0.1	1.5	12	2721.39	81.605	34.74
	SR (μm)	1.053	985	0.021	1	9.05	0.012	1.058	18.64
	MH (HV)	149.58	1000	0.1	1.01	15	3.21	148.250	5.07
EFO	MRR (mm^3/min)	2753.18	1000	0.1	1.5	12	2729.18	66.571	34.74
	SR (μm)	0.956	1000	0.02	1.05	9	0.023	0.965	26.13
	MH (HV)	149.58	1000	0.1	1.01	15	2.76	148.510	5.07
TLBO	MRR (mm^3/min)	2753.18	1000	0.1	1.5	12	2729.86	65.393	34.74
	SR (μm)	0.956	1000	0.02	1.05	9	0.021	0.964	26.13
	MH (HV)	149.58	1000	0.1	1.01	15	2.52	148.640	5.07

Figure 9.1 plots the convergence diagrams of each of the algorithms for the considered responses. All the algorithms (except FMA) show rapid progress towards the corresponding optimal solutions after only a few iterations. The convergence behaviors of GWO, AVOA, EFO and TLBO are almost the same for all the responses. These convergence diagrams primarily prove the superiority of GWO over the other algorithms in achieving the optimal solutions with least number of iterations. With respect to speed of convergence, FMA performs worst, showing poor advancement towards the optimal solutions.



a) MRR (mm^3/min) b) SR (μm)

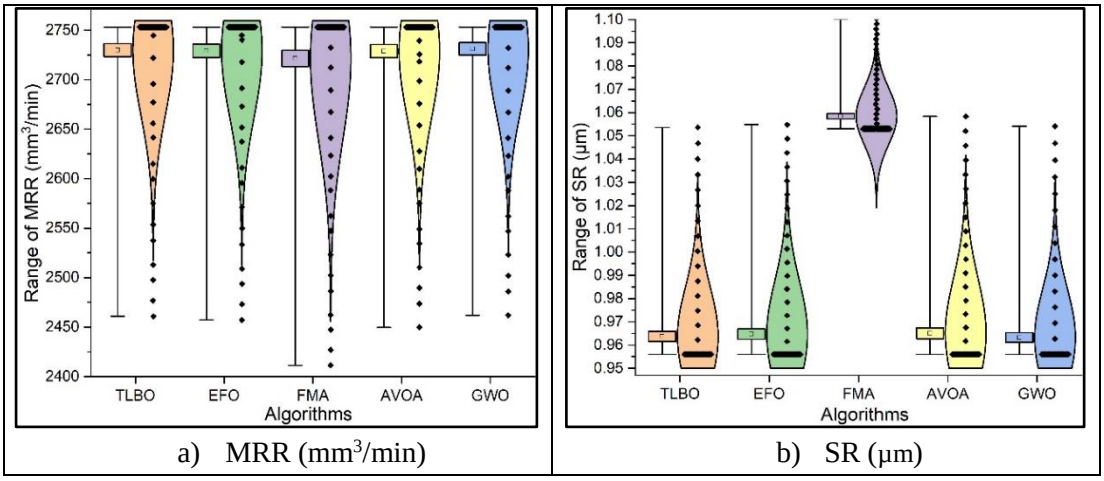
Figure 9.1 Convergence diagrams



c) MH (HV)

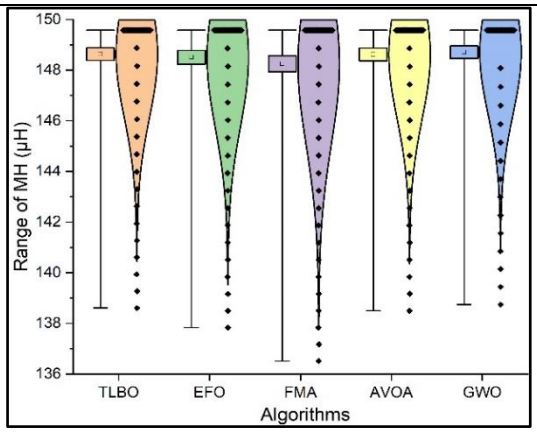
Figure 9.1 Contd.

Figure 9.2 exhibits the box plots combined with data density diagrams illustrating the distribution and variability of the responses across the five algorithms while validating supremacy of GWO over its competitors with reference to the achieved solutions. Compared to other algorithms, GWO exhibits compact distributions with small ranges and few outliers, as indicated by its narrow interquartile ranges, lower variability and dense concentration of data around the median, reflecting high consistency in the derived solutions. Similarly, EFO, AVOA and TLBO also display narrower interquartile ranges with fewer outliers; however, their data density estimations show slightly greater variation in the data spread compared to GWO. Overall, GWO, EFO, AVOA and TLBO demonstrate consistent performance with minimal variability, whereas, FMA exhibits maximum variability with a large number of outliers.



a) MRR (mm^3/min) b) SR (μm)

Figure 9.2 Developed box plots



c) MH (HV)

Figure 9.2 Contd.

Figure 9.3 illustrates the mean computation time for the three responses across the five metaheuristic algorithms, in the context of single-objective optimization. It reveals that GWO, EFO, AVOA and TLBO are computationally efficient optimizers. Considering quality of the derived solutions, convergence effort and average computational time, it can be concluded that GWO emerges out as the best performing algorithm among those evaluated.

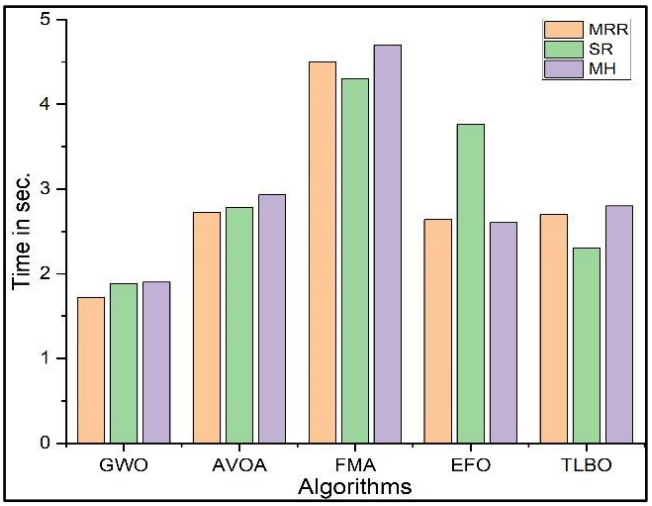


Figure 9.3 Mean computing time for single-objective optimization

9.2 Multi-objective optimization of the end milling process

The WSMO model is first applied, followed by development of the corresponding Pareto optimal front. The corresponding multi-objective function for WSMO is formulated as below:

$$Min\ (Z) = \left[-\frac{w_1 \times Y(MRR)}{MRR_{max}} + \frac{w_2 \times Y(SR)}{SR_{min}} - \frac{w_3 \times Y(MH)}{MH_{max}}\right] \quad (9.4)$$

where Y(*MRR*), Y(*SR*) and Y(*MH*) represent the RSM-based equations for MRR, SR and MH, respectively; MRR_{max}, SR_{min} and MH_{max} are the results based on single-objective optimization for MRR, SR and MH, respectively; and w_1, w_2 and w_3 are the weights allocated to MRR, SR and MH, respectively. Table 9.3 represents the multi-objective optimization results for the said end milling operation, revealing that all the five algorithms achieve better results compared to those recorded by

Shihab et al. [283]. Notably, GWO, AVOA, EFO and TLBO surpass FMA in maximizing MRR and MH while minimizing SR. To achieve the optimal performance across all the responses, the recommended milling parameters are SS = 1000 rpm, FR = 0.06 mm/rev, DOC = 1.4 mm and concentration of SiC/Gr = 12.4%. A comparative analysis shows that almost all the algorithms result in similar improvement of 12.17% in MRR (4.09% for FMA), and 10.37% in SR (0.32% for FMA), while for MH, the maximum improvement is 1.08% (1.01% for FMA) against the observations of Shihab et al. [283]. To mitigate the dependency of response weights in WSMO approach, the related 3D Pareto optimal front is developed for GWO technique, as shown in Figure 9.4. The shape and spread of the Pareto front illustrate GWO's efficiency in exploring the solution space, emphasizing necessary trade-offs for simultaneous process optimization. In this plot, green points at the upper portion and red points at the lower portion represent non-dominated solutions forming the Pareto optimal front, reflecting the best possible trade-offs among the three objectives. The 'Optimal point', marked in black on the Pareto front, highlights the best compromised solution, further validating effectiveness of GWO.

Table 9.3 Multi-objective optimization results

Optimizer	Response	Min z	Process Parameters				Optimal	% improvement
			x_1	x_2	x_3	x_4		
Shihab et al. [283]	MRR (mm^3/min)	-	1000	0.0677	1.18687	12.697	2043.35	-
	SR (μm)						1.2942	
	MH (HV)						142.368	
GWO	MRR (mm^3/min)	0.373	1000	0.06	1.4	12.4	2292	12.17
	SR (μm)						1.16	10.37
	MH (HV)						143.9	1.08
AVOA	MRR (mm^3/min)	0.373	1000	0.06	1.4	12.4	2292	12.17
	SR (μm)						1.16	10.37
	MH (HV)						143.9	1.08
FMA	MRR (mm^3/min)	0.365	990	0.022	1.13	9.03	2127	4.09
	SR (μm)						1.29	0.32
	MH (HV)						143.8	1.01
EFO	MRR (mm^3/min)	0.373	1000	0.06	1.4	12.4	2292	12.17
	SR (μm)						1.16	10.37
	MH (HV)						143.9	1.08
TLBO	MRR (mm^3/min)	0.373	1000	0.06	1.4	12.4	2292	12.17
	SR (μm)						1.16	10.37
	MH (HV)						143.9	1.08

The parallel plot for GWO algorithm, shown in Figure 9.5, examines the impact of different parametric settings on the responses and their possible interactions. It confirms that to achieve the optimal performance, while minimizing SR (1.16 μm), and maximizing MRR (2292 mm^3/min) and MH (143.9 HV), the ideal end milling parameters as SS = 1000 rpm, FR = 0.06 mm/rev, DOC = 1.4 mm and concentration of SiC/Gr = 12.4% should be the optimal choice, while performing end milling operation on the hybrid Al7075/SiC/Gr composite material.

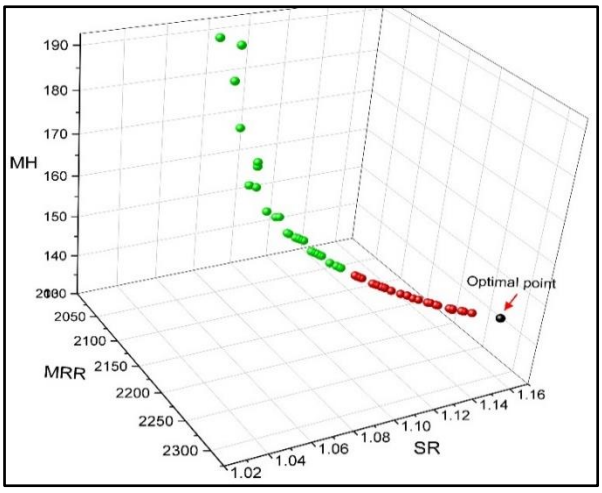


Figure 9.4 3D Pareto optimal front for GWO algorithm

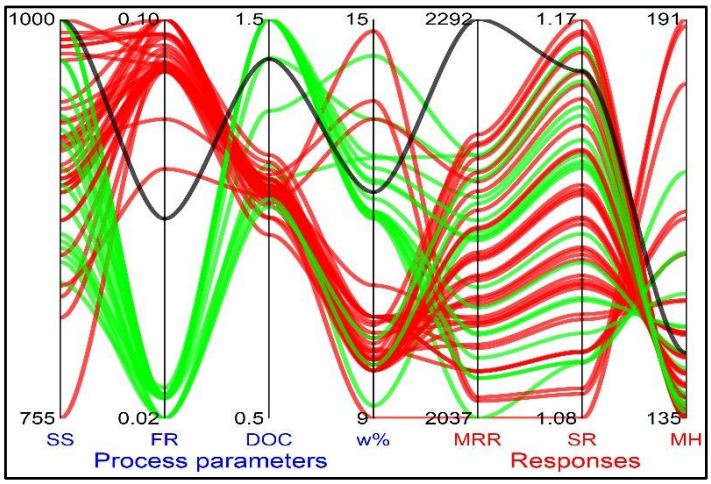


Figure 9.5 Parallel plot for GWO algorithm

Figure 9.6 compares the average computation time based on 30 independent runs of the five metaheuristic algorithms, while resolving the corresponding multi-objective optimization problem. Based on the mean computing time, the algorithms are ranked as GWO > TLBO > AVOA > EFO > FMA, with GWO demonstrating superior efficiency in identifying the optimal milling parameter combination with minimal computational effort.

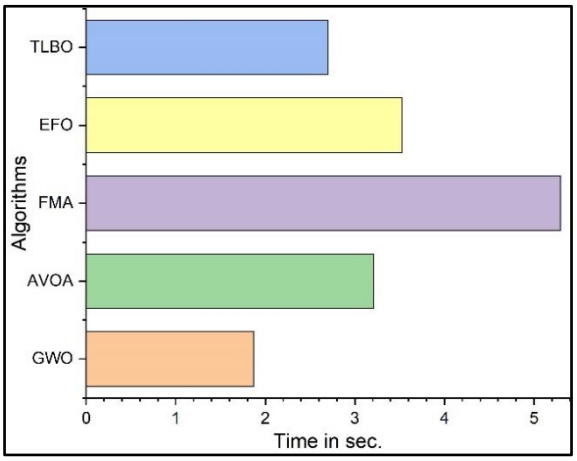


Figure 9.6 Mean computing time for multi-objective optimization

9.3 Performance analysis

This work also utilizes HV and SP values as the quality metrics to evaluate performance of the five metaheuristic algorithms. Table 9.4 highlights the best HV and SP values in bold, confirming that GWO outperforms its competitors. These results demonstrate GWO's exceptional capability in solving high-dimensional optimization problems with superior efficiency.

Table 9.4 HV and SP values

Algorithm	GWO	AVOA	FMA	EFO	TLBO
HV	**2237.2**	2206.37	2215.52	2197.26	2231.1
SP	**3.97**	4.07	4.27	4.13	3.99

To critically evaluate the performance of the five optimization algorithms, the Friedman's mean rank test is conducted at 1% significance level, as shown in Table 9.5, to determine statistically significant deviations between GWO, AVOA, FMA, EFO and TLBO optimizers. According to the results presented in Table 9.5, it is proved that GWO outperforms all other algorithms across the considered responses, while FMA demonstrates the worst performance.

Table 9.5 Friedman's mean rank test results

Algorithm	Response	GWO	AVOA	FMA	EFO	TLBO
Average	MRR	2731.09	2728.87	2721.39	2729.18	2729.86
Rank sum		269	306.5	334	307.5	283
Average rank		2.69	3.065	3.34	3.075	2.83
Rank		**1**	**3**	**5**	**4**	**2**
Average	SR	0.9632	0.9651	1.0584	0.9648	0.9637
Rank sum		226.5	275.5	500	258.5	239.5
Average rank		2.265	2.755	5	2.585	2.395
Rank		**1**	**4**	**5**	**3**	**2**
Average	MH	148.7101	148.6266	148.2543	148.5091	148.6363
Rank sum		266	298	338	316	282
Average rank		2.66	2.98	3.38	3.16	2.82
Rank		**1**	**3**	**4**	**5**	**2**

To further validate the significant difference between GWO and other algorithms, the Wilcoxon rank sum test is also conducted at 1% significance level. The results, presented in Table 9.6, reveal a notable distinction between GWO and its competitors. Table 9.6 provides the sum of negative ranks (R−), positive ranks (R+) and *p*-values for pair-wise comparisons with respect to GWO algorithm, reinforcing its superior performance.

Table 9.6 Results of the Wilcoxon rank sum test

GWO vs		AVOA	FMA	EFO	TLBO
R +	MRR	136	153	136	119
R –		3	7	4	1
p-value		4.40E-04	3.00E-04	2.20E-05	8.00E-04
R +	SR	153	505	153	118
R –		9	37	11	2
p-value		3.00E-04	1.00E-05	3.00E-04	0.001
R +	MH	136	190	153	136
R –		5	23	16	4
p-value		4.40E-04	1.40E-04	3.00E-04	4.40E-04

9.4 Results and discussion

This work investigates application of five metaheuristic algorithms for parametric optimization of an end milling operation on hybrid Al7075/SiC/Gr composite work material. These algorithms achieve improvements of 34.74% in MRR, 18.64 to 26.13% in SR and 5.07% in MH for single-objective optimization, while multi-objective optimization yields improvements of 4.09 to 12.17% for MRR, 0.32 to 10.37% for SR and 1.01 to 1.08% for MH, compared to the results reported by Shihab et al. [283]. Among the considered algorithms, GWO consistently delivers the best performance, proving to be the most efficient in both single- and multi-objective optimization of the said end milling process. In single-objective optimization, GWO attains the optimal values of 2753.18 mm^3/min for MRR, 0.956 μm for SR and 149.58 HV for MH, whereas, in multi-objective optimization, the corresponding optimal values are 2292 mm^3/min, 1.16 μm and 143.9 HV for MRR, SR and MH, respectively, at the best possible parametric combination of SS = 1000 rpm, FR = 0.06 mm/rev, DOC = 1.4 mm and concentration of SiC/Gr = 12.49%. GWO also surpasses other algorithms in convergence speed and computing time, requiring just 1.87 sec to find out the global optimal solution, reducing the runtime by 71.66, 183.42, 88.77 and 44.39% compared to AVOA, FMA, EFO and TLBO techniques, respectively. Its superiority is further confirmed using SP and HV values, as well as validation through Friedman's mean rank and Wilcoxon rank sum tests.

Table 9.7 highlights the differences between GWO and its competitors by comparing the performance in both the single- and multi-objective optimization problem, focusing on the percentage improvements in the response values and mean computing time. It depicts superior or same results across all the responses. For single-objective optimization, it performs similarly to the other four algorithms for MRR and MH, and AVOA, EFO and TLBO for SR. In multi-objective optimization, AVOA, EFO and TLBO exhibit comparable performance to GWO across all the responses. However, GWO significantly outperforms its competitors in terms of average computation time. Overall, the findings suggest that higher SS and DOC, along with moderate FR and percentage of SiC/Gr values, lead to concurrent optimization of all the responses, with GWO emerging out as the most efficient technique.

Table 9.7 Percentage improvements of the responses and computing time for GWO across its competitors

Optimizer	AVOA	FMA	EFO	TLBO	AVOA	FMA	EFO	TLBO
Technique	Single-objective				Multi-objective			
MRR	0	0	0	0	0	7.20	0	0
SR	0	10.15	0	0	0	11.21	0	0
MH	0	0	0	0	0	0.07	0	0
Time	17.95	145.91	52.92	79.15	71.66	183.42	88.77	44.39

The end milling operation is also optimized using some popular metaheuristic algorithms, such as GA, PSO, ABC, SA and ACO for both the single- and multi-objective optimization problems. The derived results, presented in Table 9.8, reveal that GWO achieves significant improvements (in percentage) in the optimal values of all the responses as well as average computational time.

Table 9.8 Comparison of performance of GWO against GA, PSO, ABC, SA and ACO

Optimizer	GA	PSO	ABC	SA	ACO	GA	PSO	ABC	SA	ACO
Technique	Single-objective					Multi-objective				
MRR	4.55	3.38	1.49	2.39	3.38	2.97	2.79	1.11	1.49	2.23
SR	5.12	4.57	2.19	3.66	3.84	2.22	3.80	1.43	2.22	1.74
SCEC	3.97	3.09	2.14	4.72	3.86	1.80	1.97	1.33	2.24	2.11
Average time	37.03	12.55	10.57	22.78	30.09	18.42	10.53	13.53	15.79	10.53

The considered algorithms are highly suitable for optimization of different machining operations due to their adaptability, efficiency and problem-solving capabilities. They excel in handling multi-objective, nonlinear and dynamic problems with minimal or no parameter tuning, as observed in TLBO and GWO, allowing them to outperform the traditional and nature-inspired techniques in complex machining environments. All the five algorithms demonstrate rapid convergence, ease of implementation and optimal performance in their respective applications. In contrast, GA and PSO rely on some specific algorithm parameters, such as crossover and mutation probabilities, inertia weight and cognitive/social coefficients (c_1 and c_2), which may significantly impact convergence of the optimal solutions. Overall, the considered algorithms surpass GA and PSO in terms of adaptability, robustness and simplicity, with GWO, AVOA, EFO and TLBO emerging as particularly advantageous, while GA and PSO remain valuable as benchmarks or for solving simpler optimization problems.

The practical feasibility of applying the five best algorithms to optimize both the traditional and NTM operations in industrial setups primarily depends on upgrading their control systems to integrate real-time data for continuous parameter adjustment. While initial challenges, such as machine limitations, tool wear and cost-effectiveness may arise, the achieved long-term benefits, like improved machining performance, reduced tool wear and cost savings make these algorithms a viable choice for the modern manufacturing environment. By addressing infrastructural requirements, including real-time feedback systems, sensor integration and system compatibility, the considered algorithms can significantly enhance machining performance in industrial setups.

10. CONCLUSIONS

This research work presents a comprehensive study on parametric optimization of five traditional and NTM processes using various classes of metaheuristic algorithms. Based on the past experimental datasets, five distinct categories of nature-inspired algorithms are applied to optimize processes, such as turning using five human-inspired algorithms (TLBO, SAR, TOA, HCO and QSA), green machining (dry milling) adopting five mating behavior-based algorithms (FMA, NMR, BOA, BMOA and AZOA), LBD using five popular preying behavior-based metaheuristic algorithms (SFO, HHO, AO, BA and GWO), AWJM employing five foraging behavior-based algorithms (AVOA, DOA, FOA, GOA and BSA) and PAC applying five unique physics-based algorithms (AOA, ASO, NRO, EFO and GSA). The efficacy of these algorithms is evaluated based on solution accuracy and variability, computational time, and convergence speed. Through detailed comparative analyses, the optimization performance of each algorithm is assessed. The following observations can be drawn from an in-depth analysis of the derived results:

a) For the considered turning process, TLBO outperforms other human-inspired algorithms by achieving an optimal mixture of the input variables. During single-objective optimization, TLBO results in 27.08, 36.4 and 31.16% improvements in MRR, SR and SCEC, respectively against those documented by the past researchers. In contrast, for multi-objective optimization, an optimal setting of CS = 434 m/min, FR = 0.17 mm/rev and DOC = 2.16 mm is obtained to attain the most favorable response values as MRR = 2679.1 mm3/s, SR = 0.631 μm and SCEC = 64.82 J/mm^3. Its supremacy may be due to its capability to resolve complex multimodal functions with quicker convergence rate while involving no tuning parameter.

b) In the green machining (dry milling) example, for single-objective optimization, FMA results in 28.57, 13.04 and 25% improvements in PF, EC and SR responses, respectively as compared to those measured by the past researchers. On the other hand, for multi-objective optimization, the most desired response values as PF = 0.87, EC = 19.11 kJ and SR = 0.34 μm, can be achieved at the optimal combination of CS = 160 m/min, DOC = 0.2 mm, FR = 0.085 mm/z and Nr = 0.6 mm.

c) Similarly, for CO_2-based micro-LBD operation on polycarbonate material, GWO emerges out as the most effective optimization tool. In single-objective optimization, it achieves improvements of 0% in Hh, 19.37% in Hd and 33.33% in HAZ compared to the earlier research work. For multi-objective optimization, the optimal settings of Pl = 3.95 W, ET = 0.1 s and FPP = - 0.91 mm, result in the most favorable response values as Hh = 1 mm, Hd = 308 μm and HAZ = 86.45 μm. GWO's superiority over the other adopted algorithms may be due to its simple principle and easy realization, fast seeking speed and high search accuracy.

d) Furthermore, in the case of AWJM on $LaPO_4/Y_2O_3$ composite, AVOA proves to be the most efficient optimization technique. Single-objective optimization results indicate significant improvements of 71.47, 43.73 and 10% in MRR, KA and SR, respectively as compared to the past observations. During multi-objective optimization, an optimal combination of AWJM parameters

as WJP = 220 bar, SOD = 2 mm and St = 20 mm/min would effectively optimize the said process with identification of the corresponding responses as MRR = 0.0362 g/s, KA = 0.203° and SR = 1.158 µm. AVOA's ability to extensively explore and exploit the search space, along with the use of a random search technique, enhances its efficiency in determining the global parametric settings for the AWJM process.

e) Additionally, EFO exceeds the other physics-based algorithms in achieving the optimal intermix of the considered PAC parameters. In the single-objective optimization, EFO results in 42.50, 35.85 and 25.55% improvements in MRR, KT and HAZ, respectively, compared to the previous research work. On the other hand, during multi-objective optimization, an optimal setting of CS = 2554 mm/min, Pg = 4 bar, AC = 45.2 A and SOD = 2.44 mm would lead to the most desirable response values as MRR = 42.34 g/min, KT = 6.07° and HAZ = 3.74 mm. EFO's superiority can be attributed to its ability to efficiently solve complex multimodal functions with faster convergence rate while requiring minimal tuning parameters.

f) Finally, the optimization performance of the best five algorithms from five distinct categories of metaheuristics is validated using the experimental data from an end milling process. Among these, GWO emerges as the most effective algorithm, yielding the best values for the considered responses. For single-objective optimization, GWO achieves improvements of 34.74% in MRR, 26.13% in SR and 5.07% in MH compared to the previous research work. In multi-objective optimization, the optimal parameter settings of SS = 1000 rpm, FR = 0.06 mm/rev, DOC = 1.4 mm and SiC/Gr = 12.4% are observed to achieve the most desired response values as MRR = 2292 mm^3/min, SR = 1.16 µm and MH = 143.9 HV.

g) Validation of the performance analysis further supports superior performance of the best algorithms, as evidenced by the maximum HV and minimum SP values compared to their competitors. The results of both the Friedman's mean rank and the Wilcoxon rank sum tests identify the best-performing algorithms in optimizing various machining processes, with the corresponding *p*-values indicating significant differences among the considered algorithms.

h) Additionally, all the selected algorithms demonstrate superior performance when benchmarked against other the state-of-the-art algorithms, validating their efficiency in delivering high quality solutions with minimal computational effort.

i) Consequently, GWO proves to be a highly effective and powerful tool in optimizing a wide range of machining processes.

Since all the machining processes are optimized using past experimental data, there is no scope in this research work to conduct real-time confirmatory experiments to validate the derived results. While this research work compares the relative performance of five distinct metaheuristic techniques, future studies may concentrate on comparing performance against other metaheuristic techniques, such as energy valley optimizer, flow direction algorithm, equilibrium optimizer, multi-verse optimizer, predator-prey optimizer, bacterial foraging optimization algorithm, elephant swarm water search

algorithm, manta ray foraging algorithm, battle royale optimization, war strategy optimization, coronavirus herd immunity optimization etc. Further research may also explore effects of varying values of different algorithm-specific parameters and response weight assignments on optimization performance. Additionally, evaluating these algorithms' effectiveness in addressing many-objective optimization problems may be beneficial. Future research should also emphasize exploring applicability of various ML algorithms in modeling the relationships between different input parameters and machining qualities, ultimately enhancing precision and process efficiency. Finally, parameter tuning techniques, hybridizing metaheuristics and leveraging advanced optimization methods may further enhance the performance of the considered algorithms.

REFERENCES

1. Hasan, M., Zhao, J., Jiang, Z.: A review of modern advancements in micro drilling techniques. Journal of Manufacturing Processes, 29, 343-375 (2017)
2. Hatala, M., Imrich O.: Mathematical modelling of plasma arc cutting technological process. In: Proc. of 13th International Research/Expert Conference on Trends in the Development of Machinery and Associated Technology, Tunisia, 65-68 (2009)
3. Ananthakumar, K., Rajamani, D., Balasubramanian, E., Davim, J.P.: Measurement and optimization of multi-response characteristics in plasma arc cutting of Monel 400™ using RSM and TOPSIS. Measurement, 135, 725-737 (2019)
4. Davim, J.P.: Nontraditional Machining Processes, Manufacturing Process Selection. Springer, London, 205-226 (2013)
5. Kalita, K., Chakraborty, S., Ghadai, R.K., Chakraborty, S.: Parametric optimization of non-traditional machining processes using multi-criteria decision making techniques: Literature review and future directions. Multiscale and Multidisciplinary Modeling, Experiments and Design, 6(1), 1-40 (2023)
6. Nair, A., Kumanan, S., Prakash, C., Mohan, D.G., Saxena, K.K., Kumar, S., Kumar, G.: Research developments and technological advancements in conventional and non-conventional machining of superalloys - a review. Journal of Adhesion Science and Technology, 37(22), 3053-3124 (2023)
7. Das, S., Chakraborty, S.: Selection of non-traditional machining processes using analytic network process. Journal of Manufacturing Systems, 30(1), 41-53 (2011)
8. Oza, A.D., Goyal, A., Buch, V., Kumar, M.: Electrochemical discharge machining process: a review on process parameters and future scope. Materials Today Proceedings, 62(13), 6956-6961 (2022)
9. Sadhu, A., Chakraborty, S.: Non-traditional machining processes selection using data envelopment analysis (DEA). Expert Systems Applications, 38(7), 8770-8781 (2011)
10. Dehestany, A., Pervaiz, S.: Multi-parametric optimization of convex spherical surface drilling of Ti6Al4V using finite element model and Taguchi coupled desirability function analysis. International Journal on Interactive Design and Manufacturing, 1-16 (2024) https://doi.org/10.1007/s12008-024-01790-6
11. Dash, L., Padhan, S., Das, A., Das, S.R.: Machinability investigation and sustainability assessment in hard turning of AISI D3 steel with coated carbide tool under nanofluid minimum quantity lubrication-cooling condition. Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science, 235(22), 6496-6528 (2021)
12. Jain, A., Kansal, H.: Green machining-machining of the future. In: Proc. of 4th National Conference on Advancements in Simulation & Experimental Techniques in Mechanical Engineering, India, 21-25 (2017)

13. Das, P.P., Chakraborty, S.: SWARA-CoCoSo method-based parametric optimization of green dry milling processes. Journal of Engineering and Applied Science, 69(1), 1-21 (2022)
14. Chatterjee, S., Das, P.P., Chakraborty, S.: Optimization of green machining processes using grey-based multi-criteria decision making methods: a comparative analysis. International Journal on Interactive Design and Manufacturing, 18(1), 33-53 (2024)
15. Boubekri, N., Shaikh, V., Foster, P.R.: A technology enabler for green machining: minimum quantity lubrication (MQL). Journal of Manufacturing Technology Management, 21(5), 556-566 (2010)
16. Liu, C., He, Y., Wang, Y., Li, Y., Wang, S., Wang, L., Wang, Y.: Effects of process parameters on cutting temperature in dry machining of ball screw. ISA Transactions, 101, 493-502 (2020)
17. Rana, P.B., Lalwani, D.I.: Optimization of turning process using amended differential evolution algorithm. Engineering Science and Technology, an International Journal, 20(4), 1285-1301 (2017)
18. Sousa, V.F., Silva, F.J.: Recent advances in turning processes using coated tools - A comprehensive review. Metals, 10(2), 170 (2020)
19. Yang, W.P., Tarng, Y.S.: Design optimization of cutting parameters for turning operations based on the Taguchi method. Journal of Materials Processing Technology, 84(1-3), 122-129 (1998)
20. Mohan, R., Harshavardhana, N., Chaudhari, M., Jeyanthi, S., Abimannan, G.: Analysis on surface finish and chip morphology during dry turning process. Materials Today: Proceedings, 46, 999-1002 (2021)
21. Shanmugasundar, G., Mahanta, T.K., Čep, R., Kalita, K.: Novel fuzzy measurement alternatives and ranking according to the compromise solution-based green machining optimization. Processes, 10(12), 2645 (2022)
22. Pattanayak, S., Panda, S.: Laser beam micro drilling - a review. Lasers in Manufacturing and Materials Processing 5(4), 366-394 (2018)
23. Bachy, B.: Laser micro-drilling process: experimental investigation, modelling based on RSM-BBD and multi-criteria optimization. Journal of the Brazilian Society of Mechanical Sciences and Engineering, 45(3), 180 (2023)
24. Rajesh, P., Nagaraju, U., Gowd, G.H., Vardhan, T.V.: Experimental and parametric studies of Nd:YAG laser drilling on austenitic stainless steel. International Journal of Advanced Manufacturing Technology, 93, 65-71 (2017)
25. Pawar, P.J., Vidhate, U.S., Khalkar, M.Y.: Improving the quality characteristics of abrasive water jet machining of marble material using multi-objective artifcial bee colony algorithm. Journal of Computational Design and Engineering, 5(3), 319-328 (2018)
26. Tiwari, T., Sourabh, S., Nag, A., Dixit, A.R., Mandal, A., Das, A.K., Mandal, N., Srivastava, A.K.: Parametric investigation on abrasive waterjet machining of alumina ceramic using response surface methodology. Materials Science and Engineering, 377, 1-8 (2018)

27. Fuse, K., Chaudhari, R., Vora, J., Patel, V.K., De-Lacalle, L.N.L.: Multi-response optimization of abrasive waterjet machining of Ti6Al4V using integrated approach of utilized heat transfer search algorithm and RSM. Materials, 14(24), 1-22 (2021)
28. Sharun, V., Ronald, B.A.: Optimization of process parameters in abrasive water jet machining of austempered ductile iron (ADI). Journal of Materials Engineering and Performance, 33(10), 4867-4882 (2024)
29. Das, P.P., Chakraborty, S.: A comparative assessment of multi-criteria parametric optimization methods for plasma arc cutting processes. Decision Analytics Journal, 6, 100190 (2023)
30. Bini, R., Colosimo, B.M., Kutlu, A.E., Monno, M.: Experimental study of the features of the kerf generated by a 200 A high tolerance plasma arc cutting system. Journal of Materials Processing Technology, 196(1-3), 345-355 (2008)
31. Rana, K., Kaushik, P., Chaudhary, S.: Optimization of plasma arc cutting by applying Taguchi method. International Journal of Enhanced Research in Science Technology & Engineering, 2(7), 106-110 (2013)
32. Mangaraj, S.R., Bagal, D.K., Parhi, N., Panda, S.N., Barua, A., Jeet, S.: Experimental study of a portable plasma arc cutting system using hybrid RSM-nature inspired optimization technique. Materials Today: Proceedings, 50, 867-878 (2022)
33. Cinar, Z., Asmael, M., Zeeshan, Q.: Developments in plasma arc cutting (PAC) of steel alloys: a review. Jurnal Kejuruteraan, 30(1), 7-16 (2018)
34. Maity, K.P., Bagal, D.K.: Effect of process parameters on cut quality of stainless steel of plasma arc cutting using hybrid approach. International Journal of Advanced Manufacturing Technology, 78(1), 161-175 (2015)
35. Zheng, M., Yu, J.: Parameter optimization design of plasma arc machining SS304 alloy by means of probabilistic multi-objective optimization. Materials Plus, 2(2), 1-6 (2023)
36. Ilii, S.M., Coteata, M., Munteanu, A.: Experimental results concerning the variation of surface roughness parameter (Ra) at plasma arc cutting of a stainless steel workpiece. International Journal of Modern Manufacturing Technologies, 2(1), 31-36 (2010)
37. Siva Kumar, M., Rajamani, D., Abouel Nasr, E., Balasubramanian, E., Mohamed, H., Astarita, A.: A hybrid approach of ANFIS-artificial bee colony algorithm for intelligent modeling and optimization of plasma arc cutting on Monel™ 400 alloy. Materials, 14(21), 6373 (2021)
38. Devaraj, R., Abouel Nasr, E., Esakki, B., Kasi, A., Mohamed, H.: Prediction and analysis of multi-response characteristics on plasma arc cutting of Monel 400™ alloy using Mamdani-fuzzy logic system and sensitivity analysis. Materials, 13(16), 3558 (2020)
39. Abualigah, L., Elaziz, M.A., Khasawneh, A.M., Alshinwan, M., Ibrahim, R.A., Al-Qaness, M.A., Mirjalili, S., Sumari, P., Gandomi, A.H.: Meta-heuristic optimization algorithms for solving real-world mechanical engineering design problems: a comprehensive survey, applications, comparative analysis, and results. Neural Computing & Applications, 34, 4081-4110 (2022)

40. Abualigah, L., Diabat, A., Geem, Z.W.: A comprehensive survey of the harmony search algorithm in clustering applications. Applied Sciences, 10(11), 3827 (2020)
41. Dokeroglu, T., Sevinc, E., Kucukyilmaz, T., Cosar, A.: A survey on new generation metaheuristic algorithms. Computers & Industrial Engineering, 137, 106040 (2019)
42. Maier, H.R., Razavi, S., Kapelan, Z., Matott, L.S., Kasprzyk, J., Tolson, B.A.: Introductory overview: Optimization using evolutionary algorithms and other metaheuristics. Environmental Modelling & Software, 114, 195-213 (2019)
43. Osaba, E., Villar-Rodriguez, E., Del Ser, J., Nebro, A.J., Molina, D., LaTorre, A., Suganthan, P.N., Coello Coello, C.A., Herrera, F.: A tutorial on the design, experimentation and application of metaheuristic algorithms to real-world optimization problems. Swarm and Evolutionary Computation, 64, 100888 (2021)
44. Yang, X.S.: Metaheuristic optimization. Scholarpedia, 6, 11472 (2011)
45. Hussain, K., MohdSalleh, M.N., Cheng, S., Shi, Y.: Metaheuristic research: a comprehensive survey. Artificial Intelligence Review, 52, 2191-2233 (2019)
46. Almufti, S.M., Shaban, A.A., Ali, Z.A., Ali, R.I., Fuente, J.D.: Overview of metaheuristic algorithms. Polaris Global Journal of Scholarly Research and Trends, 2(2), 10-32 (2023)
47. Abdel-Basset, M., Abdel-Fatah, L., Sangaiah, A.K.: Metaheuristic algorithms: A comprehensive review. Computational intelligence for multimedia big data on the cloud with engineering applications, 185-231 (2018) https://doi.org/10.1016/B978-0-12-813314-9.00010-4
48. Rajwar, K., Deep, K., Das, S.: An exhaustive review of the metaheuristic algorithms for search and optimization: taxonomy, applications, and open challenges. Artificial Intelligence Review, 56(11), 13187-13257 (2023)
49. Desale, S., Rasool, A., Andhale, S., Rane, P.: Heuristic and meta-heuristic algorithms and their relevance to the real world: a survey. International Journal of Computer Engineering in Research Trends, 2(5), 296-304 (2015)
50. Zhang, J., Khayatnezhad, M., Ghadimi, N.: Optimal model evaluation of the proton-exchange membrane fuel cells based on deep learning and modifed African vulture optimization algorithm. Energy Sources, Part A: Recovery, Utilization, and Environmental Effects, 44(1), 287-305 (2022)
51. Kurdi, H., Al-Daood, M.F., Al-Megren, S., Aloboud, E., Aldawood, A.S., Youcef-Toumi, K.: Adaptive task allocation for multi-UAV systems based on bacteria foraging behaviour. Applied Soft Computing, 83, 1-18 (2019)
52. Radovanovic, M.: Multi-objective optimization of abrasive water jet cutting using MOGA. Procedia Manufacturing, 47, 781-787 (2020)
53. Khanduja, N., Bhushan, B.: Recent advances and application of metaheuristic algorithms: a survey (2014-2020). Metaheuristic and Evolutionary Computation: Algorithms and Applications, 916, 207-228 (2021)

54. Yang, X.S.: Nature-inspired Metaheuristic Algorithms. Luniver Press, Bristol (2010)
55. Diyaley, S., Chakraborty, S.: Metaheuristics-based parametric optimization of multi-pass turning process: a comparative analysis. Opsearch, 57(2), 414-437 (2020)
56. Turgut, O.E., Turgut, M.S., Kırtepe, E.: A systematic review of the emerging metaheuristic algorithms on solving complex optimization problems. Neural Computing & Applications, 35(19), 14275-14378 (2023)
57. Kalita, K., Ghadai, R.K., Chakraborty, S.: A comparative study on the metaheuristic-based optimization of skew composite laminates. Engineering with Computers, 38(4), 3549-3566 (2022)
58. Laghari, R.A., Li, J., Mia, M.: Effects of turning parameters and parametric optimization of the cutting forces in machining SiCp/Al 45 wt% composite. Metals, 10(6), 840 (2020)
59. Abolghasema, S., Cubides, N.M.: Optimization of machining parameters for product quality and productivity in turning process of aluminum. Ingeniería y Universidad, 26, 1-27 (2022)
60. Zhang, J., Liang, S.Y., Yao, J., Chen, J.M., Huang, J.L.: Evolutionary optimization of machining processes. Journal of Intelligent Manufacturing, 17, 203-215 (2006)
61. Hema, P., Ganesan, R.: Experimental investigations on SS 304 alloy using plasma arc machining. SN Applied Sciences, 2, 1-16 (2020)
62. Liu, S., Cheng, H.: Manufacturing process optimization in the process industry. International Journal of Information Technology and Web Engineering, 19(1), 1-20 (2024)
63. Wang, Y., Ma, S., Wang, J., Wang, H., Pu, W.: Machine manufacturing process improvement and optimisation. Highlights in Science, Engineering and Technology, 62, 129-132 (2023)
64. Chukwunweike, J., Anang, A.N., Adeniran, A.A., Dike, J.: Enhancing manufacturing efficiency and quality through automation and deep learning: addressing redundancy, defects, vibration analysis, and material strength optimization. World Journal of Advanced Research and Reviews. 23(03), 1272-1295 (2024)
65. Abderazek, H., Laouissi, A., Nouioua, M., Atanasovska, I.: Optimization of turning process parameters using a new hybrid evolutionary algorithm. Journal of Mechanical Engineering Science, 238, 758-768 (2024)
66. Arunkumar, E., Devendiran, S.: Investigation, optimising the MQL-turning parameters of Nimonic 75 using weighted mayfly algorithm. Advances in Materials and Processing Technologies, 10, 1556-1573 (2023)
67. Arunkumar, E., Devendiran, S.: Investigation and optimization of multiple objectives by hybrid evolutionary algorithms for turning of Nimonic 80A under nano MQL environment. Advances in Materials and Processing Technologies, 11(1), 488-522 (2024)
68. Doan, T.K., Nguyen, T.T., Van, A.L.: Multi-performance optimization of the rotary turning operation for environmental and quality indicators. Strojniški Vestnik - Journal of Mechanical Engineering, 70, 42-54 (2024)

69. Doan, T.K., Nguyen, T.T., Van, A.L.: Multi-objective optimization of the rotary turning of hardened mold steel for energy saving and surface roughness improvements. Journal of Machine Engineering, 23, 101-121 (2023)
70. Shastri, A., Nargundkar, A., Kulkarni, A.J., Benedicenti, L.: Optimization of process parameters for turning of titanium alloy (Grade II) in MQL environment using multi-CI algorithm. SN Applied Sciences, 3, 1-12 (2021)
71. Elsheikh, A.H., AbdElaziz, M., Das, S.R., Muthuramalingam, T., Lu, S.: A new optimized predictive model based on political optimizer for eco-friendly MQL-turning of AISI 4340 alloy with nano-lubricants. Journal of Manufacturing Processes, 67, 562-578 (2021)
72. Xie, N., Zhou, J., Zheng, B.: An energy-based modeling and prediction approach for surface roughness in turning. International Journal of Advanced Manufacturing Technology, 96, 2293-2306 (2018)
73. Jeet, S., Barua, A., Bagal, D.K., Pradhan, S., Patnaik, D., Pattanaik, A.K.: Comparative investigation of CNC turning of nickel-chromoly steel under different cutting environment with a fabricated portable mist lubricator: A super hybrid taguchi-WASPAS-GA-SA-PSO approach. In: Advanced Manufacturing Systems and Innovative Product Design: Select Proceedings of IPDIMS 2020. Springer, Singapore, 43, 515-531 (2021)
74. Zohra, E.F., Jabri, A., El Barkany, A., Boharb, A., Moujibi, N.: Experimental study and optimization of energy consumption during end face turning of AISI 4140. In Proc. of 4th International Conference on Innovative Research in Applied Science, Engineering and Technology, Morocco, 1-4 (2024)
75. Noor, K., Siddiqui, M.A., Syed, A.I.: Optimization of specific energy consumption in CNC turning of a hardened alloy steel roll at low cutting speeds. Journal of Engineering, Design and Technology, 21, 1-18 (2023)
76. Nguyen, T.T., Duong, Q.D., Mia, M.: Sustainability-based optimization of the rotary turning of the hardened steel. Metals, 10, 939 (2020)
77. Zębala, W., Struzikiewicz, G., Słodki, B.: Reduction of power consumption by chip breakability control in Ti6Al4V titanium alloy turning. Materials, 13, 2642 (2020)
78. Vukelic, D., Simunovic, K., Simunovic, G., Saric, T., Kanovic, Z., Budak, I., Agarski, B.: Evaluation of an environment-friendly turning process of Inconel 601 in dry conditions. Journal of Cleaner Production, 266, 121919 (2020)
79. Usha, M., Rao, G.S.: Optimisation of parameters in turning using herbal based nano cutting fluid with MQL. Materials Today: Proceedings, 22, 1535-1544 (2020)
80. Usha, M., Rao, G.S.: Optimization of multiple objectives by genetic algorithm for turning of AISI 1040 steel using Al2O3 nano fluid with MQL. Tribology in Industry, 42, 70 (2020)
81. Laouissi, A., Yallese, M.A., Belbah, A., Belhadi, S., Haddad, A.: Investigation, modeling, and optimization of cutting parameters in turning of gray cast iron using coated and uncoated silicon

nitride ceramic tools. Based on ANN, RSM, and GA optimization. International Journal of Advanced Manufacturing Technology, 101, 523-548 (2019)

82. Serra, R., Chibane, H., Duchosal, A.: Multi-objective optimization of cutting parameters for turning AISI 52100 hardened steel. International Journal of Advanced Manufacturing Technology, 99, 2025-2034 (2018)
83. Abbas, A.T., Al-Abduljabbar, A.A., El Rayes, M.M., Benyahia, F., Abdelgaliel, I.H., Elkaseer, A.: Multi-objective optimization of performance indicators in turning of AISI 1045 under dry cutting conditions. Metals, 13, 96 (2023)
84. Fountas, N., Koutsomichalis, A., Kechagias, J., Vaxevanidis, N.: Multi-response optimization of CuZn39Pb3 brass alloy turning by implementing grey wolf algorithm. Fracture and Structural Integrity, 13, 584-594 (2019)
85. Nouioua, M., Laouissi, A., Brahami, R., Blaoui, M.M., Hammoudi, A., Yallese, M.A.: Evaluation of: MOSSA, MOALO, MOVO and MOGWO algorithms in green machining to enhance the turning performances of X210Cr12 steel. International Journal of Advanced Manufacturing Technology, 120, 2135-2150 (2022)
86. Nguyen, T.T.: An energy-efficient optimization of the hard turning using rotary tool. Neural Computing and Applications, 33, 2621-2644 (2021)
87. Tanvir, M.H., Hussain, A., Rahman, M.T., Ishraq, S., Zishan, K., Rahul, S.T.T., Habib, M.A. Multi-objective optimization of turning operation of stainless steel using a hybrid whale optimization algorithm. Journal of Manufacturing and Materials Processing, 4, 64 (2020)
88. Hakmi, T., Hamdi, A., Laouissi, A., Abderazek, H., Chihaoui, S., Yallese, M.A.: Mathematical Mmodeling using ANN based on k-fold cross validation approach and MOAHA multi-objective optimization algorithm during turning of polyoxymethylene POM-C. Jordan Journal of Mechanical and Industrial Engineering, 18, 179-190 (2024)
89. Oussama, B., Yapan, Y.F., Uysal, A., Abdelhakim, C., Mourad, N.: Assessment of turning AISI 316L stainless steel under MWCNT-reinforced nanofluid-assisted MQL and optimization of process parameters by NSGA-II and TOPSIS. International Journal of Advanced Manufacturing Technology, 127, 3855-3868 (2023)
90. Chihaoui, S., Meddour, I., AthmaneYallese, M., Belhadi, S., Safi, K.: Comparative assessment between DFA, NSGA-II coupled with TOPSIS, GRA, and TOPSIS for multiobjective optimization of gray cast iron turning process using CBN insert. Journal of Process Mechanical Engineering, 238, 1765-1878 (2023)
91. Benkhelifa, O., Nouioua, M., Cherfia, A.: Monitoring and optimization of the machining process when turning of AISI 316L based on RSM-DF and ANN-NGSAII approaches. Research Square, 1-40 (2022) https://doi.org/10.21203/rs.3.rs-1276720/v1

92. Jia, S., Wang, S., Lv, J., Cai, W., Zhang, N., Zhang, Z., Bai, S.: Multi-objective optimization of CNC turning process parameters considering transient-steady state energy consumption. Sustainability, 13, 13803 (2021)
93. Risco-Alfonso, R.D., Pérez-Rodríguez, R., Zambrano Robledo, P.D.C., Rivas Santana, M., Quiza, R.: Optimization of the cutting regime in the turning of the AISI 316L steel for biomedical purposes based on the initial progression of tool wear. Metals, 11, 1698 (2021)
94. Bagaber, S.A., Yusoff, A.R.: Energy and cost integration for multi-objective optimisation in a sustainable turning process. Measurement, 136, 795-810 (2019)
95. Pendokhare, D., Chakraborty, S.: Human-inspired metaheuristics-based optimization of turning of AISI 6061-T6 aluminium. Journal of The Institution of Engineers (India): Series C, 1-28 (2025) https://doi.org/10.1007/s40032-025-01185-w
96. Zahoor, S., Abdul-Kader, W., Shehzad, A., Habib, M.S.: Milling of Inconel 718: an experimental and integrated modeling approach for surface roughness. International Journal of Advanced Manufacturing Technology, 120, 1609-1624 (2022)
97. Mumtaz, J., Li, Z., Imran, M., Yue, L., Jahanzaib, M., Sarfraz, S., Shehab, E., Ismail, S. O., Afzal, K.: Multi-objective optimisation for minimum quantity lubrication assisted milling process based on hybrid response surface methodology and multi-objective genetic algorithm. Advances in Mechanical Engineering, 11, 1-13 (2019)
98. Moreira, L.C., Li, W.D., Lu, X., Fitzpatrick, M.E.: Energy-efficient machining process analysis and optimisation based on BS EN24T alloy steel as case studies. Robotics and Computer-Integrated Manufacturing, 58, 1-12 (2019)
99. Deng, Z., Zhang, H., Fu, Y., Wan, L., Liu, W.: Optimization of process parameters for minimum energy consumption based on cutting specific energy consumption. Journal of Cleaner Production, 166, 1407-1414 (2017)
100. Alrashdan, A., Bataineh, O., Shbool, M.: Multi-criteria end milling parameters optimization of AISI D2 steel using genetic algorithm. International Journal of Advanced Manufacturing Technology, 73, 1201-1212 (2014)
101. Kadirgama, K., Noor, M.M., Rahman, M.M.: Optimization of surface roughness in end milling on mould aluminium alloys (AA6061-T6) using response surface method and radian basis function network. Jourdan Journal of Mechanical and Industrial Engineering, 2, 209-214 (2008)
102. Jia, S., Wang, S., Zhang, N., Cai, W., Liu, Y., Hao, J., Sui, Y.: Multi-objective parameter optimization of CNC plane milling for sustainable manufacturing. Environmental Science and Pollution Research, 1-22 (2022) https://doi.org/10.1007/s11356-022-24908-3
103. Mishra, R., Singh, B.: SBLMD-ANN-MOPSO-based hybrid approach for determining optimum parameter in CNC milling. Soft Computing, 27, 7299-7320 (2023)

104. Li, C., Xiao, Q., Tang, Y., Li, L.: A method integrating Taguchi, RSM and MOPSO to CNC machining parameters optimization for energy saving. Journal of Cleaner Production, 135, 263-275 (2016)
105. Jang, D.Y., Jung, J., Seok, J.: Modeling and parameter optimization for cutting energy reduction in MQL milling process. International Journal of Precision Engineering and Manufacturing-Green Technology, 3, 5-12 (2016)
106. Vu, N.C., Dang, X.P., Huang, S.C.: Multi-objective optimization of hard milling process of AISI H13 in terms of productivity, quality, and cutting energy under nanofluid minimum quantity lubrication condition. Measurement and Control, 54, 820-834 (2021)
107. Sen, B., Debnath, S., Bhowmik, A.: Sustainable machining of superalloy in minimum quantity lubrication environment: leveraging GEP-PSO hybrid optimization algorithm. International Journal of Advanced Manufacturing Technology, 130, 4575-4601 (2024)
108. Sen, B., Hussain, S.A.I., Gupta, M.K., Mia, M., Mandal, U.K.: Swarm intelligence based selection of optimal end-milling parameters under minimum quantity nano-green lubricating environment. Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science, 235, 6969-6983 (2021)
109. Song, J., Jin, H., Wang, X., Hu, T., Han, Z., Fu, H.: Research on digital twin model for milling parameter optimization of thin-walled parts. International Journal of Advanced Manufacturing Technology, 136, 3803-3819 (2025)
110. Shi, K-N., Liu, N., Liu, C-L., Ren, J.X., Yang, S-S., Tan, W.C.: Indirect approach for predicting cutting force coefficients and power consumption in milling process. Advances in Manufacturing, 10, 101-113 (2022)
111. Li, C., Li, L., Tang, Y., Zhu, Y., Li, L.: A comprehensive approach to parameters optimization of energy-aware CNC milling. Journal of Intelligent Manufacturing, 30, 123-138 (2019)
112. Zhang, W., Sun, X., Yang, H., Liu, Y., Dong, Z., Zhao, H., Mu, S., Chen, J.: A process parameters decision approach considering spindle vibration in helical surface milling for minimising energy consumption and surface roughness value. Journal of the Brazilian Society of Mechanical Sciences and Engineering, 46, 675 (2024)
113. Mishra, A., Yau, H.T., Kuo, P.H., Wang, C.C.: Achieving sustainability by identifying the influences of cutting parameters on the carbon emissions of a milling process. International Journal of Advanced Manufacturing Technology, 135, 5409-5427 (2024)
114. Song, L., Yan, C., Tu, G., Xiang, M., Liu, Y.: Prediction and optimization of surface roughness in high-speed dry milling of 30CrMnSiNiA using GPR and MOHHO algorithm. International Journal of Advanced Manufacturing Technology, 128, 4357-4377 (2023)
115. Kumar, S., Dubey, A.K., Pandey, A.K.: Computer-aided genetic algorithm based multi-objective optimization of laser trepan drilling. International Journal of Precision Engineering and Manufacturing, 14, 1119-1125 (2013)

116. Singh, S.K., Gangwar, S.: Parametric optimization of cutting parameters of laser assisted cutting using Taguchi and genetic algorithm. Journal of Future Engineering & Technology, 11, 36-42 (2016)
117. Gowd, G.H., Rajesh, U.N.P., Vardhan, T.V.: Modeling & optimization of laser beam drilling process using genetic algorithm. International Journal of Mechanical and Mechatronics Engineering, 17, 39-47 (2017)
118. Kalita, K., Shivakoti, I., Ghadai, R.K.: Optimizing process parameters for laser beam micro-marking using a genetic algorithm and particle swarm optimization. Materials and Manufacturing Processes, 32, 1101-1108 (2017)
119. Shrivastava, P.K., Pandey, A.K.: Multi-objective optimization of cutting parameters during laser cutting of titanium alloy sheet using hybrid approach of genetic algorithm and multiple regression analysis. Materials Today: Proceedings, 5, 24710-24719 (2018)
120. Chatterjee, S., Mahapatra, S.S., Bharadwaj, V., Choubey, A., Upadhyay, B.N., Bindra, K.S.: Quality evaluation of micro drilled hole using pulsed Nd:YAG laser: a case study on AISI 316. Lasers in Manufacturing and Materials Processing, 5, 248-269 (2018)
121. Gautam, G.D., Mishra, D.R.: Evaluation of geometrical quality characteristics in pulsed Nd:YAG laser cutting of Kevlar-29/basalt fiber reinforced hybrid composite using grey relational analysis based on genetic algorithm. FME Transactions, 47, 560-575 (2019)
122. Tura, A.D., Mamo, H.B., Desisa, D.G.: Multi-objective optimization and analysis for laser beam cutting of stainless steel (SS304) using hybrid statistical tools GA-RSM. IOP Conference Series: Materials Science and Engineering, 1201, 012030 (2021)
123. Zhang, Z., Liu, S., Zhang, Y., Wang, C., Zhang, S., Yang, Z., Xu, W.: Optimization of low-power femtosecond laser trepan drilling by machine learning and a high-throughput multi-objective genetic algorithm. Optics and Laser Technology, 148, 107688 (2022)
124. Du, T., Liang, X., Yu, Y., Zhou, L., Cai, Z., Wang, L., Jia, W., Pan, X.: Optimization of femtosecond laser drilling process of DD6 single crystal alloy. Metals, 13, 333 (2023)
125. Zhao, Z., Yu, Y., Sun, R., Zhao, W., Guo, H., Zhang, Z., Wang, C.: Design of a femtosecond laser percussion drilling process for N-based superalloys based on machine learning and the genetic algorithm. Micromachines, 14, 2110 (2023)
126. Sahoo, A.K., Mishra, D.R.: Parametric optimization of response parameter of Nd-YAG laser drilling for basalt-PTTE coated glass fibre using genetic algorithm. Journal of Engineering Research, 13(1), 97-110 (2025)
127. Mahesh, G.G., Kandasamy, J.: Optimization of CO_2 laser drilling process parameters of GFRP/Al2O3/perlite composites. Materials Today Communications, 35, 105962 (2023)
128. Chatterjee, S., Mahapatra, S.S., Bharadwaj, V., Upadhyay, B.N., Bindra, K.S.: Prediction of quality characteristics of laser drilled holes using artificial intelligence techniques. Engineering with Computers, 37, 1181-1204 (2021)

129. Yadav, A., Dubey, A.K., Sharma, S.K., Upadhyaya, B.N.: Precision characterization of hole attributes in Nd: YAG laser percussion drilling of Al/SiC MMC. Proceedings of the Institution of Mechanical Engineers, Part E: Journal of Process Mechanical Engineering, 1-13 (2025) https://doi.org/10.1177/09544089251321931
130. Sahoo, A.K., Mishra, D.R.: Characterisation of basalt/glass/kevlar-29 hybrid fibre-reinforced plastic composite material through Nd: YAG laser drilling and optimisation using stochastic methods. Journal of Mechanical Science and Technology, 38(8), 4321-4331 (2024)
131. Sahoo, A.K., Mishra, D.R.: Characterization of Laser Drilling and Parametric Optimization Using Golden Jackal Optimizer. International Journal of Precision Engineering and Manufacturing, 25(11), 2299-2310 (2024)
132. Nandi, S., Kuar, A.S.: Parametric optimisation of Nd:YAG laser micro-drilling of alumina using NSGA II. International Journal of Machining and Machinability of Materials, 17, 1-21 (2015)
133. Ding, H., Wang, Z., Guo, Y.: Multi-objective optimization of fiber laser cutting based on generalized regression neural network and non-dominated sorting genetic algorithm. Infrared Physics and Technology, 108, 103337 (2020)
134. Shrivastava, P.K., Pandey, A.K.: Geometrical quality evaluation in laser cutting of Inconel-718 sheet by using Taguchi based regression analysis and particle swarm optimization. Infrared Physics & Technology, 89, 369-380 (2018)
135. Shrivastava, P.K., Pandey, A.K.: Optimization of machining parameter during the laser cutting of Inconel-718 sheet using regression analysis based particle swarm optimization method. Materials Today: Proceedings, 5, 24167-24176 (2018)
136. Swain, A., Sahu, S.K., Ekka, A., Das, S.R.: Parametric optimization of Nd:YAG laser microgrooving of alumina ceramic using integrated RSM-PSO approach, In: Proc. of Emerging Trends in Mechanical Engineering : Select Proceedings of ICETME 2018. Springer Nature, Singapore, 1-10 (2019)
137. Pramanik, D., Roy, N., Kuar, A.S., Sarkar, S., Mitra, S.: Experimental investigation of sawing approach of low power fiber laser cutting of titanium alloy using particle swarm optimization technique. Optics and Laser Technology, 147, 107613 (2022)
138. Mishra, L., Mahapatra, T.R., Mishra, D., Pattanaik, S.K.: Machinability analysis and multiple performance optimization during laser micro-drilling of CNT reinforced polymer nanocomposite. Lasers in Manufacturing and Materials Processing, 9(2), 151-172 (2022)
139. Mishra, L., Mahapatra, T.R., Mishra, D., Parimanik, S.R.: Investigation of laser micro-drilling machinability and performance optimization of polymer nanocomposites reinforced with different carbon allotropes. Proc. of the Institution of Mechanical Engineers, Part E: Journal of Process Mechanical Engineering, 283, 1354-1371 (2024)
140. Mishra, L., Mahapatra, T.R., Mishra, D., Mishra, P., Padhi, P.C.: Study of Nd: YAG laser micro-drilling machinability and parametric optimization of graphite/Epoxy and carbon black/Epoxy

nanocomposites. Proceedings of the Institution of Mechanical Engineers, Part N: Journal of Nanomaterials, Nanoengineering and Nanosystems, 1-26 (2024) https://doi.org/10.1177/23977914231217921

141. Mishra, L., Mahapatra, T.R., Mishra, D.: Performance evaluation and sustainability assessment in laser micro-drilling of carbon nanotube-reinforced polymer matrix composite using MOORA and whale optimization algorithm. Process Integration and Optimization for Sustainability, 6, 603-620 (2022)
142. Mishra, L., Mahapatra, T.R., Mishra, D., Parimanik, S.R.: Machinability study, machining performance optimization and sustainability assessment in laser micro-drilling of CNT/epoxy nanocomposite, International Journal of Machining and Machinability of Materials 25, 111-139 (2023)
143. Mishra, L., Mahapatra, T.R., Mishra, D., Parimanik, S.R., Rout, A.K.: Parametric optimization of hole quality in Nd: YAG laser drilled carbon black/epoxy composite by grey based whale optimization technique. Materials Today: Proceedings (2024) https://doi.org/10.1016/j.matpr.2024.02.022
144. Madić, M., Radovanović, M.: Application of cuckoo search algorithm for surface roughness optimization in CO_2 laser cutting. Annals of Faculty Engineering Hunedoara - International Journal of Engineering, XI, 39-44 (2013)
145. Goswami, D., Chakraborty S.: A study on the optimization performance of fireworks and cuckoo search algorithms in laser machining processes. Journal of Institution of Engineers Series C, 96, 215-229 (2015)
146. Pramanik, D., Singh, T., Roy, N., Biswas, R., Kuar, A.S., Sarkar, S., Mitra, S.: Analysis of performance characteristics with artificial intelligence based TLBO technique for laser drilling of Monel superalloy. Optics and Laser Technology, 164, 109554 (2023)
147. Kar, T., Goswami, A.: Experimental investigation on fiber laser micro drilling of Titanium grade 5: fabrication, nature-inspired optimization and analysis through image processing. Engineering Research Express, 6, 025411 (2024)
148. Tamilarasan, A., Rajamani, D.: Metaheuristic prediction models for kerf deviation in Nd-YAG laser cutting of AlZnMgCu1. 5 alloy. Modelling, 6(1), 17 (2025)
149. Pendokhare, D., Chakraborty, S.: A comparative analysis of preying behavior-based metaheuristic algorithms for optimization of laser beam drilling processes. Decision Analytics Journal, 10, 100412 (2024)
150. Ganesan, D., Salunkhe, S., Panghal, D., Murali, A.P., Mahalingam, S., Tarigonda, H., Gawade, S.R., Hussein, H.M.A.-M.: Optimization of abrasive water jet machining process parameters on onyx composite followed by additive manufacturing. Processes, 11, 2263 (2023)
151. Zain, A.M., Haron, H., Sharif, S.: Optimization of process parameters in the abrasive waterjet machining using integrated SA-GA. Applied Soft Computing, 11(8), 5350-5359 (2011)

152. Rao, V.D.P., Mrudula, M., Geethika, V.N.: Multi-objective optimization of parameters in abrasive water jet machining of carbon-glass fibre-reinforced hybrid composites. Journal of The Institution of Engineers (India): Series D, 100(1), 55-66 (2019)
153. Kumar, K.N., Babu, P.D.: Improving the machining performance of polymer hybrid composite by abrasive water jet machining for precise machining. Arabian Journal for Science and Engineering, 49, 15347-15366 (2024)
154. Ghosh, T., Martinsen, K.: CFNN-PSO: an iterative predictive model for generic parametric design of machining processes. Applied Artificial Intelligence, 33(11), 951-978 (2019)
155. Gnanavelbabu, A., Saravanan, P.: Experimental investigations of abrasive waterjet machining parameters on titanium alloy Ti-6Al-4V using RSM and evolutionary computational techniques. In Advances in Unconventional Machining and Composites. Lecture Notes on Multidisciplinary Industrial Engineering. Springer, Singapore 413-425 (2019)
156. Shukla, R., Singh, D.: Experimentation investigation of abrasive water jet machining parameters using Taguchi and evolutionary optimization techniques. Swarm and Evolutionary Computation, 32, 167-183 (2017)
157. Nabavi, M.T., Moghaddam, M.A., Fardfarimani, N., Kolahan, F.: Comparative modeling of abrasive waterjet machining process based on OA-Taguchi and D-optimal approach and optimization using simulated annealing algorithm. Scientia Iranica, 29(3), 1276-1287 (2022)
158. Joel, C., Joel, L., Muthukumaran, S., Shanthini, P.M.: Parametric optimization of abrasive water jet machining of C360 brass using MOTLBO. Materials Today: Proceedings, 37, 1905-1910 (2021)
159. Sharma, V.S., Kumar, A., Gupta, M.K., Bhanot, N.: Process parameter optimization for abrasive water jet machining of titanium alloy using meta-heuristic algorithms. In: MATEC Web of Conferences, 221, 1-5 (2018)
160. Balaji, K., Kumar, M.S., Yuvaraj, N.: Multi objective taguchi-grey relational analysis and krill herd algorithm approaches to investigate the parametric optimization in abrasive water jet drilling of stainless steel. Applied Soft Computing, 102(12), 1-25 (2021)
161. Dharmalingam, G., Prasad, M.A., Panghal, D., Salunkhe, S., Kumar, M.S., Rathinasuriyan, C.: Optimization of AWJM process parameters on 3D-printed onyx-glass fiber hybrid composite. Journal of Harbin Institute of Technology, 30(2), 84-98 (2023)
162. Jagadish, Patel, C.G.M., Sibalija, T.V., Mumtaz, J., Li, Z.: Abrasive water jet machining for a high-quality green composite: The soft computing strategy for modeling and optimization. Journal of the Brazilian Society of Mechanical Sciences and Engineering, 44(83), 1-20 (2022)
163. Palanikumar, K., Prabhudass¹, J. M., Prabha, P. S., Senthilkumar, N.: Sustainable optimization of abrasive water jet machining for MWCNT/bamboo/Kenaf hybrid polymer composites. International Journal on Interactive Design and Manufacturing, 1-18 (2024)

164. Gulia, V., Nargundkar, A.: Optimization of process parameters of abrasive water jet machining using variations of cohort intelligence (CI). In Applications of Artificial Intelligence Techniques in Engineering. Advances in Intelligent Systems and Computing, Springer, Singapore, 2, 467-474 (2018)
165. Tamilarasan, A., Renugambal,A.: An integrated RSM-improved salp swarm algorithm for quality characteristics in AWJM of Ananas Comosus-HIPS composites. International Journal of Lightweight Materials and Manufacture, 6(3), 297-310 (2023)
166. Mokkandi, P., Jayamani, M., Sekarbab, B., Kadambarajan, J.P., Nagarajan, R., Sultan, M.T.H., Shanmugavel, R.: Machinability performance of Al-NiTi and Al-NiTi-nano SiC composites with parametric optimization using GSA. Journal of the Australian Ceramic Society, 53, 599-609 (2017)
167. Adapa, S.K., Jagadish, Y.S., Raju, S.S.: Optimization of AWJM process on processing of lite bamboo reinforced polymer composite by using grasshopper algorithm. In: Dikshit, M.K., Soni, A., Davim, J.P. (eds) Advances in Manufacturing Engineering. Lecture Notes in Mechanical Engineering. Springer, Singapore 55-64 (2022)
168. Tamilarasan, A., Renugambal, A., Shunmugesh, K.: Multi-performance optimization for AWJ drilling process in cutting of ceramic tile: BBD with EOBL-GOA algorithm. Multidiscipline Modeling in Materials and Structures, 19(6), 1199-1225 (2023)
169. Tamilarasan, A., Renugambal, A., Vijayan, D.: Parametric estimation for AWJ cutting of Ti-6Al-4V alloy using Rat swarm optimization algorithm. Materials and Manufacturing Processes, 37(16), 1871-1881 (2022)
170. Rajyalakshmi, G., Gupta, Y., Panigrahi, S., Ahsan, M., Manjunath, S.P., Jayakrishna, K.: Artificial immune system approach for optimizing abrasive water jet machining process on super alloys. International Journal on Interactive Design and Manufacturing, 17(2), 847-858 (2023)
171. Jagadish, Manjunath, P.G.: Modeling and optimization of AWJM process on the processing of Banana fiber-reinforced polymer composites using Taguchi-JAYA method. Soft Computing in Smart Manufacturing, 8, 253 (2021)
172. Tripathi, D.R., Vachhani, K.H., Bandhu, D., Kumari, S., Kumar, V.R., Abhishek, K.: Experimental investigation and optimization of abrasive waterjet machining parameters for GFRP composites using metaphor-less algorithms. Materials and Manufacturing Processes, 36(7), 803-813 (2021)
173. Chaouch, F., Ben Khalifa, A., Zitoune, R., Zidi, M.: Modeling and multi-objective optimization of abrasive water jet machining process of composite laminates using a hybrid approach based on neural networks and metaheuristic algorithm. Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture, 238(9), 1351-1361 (2024)
174. Vasudevan, B., Nagarajan, L., Natrayan, L., Karthick, A., Mahalingam, S.K., Prakash, C., Chan, C.K., Panchal, H., Siddiqui, M.I.H.: Experimental study, modeling, and parametric optimization

on abrasive waterjet drilling of YSZ-coated Inconel 718 superalloy. Journal of Materials Research and Technology, 29, 4662-4675 (2024)

175. Pendokhare, D., Nandi, S., Chakraborty, S.: A comparative analysis on parametric optimization of abrasive water jet machining processes using foraging behavior-based metaheuristic algorithms. OPSEARCH, 1-42 (2025) https://doi.org/10.1007/s12597-025-00916-y
176. Jamsari, N.S., Lim, Z.X., Lim, C.S., You, K.Y., Tan, M.L.P.: The optimization of cut surface for computer numerical control plasma cutting machine. PaperASIA, 40(4b), 157-164 (2024)
177. Rao, R.V., Rai, D.P., Ramkumar, J., Balic, J.: A new multi-objective Jaya algorithm for optimization of modern machining processes. Advances in Production Engineering & Management, 11(4), 1-16 (2016)
178. Masoudi, S., Mirabdolahi, M., Dayyani, M., Jafarian, F., Vafadar, A., Dorali, M.R.: Development of an intelligent model to optimize heat-affected zone, kerf, and roughness in 309 stainless steel plasma cutting by using experimental results. Materials and Manufacturing Processes, 34(3), 345-356 (2019)
179. Hamdy, A., Fattouh, M., Abaas, S., Masoud, R.: Multi-objective optimization of plasma arc cutting process using MOORA combined with GA. Engineering Research Journal, 42(3), 218-230 (2019)
180. Dash, N., Roy, A.K., Debta, S., Kumar, K.: Optimization of process parameters in plasma arc cutting applying genetic algorithm and fuzzy logic. In: Soft Computing Techniques and Applications in Mechanical Engineering, IGI Global, 123-139 (2018)
181. Peko, I., Marić, D., Nedić, B., Samardžić, I.: Modeling and optimization of cut quality responses in plasma jet cutting of aluminium alloy EN AW-5083. Materials, 14, 5559 (2021)
182. Melaku, N.S., Bogale, T.M.: Parameters optimization in plasma arc cutting of AISI 1020 mild steel plate using hybrid genetic algorithm and artificial neural network. International Journal for Simulation and Multidisciplinary Design Optimization, 14, 20 (2023)
183. Rajamani, D., Siva Kumar, M., Balasubramanian, E.: Multi-response optimization of plasma arc cutting on Monel 400 alloy through whale optimization algorithm. In: Handbook of Whale Optimization Algorithm, Academic Press, 373-386 (2024)
184. Karthick, M., Anand, P., Meikandan, M., Siva Kumar, M.: Machining performance of Inconel 718 using WOA in PAC. Materials and Manufacturing Processes, 36(11), 1274-1284 (2021)
185. Patel, P., Nakum, B., Abhishek, K., Kumar, V. R., Kumar, A.: Optimization of surface roughness in plasma arc cutting of AISID2 steel using TLBO. Materials Today: Proceedings, 5(9), 18927-18932 (2018)
186. Patel, P., Nakum, B., Abhishek, K., Rakesh Kumar, V.: Machining performance optimization during plasma arc cutting of AISI D2 steel: application of FIS, nonlinear regression and JAYA optimization algorithm. Journal of the Brazilian Society of Mechanical Sciences and Engineering, 40, 1-15 (2018)

187. Das, M.K., Barman, T.K., Sahoo, P. and Kumar, K., 2017. Process optimization in non-conventional processes: Experimentation with plasma arc cutting. In: Handbook of Research on Manufacturing Process Modeling and Optimization Strategies, IGI Global, 82-119 (2017)
188. Karthick, M., Anand, P., Siva Kumar, M., Meikandan, M.: Exploration of MFOA in PAC parameters on machining Inconel 718. Materials and Manufacturing Processes, 37(12), 1433-1445 (2022)
189. Tamilarasan, A., Rajmohan, T., Arumugam, S., Arunpremnath, A., Mohan, K., Manohar, P.: Application of water cycle algorithm for optimizing the PAC process parameters in cutting Ti-6Al-4V Alloy. In: Advances in Materials and Manufacturing Engineering, Springer, 7, 389-396 (2021)
190. Pendokhare, D., Chakraborty, S.: Optimizing plasma arc cutting processes using physics-based metaheuristic algorithms: a comparative analysis. International Journal on Interactive Design and Manufacturing, 1-35 (2024) https://doi.org/10.1007/s12008-024-02136-y
191. Wolpert, D.H., Macready, W.G.: No free lunch theorems for optimization. IEEE Transactions on Evolutionary Computation, 1, 67-82 (1997)
192. Rai, R., Das, A., Ray, S., Dhal, K.G.: Human-inspired optimization algorithms: Theoretical foundations, algorithms, open-research issues and application for multi-level thresholding. Archives of Computational Methods in Engineering, 29, 5313-5352 (2022)
193. Rao, R.V., Savsani, V.J., Vakharia, D.P.: Teaching-learning-based optimization: a novel method for constrained mechanical design optimization problems. Computer-Aided Design, 43, 303-315 (2011)
194. Shabani, A., Asgarian, B., Gharebaghi, S.A., Salido, M.A., Giret, A.: A new optimization algorithm based on search and rescue operations. Mathematical Problems in Engineering, 2019(1), 2482543 (2019)
195. Lyu, M., Zhao, Y., Huang, C., Huang, H.: Unmanned aerial vehicles for search and rescue: A survey. Remote Sensing, 15(13), 3266 (2023)
196. Dehghani, M., Trojovský, P.: Teamwork optimization algorithm: A new optimization approach for function minimization/maximization. Sensors, 21(13), 4567 (2021)
197. Acharya, D., Das, D.K.: A novel human conception optimizer for solving optimization problems. Scientific Reports, 12, 21631 (2022)
198. Zhang, J., Xiao, M., Gao, L., Pan, Q.: Queuing search algorithm: A novel metaheuristic algorithm for solving engineering optimization problems. Applied Mathematical Modelling, 63, 464-490 (2018)
199. Askarzadeh, A.: Bird mating optimizer: an optimization algorithm inspired by bird mating strategies. Communications in Nonlinear Science and Numerical Simulation, 19, 1213-1228 (2014)

200. Galán, S.F., Mengshoel, O.J., Pinter, R.: A novel mating approach for genetic algorithms. Evolutionary Computation, 21, 197-229 (2013)
201. Ritthipakdee, A., Thammano, A., Premasathian, N., Jitkongchuen, D.: Firefly mating algorithm for continuous optimization problems. Computational Intelligence and Neuroscience, Article ID 8034573, https://doi.org/10.1155/2017/8034573 (2017)
202. Yang, X.S.: Firefly algorithms for multimodal optimization. In: Proc. of International Symposium on Stochastic Algorithms, Heidelberg, Berlin, 169-178 (2009)
203. Yang, X.S., He, X.: Firefly algorithm: recent advances and applications. International Journal of Swarm Intelligence, 1, 36-50 (2013)
204. Qi, X., Zhu, S., Zhang, H.: A hybrid firefly algorithm. In: Proc. of IEEE Advanced Information Technology, Electronic and Automation Control Conference, China, 287-291, (2017)
205. Salgotra, R., Singh, U.: The naked mole-rat algorithm. Neural Computing and Applications, 31, 8837-8857 (2019)
206. Guha, S., Das, A., Singh, P.K., Ahmadian, A., Senu, N., Sarkar, R.: Hybrid feature selection method based on harmony search and naked mole-rat algorithms for spoken language identification from audio signals. IEEE Access, 8, 182868-182887 (2020)
207. Jada, C., Vadathya, A.K., Shaik, A., Charugundla, S., Ravula, P.R., Rachavarapu, K.K.: Butterfly mating optimization. In: Intelligent Systems Technologies and Applications, Springer International Publishing, 1, 3-15 (2016)
208. Assiri, A.S.: On the performance improvement of butterfly optimization approaches for global optimization and feature selection. Plos One, 16, e0242612 (2021)
209. Makhadmeh, S.N., Al-Betar, M.A., Abasi, A.K., Awadallah, M.A., Doush, I.A., Alyasseri, Z. A.A., Alomari, O.A.: Recent advances in butterfly optimization algorithm, its versions and applications. Archives of Computational Methods in Engineering, 30, 1399-1420 (2023)
210. Sulaiman, M.H., Mustaffa, Z., Saari, M.M., Daniyal, H., Daud, M.R., Razali, S., Mohamed, A.I.: Barnacles mating optimizer: a bio-inspired algorithm for solving optimization problems. In: Proc. of 19th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, Korea, 87, 265-270 (2018)
211. Luo, S., Wang, W., Fang, M., Xu, W.: Self-encoding Barnacle mating optimizer algorithm for manpower scheduling in flow shop. arXiv preprint arXiv:2111.08246 (2021)
212. Mustaffa, Z., Sulaiman, M.H.: Stock price predictive analysis: An application of hybrid Barnacles mating optimizer with artificial neural network. International Journal of Cognitive Computing in Engineering, 4, 109-117 (2023)
213. Jia, H., Sun, K.: Improved barnacles mating optimizer algorithm for feature selection and support vector machine optimization. Pattern Analysis and Applications, 24, 1249-1274 (2021)
214. Mohapatra, S., Mohapatra, P.: American zebra optimization algorithm for global optimization problems. Scientific Reports, 13, 5211 (2023)

215. Moshtaghi, H.R., Eshlaghy, A.T., Motadel, M.R.: A comprehensive review on meta-heuristic algorithms and their classification with novel approach. Journal of Applied Research on Industrial Engineering, 8, 63-89 (2021)
216. Kumar, V., Yadav, S.M.: A state-of-the-art review of heuristic and metaheuristic optimization techniques for the management of water resources. Water Supply, 22, 3702-3728 (2022)
217. Shadravan, S., Naji, H.R., Bardsiri, V.K.: The sailfish optimizer: A novel nature-inspired metaheuristic algorithm for solving constrained engineering optimization problems. Engineering Applications of Artificial Intelligence, 80, 20-34 (2019)
218. Zhang, Y., Mo, Y.: Chaotic adaptive sailfish optimizer with genetic characteristics for global optimization. The Journal of Supercomputing, 78, 10950-10996 (2022)
219. Zhang, Y., Mo, Y.: Dynamic optimization of chemical processes based on modified sailfish optimizer combined with an equal division method. Processes, 9, 1806 (2021)
220. Ghosh, K.K., Ahmed, S., Singh, P.K., Geem, Z.W., Sarkar, R.: Improved binary sailfish optimizer based on adaptive β-hill climbing for feature selection. IEEE Access, 8, 83548-83560 (2020)
221. Ikram, R.M.A., Dehrashid, A.A., Zhang, B., Chen, Z., Le, B.N., Moayedi, H.: A novel swarm intelligence: cuckoo optimization algorithm (COA) and sailfish optimizer (SFO) in landslide susceptibility assessment. Stochastic Environmental Research and Risk Assessment, 37, 1717-1743 (2023)
222. Heidari, A.A., Mirjalili, S., Faris, H., Aljarah, I., Mafarja, M., Chen, H.: Harris Hawks optimization: Algorithm and applications. Future Generation Computer Systems, 97, 849-872 (2019)
223. Hussien, A.G., Abualigah, L., Abu Zitar, R., Hashim, F.A., Amin, M., Saber, A., Gandomi, A.H.: Recent advances in Harris Hawks optimization: A comparative study and applications. Electronics, 11, 1919 (2022)
224. Tian, F., Wang, J., Chu, F.: Improved multi-strategy Harris Hawks optimization and its application in engineering problems. Mathematics, 11, 1525 (2023)
225. Tripathy, B.K., Reddy Maddikunta, P.K., Pham, Q.V., Gadekallu, T.R., Dev, K., Pandya, S., ElHalawany, B.M.: Harris Hawk optimization: a survey on variants and applications. Computational Intelligence and Neuroscience, 2218594, 1-20 (2022)
226. Yu, J., Kim, C.H., Rhee, S.B.: The comparison of lately proposed Harris Hawks optimization and Jaya optimization in solving directional overcurrent relays coordination problem. Complexity, 3807653, 1-22 (2020)
227. Vishwakarma, R., Kesarwani, S., Verma, R.K., Debnath, K., Davim, J.P.: Using Harris Hawk algorithm for experimental study on the hole dilation mechanism during micro-machining (μM) of graphene nanoplatelets/carbon fiber (GnP/C) reinforced polymeric composite. Materials Research Express, 10, 024005 (2023)

228. Abualigah, L., Yousri, D., Abd Elaziz, M., Ewees, A.A., Al-Qaness, M.A., Gandomi, A.H.: Aquila optimizer: a novel meta-heuristic optimization algorithm. Computers & Industrial Engineering, 157, 107250 (2021)
229. Akyol, S.: A new hybrid method based on Aquila optimizer and tangent search algorithm for global optimization. Journal of Ambient Intelligence and Humanized Computing, 14, 8045-8065 (2023)
230. Li, Z., Wang, Q., Zhu, B., Wang, B., Zhu, W., Dai, Y.: Thermal error modeling of high-speed electric spindle based on Aquila optimizer optimized least squares support vector machine. Case Studies in Thermal Engineering, 39, 102432 (2022)
231. Li, X., Ma, Y., Li, Y., Li, H., Zuo, H., Sun, W.: Improved Aquila optimizer optimization algorithm based on multi-strategy fusion. In: Proc. of 2nd International Conference on Computer Science, Electronic Information Engineering and Intelligent Control Technology, China, 757-760 (2022)
232. Gul, F., Mir, I., Mir, S.: Aquila optimizer with parallel computing strategy for efficient environment exploration. Journal of Ambient Intelligence and Humanized Computing, 14, 1-16 (2023)
233. Yang, X.S.: A new metaheuristic bat-inspired algorithm. In: J.R. González, D.A. Pelta, C. Cruz, G. Terrazas, N. Krasnogor N (eds) Nature Inspired Cooperative Strategies for Optimization, Springer, Berlin, 284, 65-74 (2010)
234. Savković, M.M., Bulatović, R.R., Gašić, M.M., Pavlović, G.V., Stepanović, A.Z.: Optimization of the box section of the main girder of the single-girder bridge crane by applying biologically inspired algorithms. Engineering Structures, 148, 452-465 (2017)
235. Yang, X.S., He, X.: Bat algorithm: literature review and applications. International Journal of Bio-inspired Computation, 5, 141-149 (2013)
236. Dash, P., Saikia, L.C., Sinha, N.: Automatic generation control of multi area thermal system using bat algorithm optimized PD-PID cascade controller. International Journal of Electrical Power & Energy Systems, 68, 364-372 (2015)
237. Guo, S.S., Wang, J.S., Ma, X.X.: Improved bat algorithm based on multipopulation strategy of island model for solving global function optimization problem. Computational Intelligence and Neuroscience, Article ID 6068743 (2019) https://doi.org/10.1155/2019/6068743.
238. Mirjalili, S., Mirjalili, S.M., Lewis, A.: Grey wolf optimizer. Advances in Engineering Software, 69, 46-61 (2014)
239. Kalita, K., Shinde, D., Chakraborty, S.: Grey wolf optimizer-based design of ventilated brake disc. Journal of the Brazilian Society of Mechanical Sciences and Engineering, 43, 1-15 (2021)
240. Kalita, K., Pal, S., Haldar, S., Chakraborty, S.: A hybrid TOPSIS-PR-GWO approach for multi-objective process parameter optimization. Process Integration and Optimization for Sustainability, 6, 1011-1026 (2022)

241. Hou, Y., Gao, H., Wang, Z., Du, C.: Improved grey wolf optimization algorithm and application. Sensors, 22, 3810 (2022)
242. Singh, N.: A modified variant of grey wolf optimizer. Scientia Iranica, 27, 1450-1466 (2020)
243. Mirjalili, S.: Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. Neural Computing & Applications, 27, 1053-1073 (2016)
244. Parmaksiz, H., Yuzgec, U., Dokur, E., Erdogan, N.: Mutation based improved dragonfly optimization algorithm for a neuro-fuzzy system in short term wind speed forecasting. Knowledge-based Systems, 268, 1-32 (2023)
245. Abdollahzadeh, B., Gharehchopogh, F.S., Mirjalili, S.: African vultures optimization algorithm: A new nature-inspired metaheuristic algorithm for global optimization problems. Computers & Industrial Engineering, 158(C), 1-37 (2021)
246. Fan, J., Li, Y., Wang, T.: An improved African vultures optimization algorithm based on tent chaotic mapping and time-varying mechanism. PLoS One, 16(11), 1-28 (2021)
247. Salah, B., Hasanien, H.M., Ghali, F.M., Alsayed, Y.M., Abdel-Aleem, S.H., El-Shahat, A.: African vulture optimization-based optimal control strategy for voltage control of islanded DC microgrids. Sustainability, 14(19), 1-26 (2022)
248. Saremi, S., Mirjalili, S., Lewis, A.: Grasshopper optimization algorithm: theory and application. Advances in Engineering Software, 105, 30-47 (2017)
249. Wu, L., Wu, J., Wang, T.: Enhancing grasshopper optimization algorithm (GOA) with levy flight for engineering applications. Scientific Reports, 13(1), 1-49 (2023)
250. Qin, P., Hu, H., Yang, Z.: The improved grasshopper optimization algorithm and its applications. Scientific Reports, 11(1), 1-14 (2021)
251. Pan, W.T.: A new fruit fly optimization algorithm: taking the financial distress model as an example. Knowledge-Based Systems, 26, 69-74 (2012)
252. Li, Y., Han, M.: Improved fruit fly algorithm on structural optimization. Brain Informatics, 7(1), 1-13 (2020)
253. Lv, S.X., Zeng, Y.R., Wang, L.: An effective fruit fly optimization algorithm with hybrid information exchange and its applications. International Journal of Machine Learning and Cybernetics, 9, 1623-1648 (2018)
254. Meng, X.B., Gao, X.Z., Lu, L., Liu, Y., Zhang, H.: A new bioinspired optimization algorithm: bird swarm algorithm. Journal of Experimental & Theoretical Artificial Intelligence, 28(4), 1-15 (2015)
255. Lin, M., Zhong, Y., Lin, J., Lin, X.: Discrete bird swarm algorithm based on information entropy matrix for traveling salesman problem. Mathematical Problems in Engineering, 2018, 1-15 (2018)

256. Sridharan, S., Subramanian, R.K., Srirangan, A.K.: Physics based meta heuristics in manufacturing. Materials Today: Proceedings, 39, 805-811 (2021)
257. Hashim, F.A., Hussain, K., Houssein, E.H., Mabrouk, M.S., Al-Atabany, W.: Archimedes optimization algorithm: a new metaheuristic algorithm for solving optimization problems. Applied Intelligence, 51, 1531-1551 (2021)
258. Jiang, S.J., Chu, S.C., Zou, F.M., Shan, J., Zheng, S.G., Pan, J.S.: A parallel Archimedes optimization algorithm based on Taguchi method for application in the control of variable pitch wind turbine. Mathematics and Computers in Simulation, 203, 306-327 (2023)
259. Varol Altay, E.: Hybrid Archimedes optimization algorithm enhanced with mutualism scheme for global optimization problems. Artificial Intelligence Review, 56(7), 6885-6946 (2023)
260. Zhao, W., Wang, L., Zhang, Z.: Atom search optimization and its application to solve a hydrogeologic parameter estimation problem. Knowledge-Based Systems, 163, 283-304 (2019)
261. Kumar, R., Sikander, A.: A new order abatement method based on Atom search optimization. International Journal of Dynamics and Control, 11(4), 1704-1717 (2023)
262. Zhang, C., Hua, L., Ji, C., Nazir, M.S., Peng, T.: An evolutionary robust solar radiation prediction model based on WT-CEEMDAN and IASO-optimized outlier robust extreme learning machine. Applied Energy, 322, 119518 (2022)
263. Izci, D., Ekinci, S., Hussien, A.G.: Effective PID controller design using a novel hybrid algorithm for high order systems. PLoS One, 18(5), 0286060 (2023)
264. Wei, Z., Huang, C., Wang, X., Han, T., Li, Y.: Nuclear reaction optimization: A novel and powerful physics-based algorithm for global optimization. IEEE Access, 7, 66084-66109 (2019)
265. Abedinpourshotorban, H., Shamsuddin, S.M., Beheshti, Z., Jawawi, D.N.: Electromagnetic field optimization: a physics-inspired metaheuristic optimization algorithm. Swarm and Evolutionary Computation, 26, 8-22 (2016)
266. Ibrahim, A.M., Tawhid, M.A.: Chaotic electromagnetic field optimization. Artificial Intelligence Review, 56(9), 9989-10030 (2023)
267. Bouchekara, H.: Solution of the optimal power flow problem considering security constraints using an improved chaotic electromagnetic field optimization algorithm. Neural Computing and Applications, 32(7), 2683-2703 (2020)
268. Aranguren, I., Valdivia, A., Pérez-Cisneros, M., Oliva, D., Osuna-Enciso, V.: Digital image thresholding by using a lateral inhibition 2D histogram and a mutated electromagnetic field optimization. Multimedia Tools and Applications, 81(7), 10023-10049 (2022)
269. Rashedi, E., Nezamabadi-Pour, H., Saryazdi, S.: GSA: a gravitational search algorithm. Information Sciences, 179(13), 2232-2248 (2009)
270. Rashedi, E., Rashedi, E., Nezamabadi-Pour, H.: A comprehensive survey on gravitational search algorithm. Swarm and Evolutionary Computation, 41, 141-158 (2018)

271. Camposeco-Negrete, C.: Optimization of cutting parameters using response surface method for minimizing energy consumption and maximizing cutting quality in turning of AISI 6061 T6 aluminum. Journal of Cleaner Production, 91, 109-117 (2015)
272. Ragavendran, U., Ghadai, R.K., Bhoi, A.K., Ramachandran, M., Kalita, K.: Sensitivity analysis and optimization of EDM process parameters. Transactions of the Canadian Society of Mechanical Engineering, 43, 1-27 (2019)
273. CoelloCoello, C.A., Lamont, G.B., Van Veldhuizen, D.A.: Evolutionary algorithms for solving multi-objective problems. Springer, New York (2007)
274. Friedman, M.: A comparison of alternative tests of significance for the problem of m rankings. Annals of Mathematical Statistics, 11, 86-92 (1940)
275. Wilcoxon, F., Katti, S.K., Wilcox, R.A.: Critical values and probability levels for the Wilcoxon rank sum test and the Wilcoxon signed rank test. Selected Tables in Mathematical Statistics, 1, 171-259 (1970)
276. Tuo, S., Yong, L., Deng, F. A., Li, Y., Lin, Y., Lu, Q.: HSTLBO: A hybrid algorithm based on harmony search and teaching-learning-based optimization for complex high-dimensional optimization problems. PloS one, 12, e0175114 (2017)
277. Saraswat, A., Saini, A.: A novel hybrid fuzzy multi-objective evolutionary algorithm: HFMOEA. In: Advances in Computer Science and Information Technology. Proceedings, Part III, 2, 168-177 (2012)
278. Lim, W.H., Isa, N.A.M.: Teaching and peer-learning particle swarm optimization. Applied Soft Computing, 18, 39-58 (2014)
279. Nguyen, T.T., Mia, M., Dang, X.P., Le, C.H., Packianather, M.S.: Green machining for the dry milling process of stainless steel 304. Proceedings of the Institute of Mechanical Engineers, Part B: Journal of Engineering Manufacture, 234, 881-899 (2020)
280. Abdulwahab, A.E., Hubeatir, K.A., Imhan, K.I.: Optimization of PC micro-drilling using a continuous CO_2 laser: an experimental and theoretical comparative study. Journal of Engineering and Applied Science, 69, 98 (2022)
281. Balamurugan, K., Uthayakumar, M., Sankar, S., Hareesh, U.S., Warrier, K.G.K.: Predicting correlations in abrasive waterjet cutting parameters of Lanthanum phosphate/Yttria composite by response surface methodology. Measurement, 131, 309-318 (2019)
282. Kalita, K., Chakraborty, S.: An efficient approach for metaheuristic-based optimization of composite laminates using genetic programming. International Journal on Interactive Design and Manufacturing, 17, 899-916 (2023)
283. Shihab, S.K., Gattmah, J., Kadhim, H.M.: Experimental investigation of surface integrity and multi-objective optimization of end milling for hybrid Al7075 matrix composites. Silicon, 13(5), 1403-1419 (2021)

Dr. Shankar Chakraborty
Professor
Department of Production Engg.
Jadavpur University
Kolkata – 700 032

21/07/25

J. Inst. Eng. India Ser. C
https://doi.org/10.1007/s40032-025-01185-w

ORIGINAL CONTRIBUTION

Human-Inspired Metaheuristics-Based Optimization of Turning of AISI 6061-T6 Aluminium

Devendra Pendokhare[1] · **Shankar Chakraborty**[1]

Received: 4 December 2024 / Accepted: 4 April 2025

Abstract Turning is one of the vital machining processes widely employed in most of the manufacturing industries for removal of excess material from the exterior surface of any of the cylindrical workpieces. In this process, achieving optimal combination of various input parameters is essential for improved product quality, extended tool life and higher machining efficiency. However, optimization of a turning process is truly challenging due to involvement of multiple variables, and complex interactions between the input parameters and responses. Effective optimization should also consider resources, like energy, tools, production time and costs, as they significantly influence environmental impact, process sustainability and profitability. Based on the past data, this paper explores use of five human-inspired metaheuristic algorithms, i.e. teaching learning-based optimization (TLBO), search and rescue optimization (SAR), teamwork optimization algorithm (TOA), human conception optimizer (HCO) and queuing search algorithm (QSA) for optimization of turning of AISI 6061-T6 aluminium. Their performance is assessed based upon quality of the deduced solutions and computational effort. The Pareto optimal front is developed to search out the optimal parametric intermix for the multi-objective optimization problem. Among the considered algorithms, TLBO proves to be most effective in achieving the optimal parameter combination, improving material removal rate, surface roughness and specific cutting energy consumption by 27.08, 36.40 and 31.16% for single-objective optimization; and 15.03, 26.63 and 23.27% for multi-objective optimization, respectively, against the previous study. Values of spacing and hypervolume (two quality measures), and results of Friedman's mean rank test and Wilcoxon rank-sum test also corroborate its superiority over the other competitors.

Keywords Turning · Optimization · Human-based metaheuristics · Non-parametric test · Pareto front

Introduction

In the manufacturing industries, traditional machining operations, like turning, shaping and drilling are primarily employed for removal of undesirable material from a workpiece to furnish the final shape along with the required accuracy, tolerance and surface quality. Shape, operational cost, and mechanical and tribological properties of the work material all play decisive roles in choosing the most appropriate machining operation for a given application [1]. Turning is a fundamental operation widely utilized in the manufacturing industries for removing material from a revolving cylindrical workpiece while the cutting tool moves radially, axially or both, to achieve the desired shape geometry. It can be executed either in a single cut or multiple cuts. Turning is usually performed on a lathe, which supplies the necessary power for rotation of the workpiece at a specific speed while feeding the cutting tool at a defined rate and depth of cut (DOC). This process has become extremely popular in automotive, aerospace and other industries mainly due to its versatility and cost-effectiveness.

It has been experienced that for any of the machining processes, choice of the appropriate input parameters is extremely crucial for achieving high quality products at low cost [2]. Like other machining processes, in turning operations, several input parameters, like spindle speed

✉ Shankar Chakraborty
s_chakraborty00@yahoo.co.in

[1] Department of Production Engineering, Jadavpur University, Kolkata, West Bengal, India

Published online: 05 May 2025

International Journal on Interactive Design and Manufacturing (IJIDeM)
https://doi.org/10.1007/s12008-024-02136-y

ORIGINAL ARTICLE

Optimizing plasma arc cutting processes using physics-based metaheuristic algorithms: a comparative analysis

Devendra Pendokhare[1,2] · **Shankar Chakraborty**[1]

Received: 8 August 2024 / Accepted: 1 October 2024

Abstract
Plasma arc cutting (PAC) has become a flexible and effective method for precisely cutting complex profiles on various difficult-to-machine materials, including superalloys and composites, due to its many benefits, like higher dimensional accuracy, productivity and ability to cut thicker materials. Process optimization is important for obtaining high-quality cuts, reducing material wastage and increasing overall productivity. However, it is difficult to optimize this process because of involvement of many variables, intricate cutting mechanism, and interaction between the process parameters and responses. In this paper, applications of five newly developed physics-based metaheuristic algorithms, i.e. Archimedes optimization algorithm (AOA), atom search optimization (ASO), nuclear reaction optimization (NRO), electromagnetic field optimization (EFO) and gravitational search algorithm (GSA) are proposed for optimizing two PAC processes. Their optimization performance is compared in terms of computing effort, convergence time and solution quality. To find out the best parametric intermixes for resolving the multi-objective optimization problems, an effort is also put forward to develop the corresponding Pareto optimal fronts. For both the examples, compared to its competitors, EFO appears as the most effective metaheuristic for achieving the best combinations of the relevant process parameters. For the first example, EFO achieves 42.50, 35.85 and 25.55% improvements for single-objective optimization; and 35.71, 18.19 and 9% improvements for multi-objective optimization, in material removal rate, kerf taper and heat affected zone, respectively against the observations of the past researchers. In case of the second example, these improvements are 44.2, 26.42 and 17.12% for single-objective optimization; and 22.1, 10.36 and 6.21% for multi-objective optimization in surface roughness, kerf width and microhardness, respectively. With respect to average computation time, for multi-objective optimization, EFO saves 77.1, 36.3, 65.6 and 98.73% (for example 1); and 49.7, 33.12, 50.3 and 142.01% (for example 2) of the runtime against AOA, ASO, NRO and GSA, respectively. Results of two quality metrics (spacing and hypervolume) and two non-parametric statistical tests (Wilcoxon rank-sum test and Friedman's mean rank test) also prove superiority of EFO against the other physics-based algorithms under consideration. Thus, the primary objective of this paper is to explore application of five physics-based algorithms, specially EFO in deriving the optimal mixtures of two PAC processes resulting in their superior cutting efficiency, along with higher productivity and surface quality.

Keywords Plasma arc cutting · Optimization · Metaheuristics · Physics-based algorithm · Parameter · Response

Abbreviations

ABC	Artificial bee colony
AC	Arc current
ANFIS	Adaptive neuro-fuzzy inference system
ANN	Artificial neural network
AOA	Archimedes optimization algorithm
ASO	Atom search optimization
BBD	Box-Behnken design
CCD	Central composite design
CS	Cutting speed
DFR	Dross formation rate
EDAS	Evaluation based on distance from average solution
EFO	Electromagnetic field optimization

✉ Shankar Chakraborty
s_chakraborty00@yahoo.co.in

[1] Department of Production Engineering, Jadavpur University, Kolkata, West Bengal, India

[2] Department of Mechanical Engineering, Government Polytechnic, Thane, Maharashtra, India

Published online: 21 October 2024

Decision Analytics Journal 10 (2024) 100412

Contents lists available at ScienceDirect

Decision Analytics Journal

journal homepage: www.elsevier.com/locate/dajour

A comparative analysis of preying behavior-based metaheuristic algorithms for optimization of laser beam drilling processes

Devendra Pendokhare [a,b], Shankar Chakraborty [a,*]

[a] Department of Production Engineering, Jadavpur University, Kolkata, West Bengal, India
[b] Department of Mechanical Engineering, Government Polytechnic, Thane, Maharashtra, India

ARTICLE INFO

Keywords:
Optimization
Laser beam drilling
Sailfish optimizer
Harris hawks optimizer
Aquila optimizer
Bat optimizer
Grey wolf optimizer

ABSTRACT

Increasing demand for miniaturized components as micro-holes with diameters ranging from a few microns to hundreds of microns used in automobile, aviation, electronics, and medical industries makes laser beam drilling (LBD) a viable option in today's manufacturing environment. The LBD processes must be optimized for the best settings of their different input parameters to achieve enhanced performance and the most desired micro-hole characteristics. The earlier researchers have already attempted numerous metaheuristic algorithms. In this paper, the applicability of five robust preying behavior-based metaheuristic algorithms, i.e., sailfish optimizer, Harris hawks optimizer, Aquila optimizer, bat optimizer, and grey wolf optimizer, is validated while optimizing two LBD operations. The first example considers CO_2 laser-based micro-drilling on polycarbonate using laser power, exposure time, and focal plane position as the input variables, and hole depth, hole diameter, and heat-affected zone as the outputs. The second example deals with micro-drilling on stainless steel with gas pressure, current, cutting speed, and pulse frequency as the process parameters, and circularity at entry and exit, and hole taper angle as the responses. Although all the considered preying behavior-based metaheuristics are very robust and have excellent search ability, there has been a lack of comparative analysis of their relative performance while solving complex optimization problems with minimum computational effort. Their optimization performance is compared with respect to quality of the solutions, convergence speed and computational effort, identifying grey wolf optimizer as the most efficient one. The results of Friedman's mean rank test also prove the superiority of grey wolf optimizer over its competitors.

1. Introduction

With increasing geometrical complexity and miniaturization of the modern-day engineering components in many of the technologically advanced industries, generation of micro-holes in varying conductive and non-conductive materials has become a challenging task [1]. Among the available conventional and non-conventional machining processes, laser beam drilling (LBD) has emerged out as a viable option to generate micro-holes and complicated shape geometries in almost all types of difficult-to-cut materials, including metals, non-metals, composites and ceramics. In this process, a narrow laser beam having high intensity and very small pulse duration is focused to ablate material from the workpiece surface generating tiny holes with higher aspect ratio (>5:1), minimum heat affected zone (HAZ) and almost zero taper geometry. Thus, the material removal mechanism of LBD process is absorption of laser energy through a series of pulses at the same spot causing localized melting and vaporization. During LBD operation, a highly pressurized assist gas, i.e. air, oxygen or nitrogen is utilized to blow away the debris from the machining zone. Although it has several advantageous features, like no tool wear, higher precision and flexibility, excellent surface finish, higher repeatability, no residual stress and burr formation, no chatter and vibration, but it still suffers from other disadvantages, like low material removal rate (MRR), higher initial cost, requirement of skilled manpower etc. [2,3]. Micro-holes generated by this process have found immense applications in electronics (panel display boards, printed circuit boards, micro-electro-mechanical systems, solar cells etc.), medical industry for generation of micro-channels, packaging, manufacturing of ignition target balls, micro-heat exchanger, cooling holes in aero-engine turbine blades etc.

Among various laser sources, pulsed Nd:YAG laser and CO_2 laser, which are solid and gas lasers respectively, have become quite popular for the above-mentioned industrial applications. These laser sources facilitate micro-drilling of thick materials due to high incident peak power during the pulsating mode. It has been experimented that Nd:YAG laser performs better than CO_2 laser with respect to higher production rate, and generation of geometrical features having excellent dimensional accuracy and tolerance due to its higher energy density

* Corresponding author.
E-mail address: s_chakraborty00@yahoo.co.in (S. Chakraborty).

https://doi.org/10.1016/j.dajour.2024.100412
Received 11 October 2023; Received in revised form 30 January 2024; Accepted 1 February 2024
Available online 5 February 2024

OPSEARCH
https://doi.org/10.1007/s12597-025-00916-y

APPLICATION ARTICLE

A comparative analysis on parametric optimization of abrasive water jet machining processes using foraging behavior-based metaheuristic algorithms

Devendra Pendokhare[1] · Somnath Nandi[1] · Shankar Chakraborty[1]

Accepted: 14 January 2025

Abstract

Due to hybridization of the material removal mechanisms of abrasive jet and water jet machining processes, abrasive water jet machining (AWJM) appears as an efficient non-traditional process providing higher productivity, superior surface quality and excellent dimensional accuracy of almost all types of work materials irrespective of their mechanical and thermal properties. Optimization of this hybrid machining process is a challenging task because of involvement of multiple input parameters, contradictory responses and possible interactions between them. In this paper, based on past experimental datasets, two AWJM processes are optimized using five foraging behavior-based metaheuristic algorithms, i.e. dragonfly optimizer, African vultures optimizer, grasshopper optimizer, fruit fly optimizer and bird swarm optimizer, and their optimization performance is compared with respect to solution accuracy and variability, and computational effort. For both the processes, African vultures optimizer emerges out as the most efficient metaheuristic. For the first example (AWJM of Lanthanum phosphate/Yttria composites), it provides 71.47, 43.73 and 10% improvements for single-objective optimization; and 68.37, 43.45 and 9.53% improvements for multi-objective optimization, in material removal rate, kerf angle and surface roughness, respectively against the observations of the past researchers. On the other hand, in case of the second example (AWJM of glass fibre-reinforced polymer composites), the corresponding improvements are 41.62, 22.35 and 27.90% for single-objective optimization; and 23.66, 9.09 and 28.57% for multi-objective optimization in surface roughness, kerf taper and delamination length, respectively. Based on the Friedman's mean rank test, it can also be noticed that African vultures optimizer supersedes the remaining foraging behavior-based metaheuristic algorithms.

Keywords Optimization · AWJM · Metaheuristics · Foraging · Response

✉ Shankar Chakraborty
s_chakraborty00@yahoo.co.in

[1] Department of Production Engineering, Jadavpur University, Kolkata, West Bengal, India

Published online: 01 February 2025

Archives of Computational Methods in Engineering
https://doi.org/10.1007/s11831-024-10195-3

REVIEW

A Review on Multi-objective Optimization Techniques of Wire Electrical Discharge Machining

Devendra Pendokhare[1] · **Shankar Chakraborty**[1]

Received: 27 August 2024 / Accepted: 1 October 2024

Abstract

In the present-day manufacturing environment, wire electrical discharge machining (WEDM) has become one of the most efficient non-conventional material removal processes to generate complicated 2D and 3D profiles on many of the difficult-to-cut engineering materials. Although the material removal rate of this process is comparatively low, but it can provide high dimensional accuracy and tolerance along with excellent surface integrity. To explore its maximum potential, it is advised to operate this process at the optimal combination of its various input parameters, which can only be derived using some optimization tools. The past researchers have already applied several multi-objective optimization techniques to resolve the issue. This paper comprehensively reviews and documents applications of four major multi-objective optimization tools, i.e. desirability function approach, grey relational analysis (GRA), multi-criteria decision making methods and metaheuristic algorithms considered for parametric optimization of WEDM processes. It also extracts information regarding type of the experimental design plan, work and wire materials, dielectric utilized, and WEDM parameters and responses considered. It is observed that Taguchi's L_{27} orthogonal array has been the most commonly deployed design plan, while medium and high carbon steels, and brass have been the most prevalent work and wire materials, respectively. Most of the researchers have preferred deionized water as the dielectric and GRA as the multi-objective optimization technique. During WEDM experiments, pulse-on time and pulse-off time have appeared as the two most significant input parameters; and surface roughness has been the most important response, followed by material removal rate. The outcome of this review paper would help the future researchers to have an idea regarding initial settings of different WEDM parameters and achievable response values. It would also act as a data support for subsequent utilization in developing machine learning-based prediction models.

Abbreviations

ABC	Artificial bee colony
AHP	Analytic hierarchy process
ANFIS	Adaptive neuro-fuzzy inference system
Aoff	Arc-off time
BA	Bat algorithm
BOA	Bayesian optimization algorithm
C	Capacitance
CE	Circularity error
CNN	Convolutional neural network
CR	Corrosion rate
CSA	Cuckoo search algorithm
CYL	Cylindricity error
DBNN	Deep belief neural network
DEAR	Data envelopment analysis-based ranking
DFA	Desirability function approach
Dt	Wire diameter/thickness
EDAS	Evaluation based on distance from average solution
Fl	Lower flush pressure
Fs	Spark frequency
GA	Genetic algorithm
GET	Geometrical error top
GPR	Gaussian process regression
GRG	Grey relational grade
H	Hardness
HSA	Harmony search algorithm
Ip	Peak current
KW	Kerf width
MCDM	Multi-criteria decision making
MMC	Metal matrix composite
MODA	Multi-objective dragonfly algorithm
MOGOA	Multi-objective grasshopper optimization algorithm
MOMLNN	Multi-objective multi-layer neural network

✉ Shankar Chakraborty
s_chakraborty00@yahoo.co.in

[1] Department of Production Engineering, Jadavpur University, Kolkata, West Bengal, India

Published online: 17 October 2024