Pontifícia Universidade Católica
do Rio de Janeiro

Pedro Henrique Leite da Silva Pires Domingues

Artificial Intelligence Methods Applied to Mechanical Engineering Problems

Dissertação de Mestrado

Dissertation presented to the Programa de Pós–graduação em Engenharia Mecânica of PUC-Rio in partial fulfillment of the requirements for the degree of Mestre em Engenharia Mecânica.

Advisor: Prof. Helon Vicente Hultmann Ayala

Rio de Janeiro
April 2020

PUC-Rio - Certificação Digital Nº 1812727/CA

Pontifícia Universidade Católica
do Rio de Janeiro

Pedro Henrique Leite da Silva Pires Domingues

Artificial Intelligence Methods Applied to Mechanical Engineering Problems

Dissertation presented to the Programa de Pós–graduação em Engenharia Mecânica of PUC-Rio in partial fulfillment of the requirements for the degree of Mestre em Engenharia Mecânica. Approved by the Examination Committee.

Prof. Helon Vicente Hultmann Ayala
Advisor
Departamento de Engenharia Mecânica – PUC-Rio

Prof. Marco Antonio Meggiolaro
Departamento de Engenharia Mecânica – PUC-Rio

Prof. Roberto Zanetti Freire
Departamento de Engenharia de Produção e Sistemas – PUCPR

Rio de Janeiro, April 13th, 2020

PUC-Rio - Certificação Digital Nº 1812727/CA

All rights reserved.

Pedro Henrique Leite da Silva Pires Domingues

Mechanical Engineer graduated from the Universidade do Estado do Rio de Janeiro (UERJ) in 2017, under the guidance of Prof. Dr. José Roberto Moraes D'almeira

PUC-Rio - Certificação Digital Nº 1812727/CA

Bibliographic data

Domingues, Pedro Henrique Leite Silva Pires

Artificial Intelligence Methods Applied to Mechanical Engineering Problems / Pedro Henrique Leite da Silva Pires Domingues; advisor: Helon Vicente Hultmann Ayala. – 2020.
116 f. : il. color. ; 30 cm

Dissertação (mestrado) - Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Engenharia Mecânica, 2020.
Inclui bibliografia

1. Artificial Intelligence. 2. Multi-objective Optimization. 3. Bio-inspired Algorithms. 4. Machine Learning. 5. Supervised Learning. I. Ayala, Helon Vicente Hultmann. II. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Engenharia Mecânica. III. Título.

CDD: 621

Acknowledgments

Throughout my life, several people have helped me achieve my goals and with this work was no different, which makes me really grateful to all the people and collectives of people (i.e. institutions), who made this work possible. Therefore, the next few paragraphs are dedicated to acknowledledge these people.

First, I believe that the appearance of these people in my life is the product of some natural (or not) force, referred here as God, and for that force I am very grateful.

Also, I would like to thank my entire family, specially Ana Cristina, Evandro, Larissa, Ana Maria and Stella Maris, for always believing in me, giving me the best examples and providing me with sentimental and material support.

I would like to thank my advisor, Prof. Dr. Helon Vicente Hultmann Ayala, for giving support, offering me the opportunity to work on various articles, being patience to help me understand the subjects related to this work and helping me to solve the most diverse problems that arose along the way.

This work is also result of all Professors at PUC-Rio and UERJ who were part of my academic background, specially Prof. Dr. Jose Roberto Moraes D'almeida, my graduation advisor, Prof. Dr. Ivan Fabio Mota de Menezes for helping me understand the optimization concepts and Prof. Dr. Marco Antonio Meggiolaro for being the person who instructed me before I enrolled in the master's. So, thank you.

I would also like to thank the Prof. Dr. Leandro dos Santos Coelho and Prof. Dr. Roberto Zanetti Freire, who directly helped me in the articles development, together with my advisor.

Special thanks to all my friends, especially those made during the master's degree, Anna Raphaela Silva Ferreira and Rafael Rosenberg Santos for the moments of rest and work.

Thank you to PUC-Rio for the Master's scholarship provided.

Thank you to Prof. Dr. Luís Fernando Alzuguir Azevedo and SIMDUT for allowing me to collect data for this work

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001

PUC-Rio - Certificação Digital Nº 1812727/CA

Abstract

Domingues, Pedro Henrique Leite Silva Pires; Ayala, Helon Vicente Hultmann (Advisor). **Artificial Intelligence Methods Applied to Mechanical Engineering Problems**. Rio de Janeiro, 2020. 116p. Dissertação de Mestrado – Departamento de Engenharia Mecânica, Pontifícia Universidade Católica do Rio de Janeiro.

Real-world mechanical engineering problems may comprise tasks of i) multi-objective optimization (MO) or ii) regression, classification and prediction. The use of artificial intelligence (AI) based methods for solving these problems are widespread for i) demanding less computational cost and problem domain information to solve the MO, when compared with mathematical programming for an example; and ii) presenting better results with simpler structure, adaptability and interpretability, in contrast to other methods. Therefore, the present work seeks to i) optimize a proportional-integral-derivative control (PID) applied to an anti-lock braking system (ABS) and the heat exchanger design of plate-fin (PFHE) and shell-tube (STHE) types through AI based optimization methods, seeking to develop new versions of the applied methods, e.g. multi-objective salp swarm algorithm (MSSA) and multi-objective heuristic Kalman algorithm (MOHKA), which enhance the optimization performance; ii) develop a pipeline leak detection system (LDS) sensitive to fuel theft by training decision trees (DTs) with features based on time and principal component analysis (PCA), both extracted from pressure transient data of regular pipeline operation and fuel theft; iii) constitute an application guide for control and design MO problems, feature extraction process and machine learning classifiers (MLCs) training through supervised learning; and, finally, iv) demonstrate the potential of AI-based techniques.

PUC-Rio - Certificação Digital Nº 1812727/CA

Keywords

Artificial Intelligence; Multi-objective Optimization; Bio-inspired Algorithms; Machine Learning; Supervised Learning.

Resumo

Domingues, Pedro Henrique Leite Silva Pires; Ayala, Helon Vicente Hultmann. **Métodos de Inteligência Artificial Aplicados a Problemas de Engenharia Mecânica**. Rio de Janeiro, 2020. 116p. Dissertação de Mestrado – Departamento de Engenharia Mecânica, Pontifícia Universidade Católica do Rio de Janeiro.

Problemas reais de engenharia mecânica podem compreender tarefas de i) otimização multi-objetivo (MO) ou ii) regressão, classificação e predição. Os métodos baseados em inteligência artificial (AI) são bastante difundidos na resolução desses problemas por i) demandarem menor custo computacional e informações do domínio do problema para a resolução de uma MO, quando comparados com métodos de programação matemática, por exemplo; e ii) apresentarem melhores resultados com estrutura mais simples, adaptabilidade e interpretabilidade, em contraste com outros métodos. Sendo assim, o presente trabalho busca i) otimizar um controle proporcional-integral-derivativo (PID) aplicado a um sistema de frenagem anti-travamento de rodas (ABS) e o projeto de trocadores de calor de placas aletadas (PFHE) e casco-tubo (STHE) através de métodos de otimização baseados AI, buscando o desenvolvimento de novas versões dos métodos aplicados, e.g. *multi-objective salp swarm algorithm* (MSSA) e *multi-objective heuristic Kalman algorithm* (MOHKA), que melhorem a performance da otimização; ii) desenvolver um sistema de detecção de vazamento em dutos (LDS) sensível ao roubo de combustível a partir do treinamento de árvores de decisão (DTs) com *features* baseadas no tempo e na análise de componentes principais (PCA), ambas exraídas de dados de transiente de pressão de operação normal do duto e de roubo de combustível; iii) constituir um guia de aplicação para problemas de MO de controle e projeto, processo de extração de *features* e treinamento de classificadores baseados em aprendizado de máquina (MLCs), através de aprendizado supervisionado; e, por fim iv) demonstrar o potencial das técnicas baseadas em AI.

PUC-Rio - Certificação Digital Nº 1812727/CA

Palavras-chave

Inteligência Artificial; Otimização Multiobjetivo; Algoritmos Bio-inspirados; Aprendizado de Máquinas; Aprendizado Supervisionado.

Table of contents

PUC-Rio - Certificação Digital Nº 1812727/CA

PUC-Rio - Certificação Digital Nº 1812727/CA

List of figures

PUC-Rio - Certificação Digital Nº 1812727/CA

PUC-Rio - Certificação Digital Nº 1812727/CA

List of tables

PUC-Rio - Certificação Digital Nº 1812727/CA

List of Abbreviations

ABLF – Asymmetric Barrier Lyapunov Function
ABSs – Anti-lock Braking Systems
ACC – Ambiguity Correlation Classification
ACO – Ant Colony Optimization
AI – Artificial Intelligence
ANN – Artificial Neural Network ANPW – Attenuation of Negative Pressure Waves
ASO – Anarchic Society Optimization
CONCAWE – Conservation of Clean Air and Water in Europe COT – Crude Oil Theft
CSS – Charged System Search
DA – Dragonfly Algorithm
DE – Differential Evolution
DM – Decision Maker
dof – Degree of Freedom
DTs – Decision Trees
ED – Euclidian Distance
ELGP – Epigenetic Linear Genetic Programming
FDI – Fault Detection and Isolation
FLC – Fuzzy Logic Controller
GA – Genetic Algorithm
GP – Global Programming
GSA – Gravitational Search Algorithm
GWO – Grey Wolf Optimizer
HEs – Heat Exchangers
HKA – Heuristic Kalman Algorithm
HV – Hypervolume
IA-AIS – Improved Adaptive Artificial Immune System
ICA – Imperialist Competitive Algorithm
ICAO – International Civil Aviation Organization
IGD – Inverted Generational Distance
IGD_a – Area Under the Inverted Generational Distance Evolution Curve
IGD_f – Final Inverted Generational Distance Value
ITAE – Integrated Time Multiplied by the Absolute Error
LDS – Leak Detection Systems

PUC-Rio - Certificação Digital Nº 1812727/CA

ML – Machine Learning
MLCs – Machine Learning classifiers
MLPNN – Multi-layer Perceptron Neural Network
MO – Multi-objective Optimization
MODA – Multi-objective Dragonfly Algorithm
MOHKA – Multi-objective Heuristic Kalman Algorithm
MOP – Multi-objective Optimization Problem
MSSA – Multi-objective Salp Swarm Algorithm
NB – Naive Bayes
NPW – Negative Pressure Wave
NSGA – Non-dominated Sorting Genetic Algorithm
NSGA-II – Non-dominated Sorting Genetic Algorithm Version 2
OBLI-MSSA – Opposite Based Learning Initialization Multi-objective Salp Swarm Algorithm
PCA – Principal Component Anlysis
PDEs – Partial Differential Equations
PF – Pareto Front
PFHE – Plate-fin Heat Exchanger
PID – Proportional-Integral-Derivative
PO – Pareto Optimal
PPFC – Principal Product Function Components
PSO – Particle Swarm Optimization
RPV – Reservoir-Pipe-Valve
S – Spacing
SCA – Society and Civilization Algorithm
SSA – Salp Swarm Algorithm
STHE – Shell-tube Heat Exchanger
SVD – Singular Value Decomposition
SVM – Support Vector Machine
SVR – Support Vector Regression
VEGA – Vector Evaluated Genetic Algorithm
VMD – Variational Mode Decomposition
WPD – Wavelet Packet Decomposition
WPT – Wavelet Packet Transform
ZDT1 – Zitzler-Deb-Thiele Test Problem 1

PUC-Rio - Certificação Digital Nº 1812727/CA

PUC-Rio - Certificação Digital Nº 1812727/CA

If education alone does not transform society,
neither does society change without

Paulo Freire *Brazilian educator and philosopher.*

1
Introduction

The majority of engineering design, management and decision making problems may be stated as an optimization problem, usually considering minimizing a cost function, due to the dominance of this variable in the decision making. However, real-world applications are often complex and consider two or more conflicting objectives for the decision making process, which makes the multi-objective optimization (MO) the most suitable approach for this situation, producing solutions with different trade-offs between the objectives and allowing the designer or another decision maker (DM) to select the best final solution [7,15]. Therefore, the MO approach does not generates an unique solution, but a set of solutions of mathematically equal quality, known as non-dominated or Pareto optimal (PO) solutions. This set of solutions is called Pareto front (PF) and Pareto set, when represented in the objective and decision space, respectively [7, 16, 17].

One example of real-world multi-objective optimization problem (MOP) is the anti-lock braking systems (ABSs) improvement. The ABSs were originally created to prevent wheel locking to which traditional brakes are subjected, when abruptly triggered or during slippery surfaces braking, through braking torque control [11, 18]. Therefore, due to the security requirement of these systems, the study of ABSs performance improvement has been in evidence in the last years (e.g. [19–22]). However, these systems may also provide comfort during the brake activity [22], which makes multi-objective approach the most appropriate procedure for providing a set of solutions with different trade-offs between comfort and performance.

Another example of real-world MOP is the heat exchangers design optimization. Heat exchangers (HEs) are common devices used in various industry fields, with the main purpose of transferring thermal energy among two or more fluids, or also, a fluid and a solid surface [23], for this reason, the HE effectiveness directly impacts the processes efficiency to which it is applied. In order to improve the heat transfer effectiveness, several design features could be considered, as the variation of HE dimensions, the use of treated, rough or extended contact surfaces, coiled tubes and fluid vibration, among others [24, 25]. However, these design features bring a trade-off relationship

PUC-Rio - Certificação Digital Nº 1812727/CA

between the HE design economic cost and effectiveness, which is higher with lesser energy resource consumption.

In order to solve MOP, the classical mathematical programming (i.e. deterministic optimization approach) is not recommended, given that this procedure requires significant or sometimes unsustainable computational cost for demanding previous problem domain knowledge to direct the search or limit the search space. These limitations made the use of artificial intelligence (AI) optimization methods (i.e. stochastic approach) widespread in the MOP solution [7, 17].

Another classes of engineering problems, which comprehends classification, prediction or regression tasks, may not be addressed by MO algorithms. For these purposes, the use of AI techniques known as 'machine learning' (ML) are more conducive.

Machine learning is the science field focused on the development of techniques that allow a computer to learn or extract information from a data set by their own, without being directly programmed for this purpose [26]. The ML methods sought inspiration in psychological learning theories, i.e. i) 'behaviorism', where the learning comes from observation aspects; ii) 'cognitive theory', where learning is based on brain physiology; and iii) 'constructivism', where the learning is understand as a continuous process, in which new ideas and concepts are actively built [2]. Thus, ML techniques are applied in several products and industrial fields due to i) its simplicity and performance compared to other methods, when applied to complex problems; ii) the environment change (i.e. new data) adaptation ability; and iii) the possibility of extracting information from the problems, based on the created models [26].

One example of real-world engineering problem that could be dealt with ML methods is the fuel pipeline theft or pipeline illegal tapping problem. In the recent years, the use of pipelines for the transportation of fossil fuels, chemicals and water has become widespread due to their economic, ecological and transport efficiency. On the other hand, in leakage situations, large quantities of product are lost, which may cause economic loss and environmental damage [27]. Therefore, focusing on reducing the impacts of leaks, several leak detection systems (LDS) have been developed [28], being the main detection approaches based on acoustic waves (e.g. [29, 30]), optical fiber signals (e.g. [31, 32]) and negative pressure waves (e.g. [33, 34]) [35]. In 2018, the entity responsible for ensuring the safe use of petroleum products in Europe (CONCAWE) made the report no. 6/18 [36], where it was evident that i) the leakage frequency in cold pipelines increased at least 5 times in the period 2011-2016, when compared to the period 2006-2010 and ii) this increase is due to fuel theft activities,

PUC-Rio - Certificação Digital Nº 1812727/CA

since excluding this spillage cause, the leakage frequency drops. The economic impact of pipeline illegal tapping is heavy, e.g. only in the Niger Delta region, Nigeria, the losses related to crude oil theft (COT) reach 12 billion american dollars annually [37]. The pipeline illegal tapping has two approaches, i) an 'offensive theft' where a high amount of product is rapidly withdrawn, which is easily detected if the pressure transmission cables are not cut by the violators and ii) a 'sophisticated theft' where low flow rates are employed to prevent LDS from locating the tapping point [38]. While the first theft approach seems to be a security issue, the second one can be addressed by the improvement of LDS or the development of specific theft detection techniques. Therefore, some examples of machine learning application are provided in Section 1.2.

1.1 Artificial Intelligence Taxonomy

Briefly, the taxonomy of so-called 'artificial' or 'computational' intelligence is divided into major two groups: i) 'hard computing', where the precise models yield accurate solutions quickly; and ii) 'soft computing', where the imprecisions and uncertainties are used to design approximate models, which allows the application in complex problems [2].

The soft computing, in turn, is divided into five subgroups: i) 'fuzzy logic', proposed by Zadeh [39,40], the fuzzy logic models the human reasoning and inference about imprecise or incomplete information, allowing the fuzzy-based method to learn from experiential knowledge [2]; ii) 'neural networks', first introduced in 1943 by McCulloch and Pitts [41], the ANN are bio-inpired methods based on the brain learning process, where multiple simple processing units (neurons) are distributed in layers [26, 42]; iii) 'probabilistic methods', introduced by Erdos and Spencer in 1974 [43], these techniques deal with the uncertainty and imprecision by attaching to structure families a probability space, therefore, each sample point of a particular structure has a positive corresponding probability in this space [2]; iv) 'global search optimization methods', which comprehends the evolutionary and swarm intelligence algorithms. These methods are nature-inpired (i.e. based on political and social aspects of the society, physical phenomena or biology aspects, such as genetic inheritance and animal behavior) [1,2]; and finally v) 'machine learning', which are based on psychological learning theories, as described in Section 1. The taxonomy of AI is shown in Figure 1.1.

The global search optimization methods may be referred as nature-inpired or bio-inpired algorithms. In essence, these two concepts present close meanings, but nature-inpired is a major class that encompasses the bio-

PUC-Rio - Certificação Digital Nº 1812727/CA

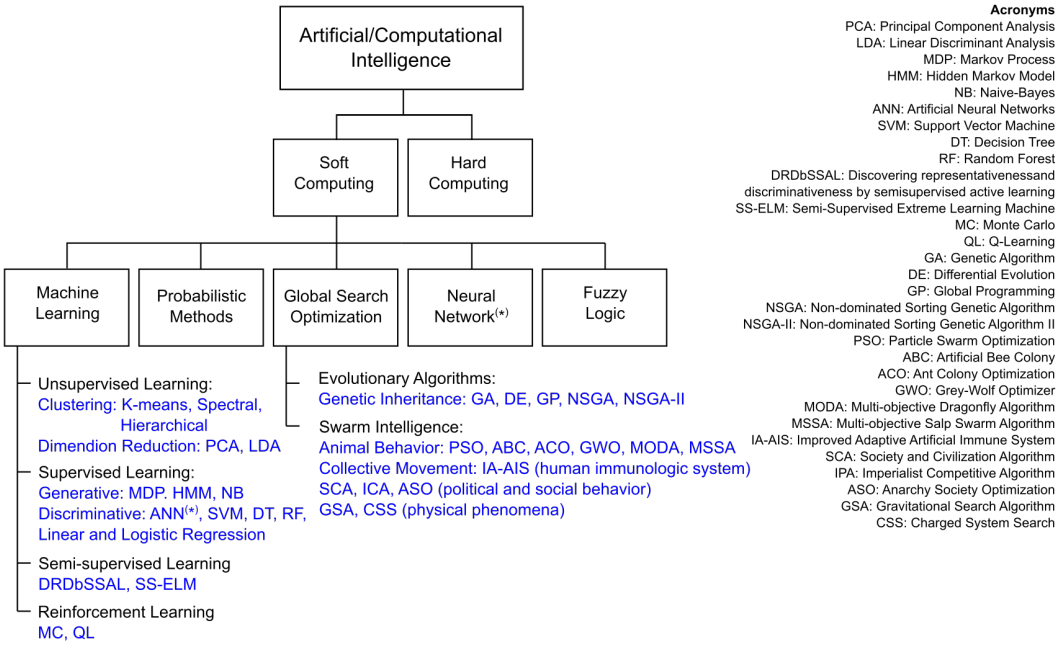

Figure 1.1: Artificial intelligence taxonomy [1–5].

PUC-Rio - Certificação Digital Nº 1812727/CA

inpired class [1]. The bio-inspired computation research area was fostered by genetic algorithm (GA), an evolutionary algorithm that deeply impacted the scientific community [1]. Thus, the bio-inspired computation sought inspiration from [1, 17, 44]: i) modern synthesis idea, e.g. GA [45], vector evaluated genetic algorithm (VEGA) [46], genetic programming (GP) [47], differential evolution (DE) [48], non-dominated sorting genetic algorithm (NSGA) [49] and its second version (NSGA-II) [50]; ii) animals behavior or collective movement of organisms, e.g. particle swarm optimization (PSO) [51], ant colony optimization (ACO) [52], improved adaptive artificial immune system (IA-AIS) [53]; grey wolf optimizer (GWO) [54], dragonfly algorithm (DA) and its multi-objective version (MODA) [9], salp swarm algorithm (SSA) and its multi-objective version (MSSA) [55]. While nature-inspired algorithms are based on [1]: i) physical phenomena, e.g. gravitational search algorithm (GSA) [56], charged system search (CSS) [57]; ii) political and social behaviors, e.g. society and civilization algorithm (SCA) [58], imperialist competitive algorithm (ICA) [59], anarchic society optimization (ASO) [60]. Moreover, neural networks are also considered bio-inspired techniques [26, 61].

Still, global search optimization methods may be also classified as heuristic, meta-heuristic and hyper-heuristic. Basically, the i) heuristic methods perform the optimization iteratively, following a set of guidelines, e.g. heuristic Kalman algorithm (HKA) [62] and its multi-objective version (MOHKA) [63], GSA, CSS, SCA, ICA, ASO and GA; ii) meta-heuristic techniques are heuris-

tic optimization algorithms with different strategies for the search space exploration (local) and exploitation (global), making the meta-heuristics less likely to get stuck in a local optimum, e.g. PSO, ACO, GWO, DA and SSA; iii) hyper-heuristics are searching/learning routines which combine or select heuristics for an optimization problem, practically automating the creation process of new heuristic algorithms [1, 64, 65].

As for ML techniques, four categories according to the learning process are provided: i) unsupervised learning, where the program identifies regularity or irregularity patterns in the data set, being used to label data groups (data cluster) or reduce data dimension; ii) supervised learning, where each data provided for the ML training has a label indicating the group/class to which it belongs, being used for classification applications, creating models to generate or discriminate data; iii) reinforcement learning, where the learning process is achieved by iteratively applying the program into an environment, in which each action has a reward or penalty. Thus, through the observation of the environment and the performance of previous iterations, according to the reward/penalty values, the program learns and selects the better actions; and, finally iv) semi-supervised learning, which comprises techniques that merge both supervised and unsupervised learning approaches. Also, ANN may also be considered as ML [26, 61].

PUC-Rio - Certificação Digital Nº 1812727/CA

1.2 Artificial Intelligence Applied to Mechanical Engineering Real-World Problems

The use of AI optimization methods for MOP resolution has become a trend in recent years [17], some examples are reported next.

Yang and Deb proposed a multi-objective version of the cuckoo search algorithm and applied the novel method to solve two structural design multi-objective problems, in 2013. The first problem considered the beam design according to fabrication cost and deflection minimization, while the second problem made the optimization of the disk brake design through the overall mass and braking time minimization [66].

In 2016, La Cava et al. proposed the use of epigenetic linear genetic programming (ELGP), i.e. an evolutionary MO method, to generate simbolic models of wind turbine, according to model intelligibility and accuracy goals [67].

Following to 2017, Nguyen and Vo used a modified cuckoo search algorithm to solve a hydrothermal scheduling problem, considering the minimization of power generation cost and thermal generators pollution emission

goals submitted to power balance, hydraulic and generator operating limit constrains [68]. In the same year, Pavao et al. proposed the use of a hybrid meta-heuristic based on simulated annealing (SA) and rocket firework optimization to develop the MO of a heat exchanger network, considering both total annual coast and environmental impact minimization [69].

Other mechanical engineering applications, which demand prediction or classification tasks may be addressed by ML techniques, as depicted next.

Li et al., in 2016, proposed a gearbox fault diagnosis that merge two training features through the random forest method. The acoustic and vibration signals are processed by wavelet packet transform (WPT), yielding statistical parameters that are applied to two deep Boltzmann machines. In them, the statistical features are produced, being merged by random forest later [70].

In 2017, Tahan, Muhammad and Karim proposed an automatic fault detection system for gas turbine engines. The method studied multi-net ANN models for detecting the faults in real time. These models were trained from several gas turbine simulated outputs [71].

Following to 2018, Winkler et al. proposed a data extraction and processing framework to model the failure of pipes in water distribution networks. The framework was based on decision trees (DTs), trained through existing data of age, diameter, number of failures, logarithmic length, material and type of the pipe and boosted with bootstrap aggregation and other boosting techniques [72].

Also, some other applications may be seen in the Wuest et al. review of ML advantages, challenges and applications in the manufacturing field [73], as well as the Bruton, Noack and Koumoutsakos review of ML for fluid mechanics [74].

The articles mentioned above help to justify the application of AI techniques to real-world mechanical engineering problems solving, since these problems: i) generally consider more than one optimization goal and MOP usually are high-dimensional, discontinuous, multi-modal or NP-hard [7], making the use of mathematical programming techniques not recommended, for demanding an unsustainable computational cost, as well as problem domain information [7]; and ii) when addressed by classification, prediction or regression applications, generally present a high complexity or an online operation requirement, making characteristics of high performance with simplified architecture and adaptability desirable.

PUC-Rio - Certificação Digital Nº 1812727/CA

1.3 Literature Review

The literature review on how the ABS improvement, HE design optimization and LDS development problems are addressed by the scientific community is presented in the Sections 1.3.1, 1.3.2 and 1.3.3, respectively. Still, greater importance was given to the problem of developing a theft-sensitive LDS, which resulted in a more complete literature review on LDS.

1.3.1 Anti-lock Braking Systems Improvement

This section introduces a brief review of how ABS systems have been developed and improved in the recent years: in 2010, Sharkawy developed an ABS based on a self-tuning non-linear fuzzy-PID (proportional–integral–derivative) controller optimized through GA. A single-wheel model was considered for the ABS construction, where the controlled variable was the slip. Also, the optimization was made seeking to minimize i) the braking distance and ii) the difference bewtween control and reference signal through integrated time multiplied by the absolute error (ITAE) metric, seeking to obtain the fastest response of the controller [19].

One year later, Raesian, Khajehpour and Yaghoobi proposed a self-tuning ABS based on a non-linear neuro-fuzzy PID controller. The study considered a conventional PID with an adaptive neuro fuzzy inference system capable of tuning the PID coefficients. The controlled variable was the slip and a single-wheel model was used [20].

In 2014, Guo, Jian and Lin proposed an ABS with regenerative braking, which seeks to maintain the optimal slip value through a fuzzy logic controller (FLC). The simulations were made considering a 3-degrees of freedom ('dof') vehicle model, as well as, Burckhardt friction, motor and hydraulic brake system models [75].

Following to 2015, Qiu, Liang and Dai proposed a non-linear ABS control based on asymmetric barrier Lyapunov function (ABLF) and backstepping dynamic surface control. The ABLF was used to make the sliding restrictions more flexible for runway conditions and the transition among them, while the dynamic surface control avoids the repeated differentiations introduced by the ABLF procedure. Also, the ABS approach considered the slip as the controlled variable and a single-wheel model [21].

Davico et al. presented an algorithm to prevent aircraft wheel locking based on deceleration of the wheel center, in 2017. For that purpose, they used a 'control-oriented landing gear model', also the performance and comfort were

PUC-Rio - Certificação Digital Nº 1812727/CA

evaluated in dry, wet and in transition situations between wet and dry runway condition. Still, the gear walk pheomena was considered in the landing gear model [22].

Finally, in 2018, Radac and Precup proposed a fast and nonlinear ABS based on a model-free tire slip control. In order to achieve this goal, a reinforcement Q-learning optimal control approach was applied to a batch neural fitted scheme, which comprehends two neural networks, the first for value function approximation and the second to perform the control [76].

The ABS literature review revealed that the: i) single-wheel models (e.g. 'single corner model') are sufficient and often used for the study and development of ABS; ii) majority of controllers present a high complexity in comparison with the conventional PID, generally using fuzzy, neural networks or other techniques to incorporate non-linearities; iii) use of new meta-heuristics for the controller tuning was little observed, with traditional optimization methods or neural networks being preferable.

1.3.2 Heat Exchanger Design Optimization

Seeking to review HE design problems: Khan and Li proposed, in 2017, a novel MO method constituted from the combination of GA, DE and adaptive SA. Four MO benchmarks were considered to validate the method, which was subsequently applied in a plate-fin heat exchanger (PFHE) design problem. The PFHE optimization was made three times, each one considering different goal combinations, always composed of the heat transfer rate and a cost metric, that last varying between design, annual and total cost [77].

One year later, Dhavle et al. proposed the design optimization of a shell-tube heat exchanger (STHE) for three different operation conditions through several AI optimization methods. The HE total annual cost (i.e. the sum of design and annual operation costs) was considered as the optimization objective [78].

In 2019, Vasconcelos Segundo, Mariani and Coelho proposed a new meta-heuristic inspired by the owls behavior. The introduced method was tested in several single and multi-objectives benchmarks, yielding competitive capcity, convergence, diversity and convergence-diversity results in comparison with other well-known optimization techniques. Afterwards, the novel algorithm was applied to single and multi-objective STHE design problems, considering the total cost alone and with the HE efficiency as objectives, respectively. [79].

Still, another HE design optimization study was developed by Pavao et al. [69], which was also described in Section 1.2.

PUC-Rio - Certificação Digital Nº 1812727/CA

Two trends were observed in the HE design literature review: i) the use of total cost and some HE performance metric (e.g. heat transfer rate and efficiency) as optimization objectives; and ii) the use of HE design problems to test and compare new meta-heuristics.

1.3.3 Leak Detection System Development for Pipeline Illegal Tapping

Finally, several LDS have been proposed during the last decade, which considered the variation of methods used to recognize the leak patterns, the type of data collected from the pipe flow and the processing techniques to which the data is submitted, as shown below.

In 2010, Yang, Xiong and Shao proposed an entity-part searching method based on negative pressure waves and wavelet transform, which used Romberg integral and Dichotomy Searching methods to iteratively calculate the gas velocity and pinpoint the leak location in gas pipelines, respectively [80]. Meanwhile, Qu et al proposed a pipeline leak detection, locating and pre-warning system for gas leakage in real time, which was based on support vector machine (SVM) and wavelet packet decomposition (WPD) for extracting features from the pipeline vibration signals [81].

Two years later, Mandal, Chan and Tiwari proposed a LDS for crude oil and liquid fuel pipelines based on SVM optimized through artificial bee colony algorithm, where the features were extracted from the data of pressure and flow rate at inlet and outlet, as well as the flow rate difference. The rough set theory was applied for the feature dimension reduction [82].

Instead of locating the leak, in 2014, Sun et al proposed a system capable of recognizing the aperture of small leaks in gas pipelines, which considered the local mean decomposition of the leakage pressure signals into product function components, following by the identification of the principal product function components (PPFC) through Kurtosis features. The PPFC are decomposed by WPD and reconstructed through Hilbert transform and the normalized envelop spectrum entropy feature obtained is used as input for the SVM [83].

In addition to the pipelines, LDS is also used in other applications, in 2015, Rostek, Morytko and Jankowska proposed a LDS for fluidized-bed boilers based on artificial neural networks (ANN) with two stages, i) early fault detection by virtual sensors and ii) leak isolation through fault state classification [84].

In 2016, Xiao et al proposed a LDS for small leaks, which considered the variational mode decomposition (VMD) of the acoustic sensor's signal, following by the construction of an adaptive de-noising algorithm based on

PUC-Rio - Certificação Digital Nº 1812727/CA

probability density function for processing noise components and reconstruct de-noised ones. Finally, ambiguity correlation classification (ACC) method is used to analyze the reconstructed components [85]. In the same year, Zadkarami et al proposed a fault detection and isolation (FDI) system capable of recognize pipeline leakage and suggest its location and severity. The FDI system was based on multi-layer perceptron neural network (MLPNN) classifier trained from statistical, wavelet transform and a merge of these two features, which were extracted from pipeline inlet pressure and outlet flow rate [86].

As a continuation of previous work, in 2017, Zadkarami et al proposed a leak diagnosis system based on the fused outputs of two MLPNN classifiers through Dempster-Shafer technique. The first and second MLPNN were trained from statistical and wavelet transform based features, respectively, also extracted from inlet pressure and outlet flow rate [87]. Also using ANN for LDS development, Rahmati et al proposed a gas pipeline LDS. The pipeline was sectioning into several parts and the inlet and outlet pressure of each part were considered for the leakage gas flow data generation with which the ANN would be trained [88]. In order to evaluate the performance of multi-label learning for water pipelines LDS construction, Kayaalp et al proposed LDS based on pressure data acquired from wireless sensors and three multi-label learning methods, namely random k-label sets, binary relevance k-nearest neighbors and binary relevance with SVM [89]. For other applications, Li et al proposed a LDS for gas pipeline valve based on SVM and kernel principal component analysis (kernel PCA), which was used to reduce dimension and extracts the most important features of acoustic emission sensors data. In terms of comparison, the authors built LDS using the aforementioned methodology, but varying the classifier for the k-nearest neighbor, ANN, naive Bayes (NB) and decision tree (DT) [90].

In 2019, Wang et al proposed a novel particle swarm optimization algorithm with saturation and mixed time-delays, in order to improve the accuracy of an oil pipeline LDS based on SVM, which was trained with acoustic wave sensor acquired data of normal operation and leakage [91]. Still under an optimization approach, Jia et al proposed a LDS, considering an optic fiber based distributed hoop strain sensing and support vector regression (SVR), which was optimized by genetic algorithm [92]. In the same year, Li et al proposed the attenuation of negative pressure wave (ANPW) methodology for pipeline LDS development. The novel ANPW method uses momentum and continuity equations to deduce the propagation equation of negative pressure wave (NPW) considering the pressure change rather than time difference, avoiding the traditional NPW problems regarding the velocity disturbance

PUC-Rio - Certificação Digital Nº 1812727/CA

and the time difference identification difficulty [35]. He et al proposed a pipeline monitoring and accidental leakage handling system framework based on big data, cloud computing, and internet of things technology, using a new NPW based leak localization methodology, and an emergency shutdown strategy in case of leaking, which calculates and selects the strategy that minimize the product spilled volume [93]. Still using the time that a NPW demands to propagate and back from the valve to the leak location, Diao et al proposed a LDS for reservoir-pipe-valve (RPV) structures, considering a modified transient-based method and an unsteady friction model [94]. Seeking to achieve a real time and accurate LDS for oil pipeline with low false alarm rate, Liu et al proposed the use of Markov features from pressure data, least square SVM and a two-stage decision scheme, which alternates between a short and a long term detection model according to the sample difficulty level [95]. In order to base the development of new LDS for high-pressure hydrogen ducts, Nagase et al proposed a mass flow rate and pressure distribution prediction model, considering unsteady expansion wave, the hydrogen acceleration caused by the cross-sectional area reduction between the duct and leak point, and finally, the Borda-type and cylindrical pipeline orifice contraction coefficient models [96]. Finally, Xie, Xu and Dubljevic proposed a LDS based on pipeline non-linear coupled first-order hyperbolic partial differential equations (PDEs) model and on statistical features extracted from upstream and downstream flow velocity. The method considered the building of a discrete-time Luenberger observer by solving the Riccati equation operator, allowing the reconstruction of the pressure and mass flow velocity evolution with limited measurements [97].

The recent literature review on LDS allowed us to draw four conclusions: i) the use of machine learning classifiers (MLCs) for the development of these systems is widespread. The MLCs are non-parametric classifiers usually able to identify and model complex patterns, often producing more accurate results when compared to parametric classifiers, especially for complex data with a large number of features [98]; ii) the supervised learning approach was used in all articles mentioned in LDS review, being the most common procedure for classification tasks [26]. In the supervised learning approach, the desired solutions (class labels) are provided to the algorithm along with training features [26]; iii) the use of DT classifiers for LDS construction was little explored, having been applied only in [90], where the performance yielded by the DT classifier was surpassed by the SVM one. This may be an indication that, for the LDS problem, DT classifiers perform less than other MLCs, justifying the DT approach low frequency of application for LDS construction.

PUC-Rio - Certificação Digital Nº 1812727/CA

However, DT classifiers are used in other oil and gas industry applications (e.g. [99–103]); iv) the use of PCA-based features for training MLCs focused on leak detection was also little explored, only being applied in [90]. Nonetheless, the PCA methodology is used in other refining industry applications (e.g. [102, 104–107]).

1.4 Objectives

The main goals of this thesis is to i) develop and improve AI techniques; ii) constitute a guiding framework for control and design MOPs, feature extraction process and machine learning classifier training through supervised approach; and iii) demonstrate the AI based methods potential in solving real-world mechanical engineering problems.

In order to accomplish the thesis general objectives the following propositions were made: i) develop new AI optimization methods by the performance enhancement of MSSA and MOHKA, considering the MOPs of ABS improvement and HE optimal design, respectively and developing theft-sensitive LDS from DTs trained through supervised learning approach; ii) guide the application of: AI techniques, such as NSGA-II, MODA and MSSA in the PID tuning for the ABS improvement and MOHKA for HE optimal design, being both MO; extraction of machine learning training features from pressure transients data through time and PCA based approaches; and, DT training for the development of a theft-sensitive LDS, considering the supervised approach and the feature extraction process just mentioned; and finally iii) yield valid results for the ABS, HE design MOP, as well as for the theft-sensitive LDS development problem.

PUC-Rio - Certificação Digital Nº 1812727/CA

1.5 Contributions

The following contributions in the AI field were made during the master's period:

- Bio-Inspired Multi-objective Tuning of PID-Controlled Antilock Braking System [108];
- Genetic Algorithm for Topology Optimization of an Artificial Neural Network Applied to Aircraft Turbojet Engine Identification [109];
- Multi-Objective Optimization of Heat Exchanger Design Through Heuristic Kalman Algorithm [110];

– Dynamic Multi-criteria Classifier Selection for Illegal Tapping Detection in Oil Pipelines [111].

Aware of the advantages of using AI optimization methods for solving multi-objective engineering problems and the results quality yielded by the ML techniques when applied to mechanical engineering problems, the present work condenses two already made and one forthcoming contribution, that were developed seeking to i) improve an antilock braking system (ABS) control through the MO approach, considering three bio-inspired optimization algorithms [108]; ii) optimize the design of plate-fin and shell-tube heat exchangers considering the MO approach and a recent heuristic [110]; and iii) develop theft sensitive leak detection systems for an oil pipeline based on ML and features extracted from pressure transient signals. An abstract of each work, listing the main contributions for the state of the art are provided next and, in order to associate the results obtained in each contribution to the objectives sought in it, as well as to relate the specific goals of each work with the general thesis objectives, Figure 1.2 is provided.

PUC-Rio - Certificação Digital Nº 1812727/CA

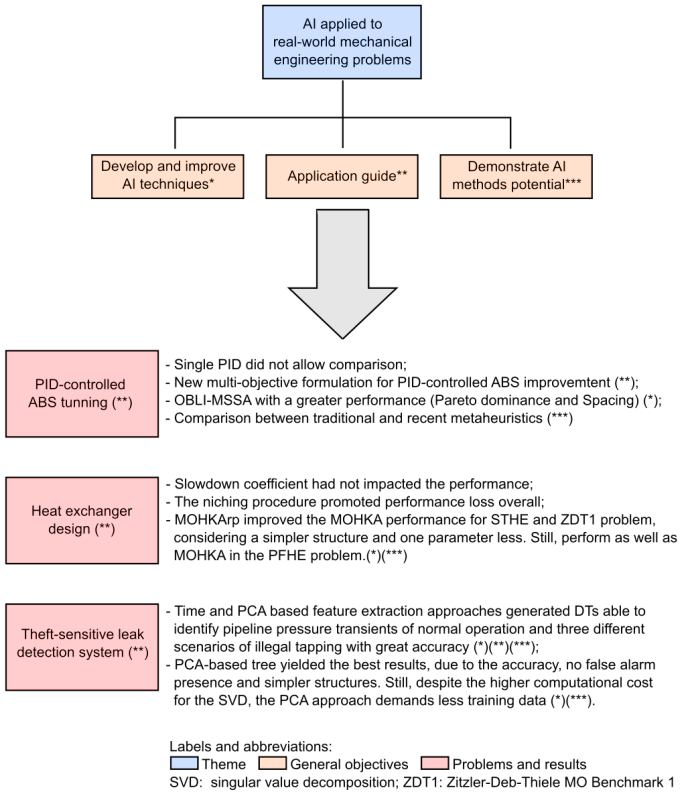

Figure 1.2: Relationship between thesis general goals and each contribution specific objectives.

1.5.1
Bio-Inspired Multi-objective Tuning of PID-Controlled Antilock Braking System

Antilock braking systems were proposed to overcome wheel lock events that might occur in sudden/heavy braking situations or when there is a change in road conditions during the braking activity (e.g. oil, sand or water in the driving way or landing strip), avoiding accidents. The ABS resolution requires a controller and, among controllers, the PID one is the most widespread in the industrial field due to its effectiveness and simplicity. In the literature, several ways of tuning PID controllers are provided, but recently, the use of MO methods has become attractive for this application. Therefore, the first work proposed the optimization of a PID-controlled ABS, considering both the performance and comfort developed by the system during the braking, being the optimization made through NSGA-II, MODA and MSSA, as well as a new MSSA version with initialization and evolution grounded in opposite based learning idea (OBLI-MSSA). Also, the spacing (S), Euclidian distance (ED) and hypervolume (HV) comparison metrics were used to deduce which optimization method yielded the best performance. Briefly, i) a new multi-objective formulation for improvement of PID-controlled ABS; ii) a new MSSA version, known as OBLI-MSSA; and iii) the comparison of recent bio-inspired meta-heuristics (MODA and MSSA) with traditional optimization methods (NSGA-II) to follow the state of the art evolution are the main contributions of this work.

1.5.2
Multi-Objective Optimization of Heat Exchanger Design Through Heuristic Kalman Algorithm

Heat exchangers are equipment widely used in the industry, which makes its design optimization a hot topic. Since that the heat exchangers design optimization usually considers its effectiveness and cost as goals, the academic community has been addressing the use of heuristic based MO algorithms for this purpose. Among the recent proposed heuristics is the multi-objective heuristic Kalman algorithm (MOHKA), which was developed for non-convex constrained optimization problems and stands out for the ease of implementation and the presence of few adjustable parameters. Basically, the MOHKA adjustable parameters are the population size (N), the maximum number of evaluations (E_{max}), the amount of solutions considered for the measure ξ calculation (N_ξ), being ξ an important variable for the evolution process, with which the population variance is calculated and, finally, the

PUC-Rio - Certificação Digital Nº 1812727/CA

slowdown factor (a_k) responsible for curbing the convergence, avoiding local minima, being based on the slowdown coefficient α. Therefore, considering the MO of PFHE, STHE and the optimization benchmark ZDT1 problem [112], the second work proposed five new MOHKA versions with i) randomness addition by N_ξ suppression; ii) new architecture, where new populations are derived only from ξ; and, iii) change of the diversity preservation mechanism from 'crowding distance' to 'niching procedure'. Also, the variation of α is studied, considering a range of [0.4,0.9]. For the MOHKA versions comparison, the area under the inverted generational distance (IGD) evolution curve and the final IGD value are considered as metrics and are compared through Wilcoxon rank sum test. In essence, the main contributions sought are i) the MOHKA performance improvement; ii) enhance the MOHKA attractive features, by reducing the number of parameters and simplifying its structure; and iii) assess the impact of the slowdown coefficient on the PF convergence.

1.5.3 Pressure Transient Signals Feature Extraction for Illegal Tapping Detection with Supervised Learning

The use of pipelines for the transportation of fuels, chemicals and water has become widespread due to the low cost and environmental impact, in addition to high efficiency. In recent years, there has been an increase in the number of pipeline failures owing to the fuel theft attempts growth. Therefore, to counteract this trend and reduce/avoid the environmental damage and economic loss resulting from the high amount of spilled product that a pipeline failure generally promotes, the scientific community has been studying and developing LDS sensitive to theft. Reviewing the LDS design literature, the use of ML classifiers (MLCs) trained with a supervised learning approach is highlighted, but among the MLCs, the use of a simple and common one, known as decision tree (DT) classifier, as well as the use of features extracted through PCA is little explored. In order to extract the features, pressure sensors meaurements stands out for being highly used in pipeline network supervisory systems and presenting lower costs when compared to other sensor options, such as optical fiber ones. Aware of this situation, the third work proposed to reproduce in a physical pipeline model based on a real one, the pressure transients of pipeline leakage and regular operation for three different scenarios. The pressure transient data was used to generate features based on time and PCA made through singular value decomposition (SVD), with which two DT classifiers were trained. The main contributions sought are to i) propose two different data-driven modeling workflows for theft-sensitive LDS;

PUC-Rio - Certificação Digital Nº 1812727/CA

ii) evaluate the use of DT classifiers for this application; and iii) evaluate which feature extraction approach generates the best DT model, according to the prediction performance in different scenarios.

1.6 Organization

The rest of this thesis is structured as follows: Chapter 2 introduces general concepts necessary for the artificial intelligence techniques comprehension, the comparison metrics for the algorithms performance, as well as the algorithms themselves; Chapters 3, 4 and 5 present the problem description, main contributions with the approach chosen to address the problem, results and conclusion of the PID-controlled ABS tuned through bio-inspired meta-heuristics, the heat exchanger design through MOHKA and LDS development through ML techniques problems, respectively; finally, Chapter 6 summarizes all conclusions and future work for each developed study.

PUC-Rio - Certificação Digital Nº 1812727/CA

2
Methodology

This section is dedicated to describe the methods used in the contributions made in the present work. Therefore, Section 2.1 introduces the basic optimization concepts; Section 2.2 presents the multi-objective optimization approach through Pareto Optimally Theory; Section 2.3 defines the mathematical expressions used to compare the optimization performance between two or more AI algorithms; Section 2.4 declares the GA, which inspired the structure of many other AI optimization algorithms and facilitates the comprehension of the other used methods, also described in this section; and, finally, Section 2.5 depicts the ML techniques used in this dissertation.

PUC-Rio - Certificação Digital Nº 1812727/CA

2.1
Basic Concepts of Optimization

The present section seeks to introduce basic optimization concepts necessary for understanding the multi-objective approach (Section 2.2) and the work routine of the stochastic multi-objective optimization methods (Section 2.4) used in the three works developed.

The main goals in the design engineering and decision making understands an optimization problem solving [15]. Therefore, the optimization of a generic process, as shown in the Figure 2.1, consists of producing and testing various solutions (input), aiming to improve (minimize or maximize) the process answer to that solution (output) [6].

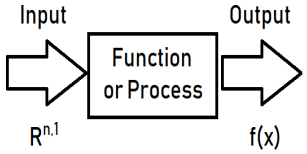

Figure 2.1: Representation of a generic process [6].

In the optimization, each solution is represented in the 'decision/genotype space' (R^n) and 'objective/phenotype space'(R^g), being n and g the number of decision variables (i.e. the problem dimension) and goals, respectively. Thus, jointly, the input and output variable values constitute points in the decision

and objective spaces. For a better visualization, in Section 2.2, the Figure 2.2 shows the decision and objective spaces for a bi-objective optimization problem ($g = 2$), where each solution considers $n = 3$ decision variables.

A basic optimization problem generally considers a single objective, what is known as a single-objective optimization problem. Thus, the standard statement of a basic optimization problem (Eq 2-1) is defined as the minimization/-maximization of a called objective or cost function ($f(x)$), subjected to equality ($h_i(x) = 0; i = 1, \ldots, p$) or inequality ($b_j(x) \leq 0; j = 1, \ldots, m$) restrictions. The objective functions are mathematical abstractions of qualitative concepts (e.g. cost and performance), and as the objective function may depend on more than one parameter/variable, x is a vector solution composed by n decision variables ($x = [x_1, \ldots, x_n]^T$). The x vector solution belongs to the space Ω of possible solutions, which is determined by the upper and lower bounds of each decision variable ($x_k^l \leq x_k \leq x_k^u; k = 1, \ldots, n$) [7].

$$\begin{cases} min \quad f(x) & x \in R^n \\ s.t. & \\ \qquad h_i(x) = 0 & i = 1, \ldots, p \\ \qquad b_j(x) \leq 0 & j = 1, \ldots, m \\ \qquad x_k^l \leq x_k \leq x_k^u & k = 1, \ldots, n \end{cases} \tag{2-1}$$

The previous statement helps to define the concept of i) 'search space', which is the problem domain or the subspace of the decision space limited by the upper and lower bounds of the decision variables; and ii) 'optimum' for single-objective optimization problems, which is referred as 'global optimum'.

The global optimum is the best solution among the possible solutions (i.e. in the search space), that when applied to the process function and compared with all other valid solutions, returns the minimum or maximum value, for minimization or maximization problems, respectively. The methods capable of finding the global optimum solution are known as global optimization method. Mathematically, for minimization problems: considering a function $f : \Omega \subseteq R^n \rightarrow R, f \neq \emptyset$, for a solution $x \in \Omega$, the value of f^* defined as $f(x^*)$ and being greater than minus infinity ($f^* \triangleq f(x^*) > -\infty$) is called a global minimum, if and only if $\forall x \in \Omega, f(x^*) \leq f(x)$ [7].

2.2 Multi-Objective Optimization

The present Section introduces the multi-objective approach, which bases the optimization methods that will be presented in Section 2.4.

PUC-Rio - Certificação Digital Nº 1812727/CA

The multi-objective optimization approach has a standard statement similar to the single-objective optimization one, but now considering the minimization or maximization of a vector $F(x)$ composed by two or more objectives, with each vector solution x consisting of n decision variables between their upper and lower bounds ($x_k^l \leq x_k \leq x_k^u; k = 1, \ldots, n$), and submitted to equality ($h_i(x) = 0; i = 1, \ldots, p$) or inequality ($b_j(x) \leq 0; j = 1, \ldots, m$) restrictions. Also, the standard statement of a multi-objective problem is represented by Eq.(2-2) [7].

$$\begin{cases} min \;\; F(x) = [f_1(x), \ldots, f_g(x)] \;\; x \in R^n; g \geq 2 \\ s.t. \\ \qquad h_i(x) = 0 \qquad\qquad i = 1, \ldots, p \\ \qquad b_j(x) \leq 0 \qquad\qquad j = 1, \ldots, m \\ \qquad x_k^l \leq x_k \leq x_k^u \qquad\qquad k = 1, \ldots, n \end{cases} \tag{2-2}$$

As seen in Section 2.1, for the single-objective optimization, the 'optimum' solution may be easily inferred by comparing the optimization outputs, since only one objective is taken into account for the optimization and/or solution comparison. However, when two or more objectives are considered for the optimization process, the single-objective idea of 'optimum' is no longer suitable, seeing that the outputs now present a trade-off between the optimization metrics. In that case, the multi-objective approach is recomended [7, 17].

Therefore, the most common definition of 'optimum' for MOP is set by the Pareto Optimally Theory, coming from the Pareto Optimum or Edgeworth-Pareto Optimum concept proposed by Francis Ysidro Edgeworth in 1881 [113] and generalized by Vilfredo Pareto [114] in 1897 [7]. The definitions of the Pareto Optimally Theory are provided next [7]:

Definition 1. Pareto optimal: A solution vector $x \in \Omega$ is said a Pareto Optimal or true Pareto solution, with respect to (w.r.t) Ω, iff $\neg x \in \Omega : f_i(x) \leq f_i(x^*) \wedge f(x) \neq f(x^*); \forall i = 1, 2, \ldots, k$.

Definition 2. Pareto dominance: A solution vector x^1 dominate another feasible solution x^2 (or formally $x^1 \succ x^2$) iff $f_i(x^1) \leq f_i(x^2) \wedge \exists j : f_j(x^1) \leq f_j(x^2); \forall ij = 1, 2, \ldots, k$. The solution x^1 is said non-dominated, if no other solution dominates x^1.

Definition 3. Pareto set: A set of non-dominated solutions $x^* \mid \neg\exists x : x \succ x^*$ is said to be a Pareto set.

Definition 4. Pareto front: The set of vectors in the objective space which are image of a Pareto set, i.e. $F(x^*) \mid \neg\exists x : x \succ x^*$. The PF composed of true Pareto solutions is known as PF_{true}.

PUC-Rio - Certificação Digital Nº 1812727/CA

Briefly, for working with a set of possible solutions, the multi-objective optimization is capable of finding several PO solutions in a single optimization run, rather than having to perform multiple runs as in mathematical programming techniques [7]. Also, the representation of Pareto set and Pareto front for a bi-objective minimization problem are shown in Figure 2.2.

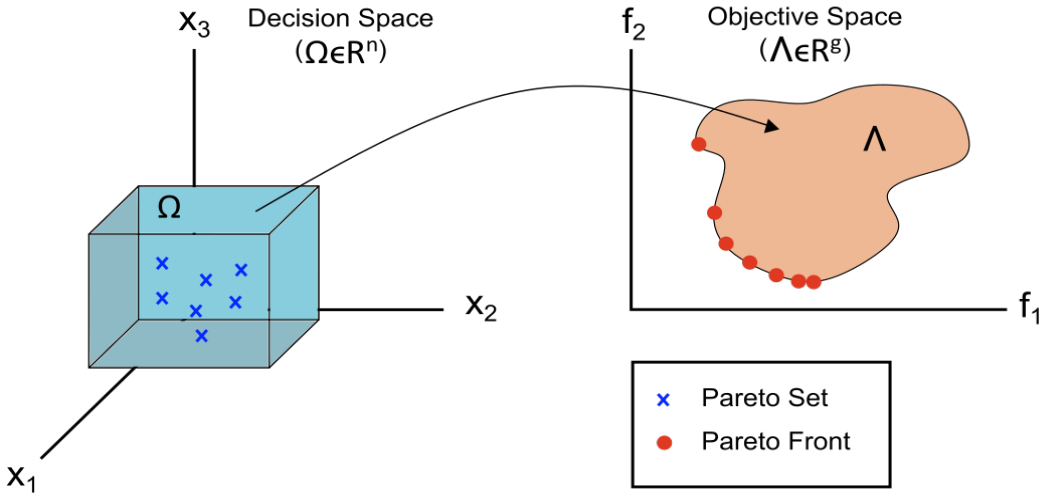

Figure 2.2: Representation of the solutions in decision and objective spaces [7].

The degree of freedom ('dof') on the context of the MOP is substantial to evaluate the possibility of optimization. Hence, the number p of equality constrains must be less than the number n of decision variables to exist sufficient 'dof' for optimization. If $p \geq n$, the problem is said overconstrained, not having enough equations for the number of unknows [7].

2.3 Performance Comparison

As seen in Section 2.2, the optimization performed through multi-objective approach provides a PF as an answer, and when working with stochastic optimization methods, the same method may generate widely varied PFs when the optimization is finished. Thus, to make a fair performance comparison of these methods: i) all optimization algorithms must have the same number of evaluations as stop criterion, since this operation usually carries the highest computational cost. Another option would be to match the number of optimization iterations, but when the methods present different architectures, there may be distinct number of evaluations per iteration, making the comparison unfair; ii) all algorithms must have the same number of optimization rounds, i.e. a complete optimization process that lasts until the

PUC-Rio - Certificação Digital Nº 1812727/CA

stopping criterion is satisfied, so that the PF extracted from all optimization rounds results has a statistically relevant comparison. Still, the greater is the optimization rounds number, greater is the comparison relevance; and iii) comparison metrics are applied to the resulting PFs, in order to assess their quality. The evaluation criteria for each comparison metric used in the three works, as well as their mathematical description is presented in this section.

2.3.1 Spacing

Spacing metric (S) seeks to quantify how well the solutions are scattered over the PF and does not require knowledge of the true PF. For a bi-objective optimization problem with objective vector $F(x) = [f_1(x), f_2(x)]$, Spacing is defined as follows [7]:

$$S \triangleq \sqrt{\frac{1}{P-1}\sum_{i=1}^{P}(\bar{d} - d_i)^2} \tag{2-3}$$

where P is the number of vectors or solutions in the PF; d_i is the Euclidean distance between the ith solution and the nearest solution, in the objective space; and $\bar{d}$ is the mean of all d_i.

For a better comprehension, the Figure 2.3 presents the visual description of the spacing metric. When $S = 0$ all members will be evenly spaced. In MO, preference is given to the smaller S, which means a more constant distribution of solutions over the PF.

2.3.2 Euclidean Distance

The mean of Euclidean distance of all vectors defined by points of the PF and origin in the objective space is knows as Euclidian Distance metric (ED), as illustrated in Figure 2.3. For a minimization problem, the smaller the result, closer to the origin (i.e. the minimum) is the PF, therefore, better is the method performance. Mathematically, for a bi-objective problem:

$$ED = \frac{1}{P}\sum_{i=1}^{P} \parallel \overrightarrow{o} - f^i(x) \parallel \tag{2-4}$$

where $\overrightarrow{o}$ is the origin vector in R^2; $f^i(x)$ is objective vector for the ith solution; and $\parallel . \parallel$ is an operator that calculates the l^2-norm.

PUC-Rio - Certificação Digital Nº 1812727/CA

2.3.3 Hypervolume

Hypervolume (HV) is the evaluation function defined by the hyper area calculated from the sum of rectangles formed between the PF solutions and some reference point, which may be the point $R(1, 1)$, in a normalized objective space. The reference point and the PF form in the objective space a hyper-edge and a hyper-surface, respectively. Together, the calculated hyper areas constitute a hypervolume, as shown in Figure 2.3. Therefore, the closer to 1 is the HV (i.e. the maximum possible volume, when considering normalized objective axes), the closer to the objective axes the analyzed PF will be, thus being better when considering a minimization problem. The hypervolume conceptual description is presented below [7]:

$$HV \triangleq \{\bigcup_i vol_i \mid vec_i \in PF\} \tag{2-5}$$

where vec_i is a non-dominated vector in PF; and vol_i is the area between the origin and vector vec_i.

2.3.4 Inverted Generational Distance

The IGD calculates the mean square of the shortest distances d_j, from each true Pareto-optimal solution j and the nearest computed solution, for that reason, the IGD is a metric that consider not only the convergence but the distribution of the resulting PF [115]. For a better comprehension, the Figure 2.3 [7] brings a visual description of the d_j vector and, being n the total number of solutions in the true PF, the IGD is mathematically expressed as [116]:

$$IGD = \frac{\sqrt{\sum_{j=1}^{n} d_j^2}}{n} \tag{2-6}$$

However, two IGD-dependent metrics are assumed for the performance analysis and comparison of the optimization methods. The first metric is referred as IGD_a and is the integral of the 'IGD x evaluations curve', where, for each iteration the IGD and evaluation numbers yielded are archieved in order to set up a convergence curve to the true PF at the end of the optimization. The axes are normalized and the smaller the area under the curve, greater is the algorithm convergence to the real PF. The second metric is referred as IGD_f and is the IGD value at the end of the optimization, which indicates how nearest is the computed PF to the true PF, being the lowest values the best. Both IGD_a and IGD_f metrics are illustrated in Figure 2.4.

PUC-Rio - Certificação Digital Nº 1812727/CA

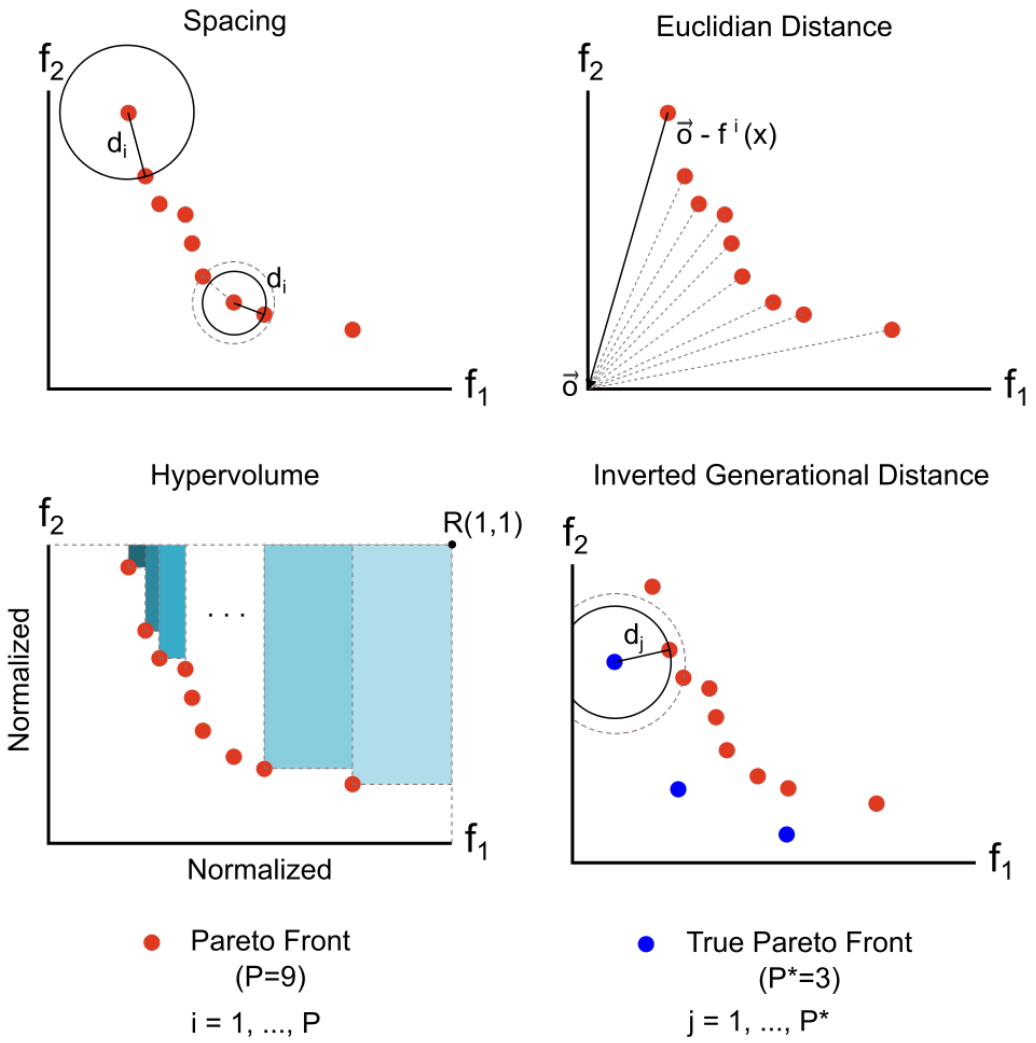

Figure 2.3: Visual description of the comparison metrics.

2.3.5 Wilcoxon Rank Sum Test

The Wilcoxon test is a non-parametric statistical procedure that seeks to assess whether two sets of data (X_1 and X_2) effectively represent two distinct groups.

The test is a non-parametric analogous of the paired t-test and is defined as follows: being $d_i = X_{1,i} - X_{2,i}$ the performance score difference in relation to two sets of data, for $i = 1, \ldots, n$ problems, R^+ and R^- are calculated as depicted in Eq.(2-7a) and Eq.(2-7b), respectively. Still, so that no problem is prioritized, the normalization between $[0, 1]$ of the compared scores may be done. From the smaller rank value R^+ or R^-, the variable T is derived (i.e. $T = min(R^+, R^-)$), and with the number of problems/'dof' n, the p-value related to the negation of the null hypothesis (h_0) that 'both groups are equal'

PUC-Rio - Certificação Digital Nº 1812727/CA

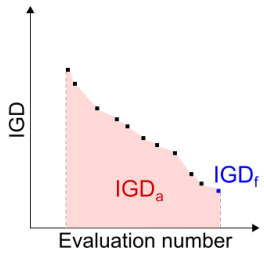

Figure 2.4: Representation of IGD_a and IGD_f metrics on a IGD evolution curve.

is extracted from interpolation of Table B.12 'critical values of Wilcoxon T distribution', present in Biostatistical Analysis [117]. In summary, the test states that the populations or groups X_1 and X_2 are distinct and that the probability of this sentence being wrong is the p-value. In this work, the Wilcoxon rank sum test was performed by a pairwise multiple comparison of mean ranks package [118] for R [119].

$$R^+ = \sum_{d^i>0} rank(d^i) + \sum_{d^i=0} rank(d^i) \tag{2-7a}$$

$$R^- = \sum_{d^i<0} rank(d^i) + \sum_{d^i=0} rank(d^i) \tag{2-7b}$$

2.4 Multi-objective Optimization Algorithms

For single-objective problems, the classical mathematical methods (i.e.) proved to be very effective to several economic and engineering problems [120]. However, the classical approach is not suitable when the problem considers more than one objective, since that, in this situation, the mathematical programming techniques assume a high computational cost making the optimization impossible, difficult or demanding knowledge of the problem domain to reduce the search space [7, 17]. In this context, the use of stochastic methods for the resolution of MOP has become widespread. Therefore, the present Section seeks to describe i) the GA, an important stochastic method that deeply impacted the optimization field, inspiring the development of many other techniques and promoting the research area, that would come to be called 'bio-inspired computation' [1]; and ii) the stochastic optimization algorithms used in the papers developed during this research.

PUC-Rio - Certificação Digital Nº 1812727/CA

2.4.1 Genetic Algorithm

The genetic algorithm was proposed in 1975 by John Holland, but first applied in 1989, by one of his students, solving a complex problem of gas-pipeline transmission control for his thesis [121]. The GA is a stochastic optimization method based on the modern synthesis idea, expressed as merger of the 'Mendelian genetic' (reproduction and mutation) with 'Darwinian evolutionary theory', also known as 'survival of the fittest' (selection) [7, 16].

First proposed for single-objective optimization problems, the GA considers the creation of a N_{pop} sized initial population, according to each decision variable upper and lower bounds. Each solution in the population is called a chromosome and has its cost evaluated by the objective functions. The variables may be encoded as real or binary numbers, in the last case, the parameter N_{bits} defines the number of bits encoding each variable and a decoding step is applied on the chromosomes before solutions cost evaluation. The next step is the selection of N_{keep} mates, which occurs based on a crossover rate X_{rate} ($N_{keep} = X_{rate}N_{pop}$). The parent pairs may be selected from the population top to bottom, randomly, having a selection probability inversely proportional to their costs (in the case of a minimization problem) or through tournament. Following, the crossover/mating operation cuts the parents cromossomes in one, two, three points or following a mask, generating offsprings from the cut pieces recombination or distributing the bits/decision variables of each parent alternatively to their offsprings ('uniform crossover'). Usually, two parents produces two offsprings, therefore $N_{pop} - N_{keep}$ offsprings are generated at the end of the mating process. After the crossover, random mutations are applied in the bits/decision variables, according to a mutation probability μ, generally the best solutions are not submitted to that operation, also, in the final iteration the mutation process is generally not applied. Finally, the algorithm evaluates if the stop criteria (e.g. number of evaluations, iterations or valid solutions) was satisfied: i) in negative case, a new population is generated through the processes of 'decoding', in case of binary encoding, 'fitness evaluation', 'select mates', 'mating' and 'mutation'; and ii) in positive case, the optimization comes to an end. The single-objective GA workflow is presented in Alg. 1 and, for multi-objective optimization, the GA considers the sum of weighted cost funtions or the multi-objective approach [6].

Therefore, the main advantages of the GA are the: i) possibility to optimize with continuous or discrete variables; ii) no problem knowledge requirement for derivative operation; iii) parallel searches of the decision space, as well as the ease to use multiple computers at the same time for

PUC-Rio - Certificação Digital Nº 1812727/CA

Algorithm 1 GA pseudocode

```
Define cost functions, decision variables bounds, stop criteria and GA
parameters (N_pop,N_bits,N_keep,X_rates,μ);
Generate initial population (real or binary encoding);
while Stopping criteria is not satisfied do
    if Encoding = Binary then
        Decode chromosomes;
    end if
    Find cost of each chromosome;
    Select mates;
    Mating/crossover;
    Mutation;
end while
```

the optimization task; iv) support of solutions with large number of decision variables; v) power to deal with complex cost surfaces, by the presence of a mutation operator that avoids local minima; vi) generation of a set of nondominated solutions, allowing more decision making options; and vii) possibility to encode the variables as binary or real numbers [6].

PUC-Rio - Certificação Digital Nº 1812727/CA

2.4.2 Non-dominated Sorting Genetic Algorithm II

Based on GA [121], in 1994, Srinivas and Deb proposed NSGA [49]. In 2002, Deb, Pratap, Agarwal and Meyarivan reformulated the NSGA, seeking to reduce computational complexity in the sorting process of non-dominated solutions, adding elitism and eliminating the need for an adjustable parameter called *sharing parameter*. The algorithm obtained was named NSGA-II and presented, in the majority of tested problems, a better spread of solutions and convergence to true PF, when compared to multi-objective optimization evolutionary algorithms focused on the creation of a diverse PF [8].

The NSGA-II is a fast and elitist multi-objective genetic algorithm that differs from the original GA [121] in the way that the selection of crossover pairs is made [17]. The selection process is based on i) a front score, named rank; and ii) a crowding distance value. The rank value is ordered according to Pareto non-dominance and allows the representation of multiple objectives in a single fit value, giving efficiency to the method [7, 17]. When comparing two solutions of the same rank, both non-extreme, crowding distance operator constructs a cuboid between two closest neighbors and ranks from the largest to the smallest perimeter, preferring to choose solutions with wider cuboid perimeter, that means the most isolated ones, to fill new population [8, 17]. For a better understanding of the NSGA-II method, the Figure 2.5 shows the

selection mechanism scheme, also, its pseudocode is presented in Alg. 2.

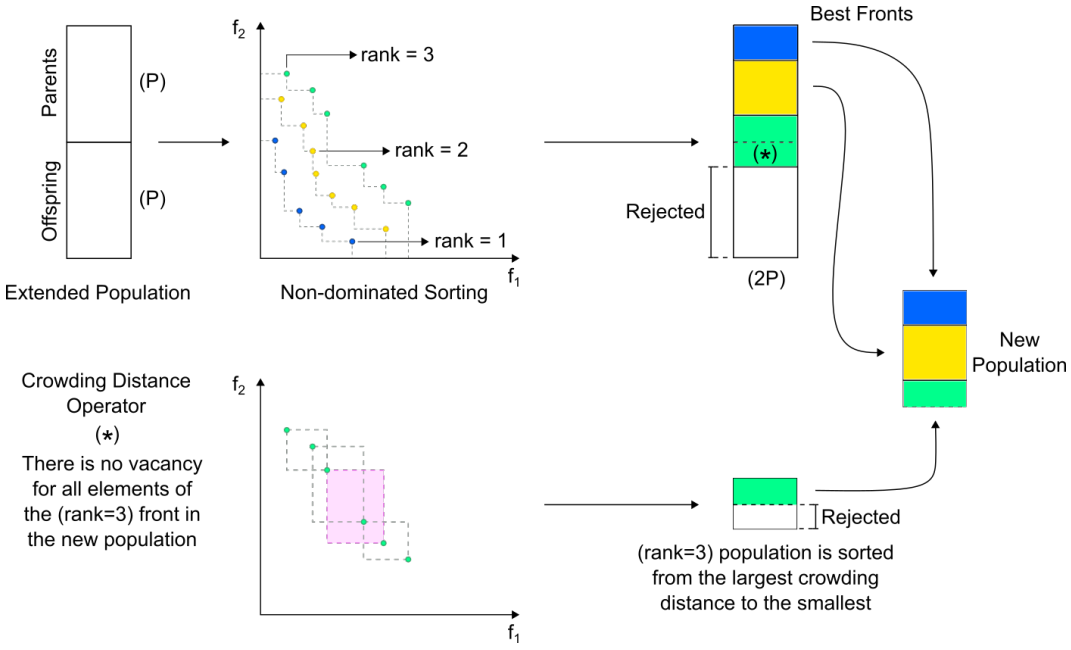

Figure 2.5: Selection process scheme for NSGA-II [8].

Algorithm 2 NSGA-II pseudocode

1: Create a population of size P, evenly distributed in the search space and initialize the generation counter $t = 1$;
2: Evaluate the cost of each individual and order the population according to Pareto non-dominance;
3: Assign a rank value according to the degree of non-dominance;
4: **while** Stopping criteria is not satisfied **do**
5: Perform parent selection through a binary tournament;
6: Generate a offspring population of size P, through the uniform crossover and mutation operators;
7: Evaluate the cost of each individual in the offspring population;
8: Add the offspring population to the parent one, forming an extended population of size $2P$;
9: Sort the population extended according to Pareto dominance;
10: Fill the new population of P individuals with the best fronts of the extended population;
11: Call the crowding distance operator to decide which individuals to keep in the new population, if the set of solution for a specific rank may only be partially introduced into the new population;
12: Update the generation number ($t = t + 1$);
13: **end while**

PUC-Rio - Certificação Digital Nº 1812727/CA

2.4.3 Multi-Objective Dragonfly Algorithm

The MODA is the multi-objective version of DA method, being inspired in the dragonflies behavior, which only swarms for predatory and migratory activities. The hunting behavior of the swarm resembles as the exploration stage (local search) in the optimization, with local movements and abrupt changes in the flying path, while the migratory behaviour resemble as the exploitation stage (global search) in the optimization, making a massive number of solutions to walk in one direction, over long distances [9]. The Figure 2.6 illustrates the change in search behavior of MODA, that allows classifying the method as a meta-heuristic.

The step vector of dragonflies ΔX_{t+1}, which is the vector that indicates the displacement of solution $i = 1, \ldots, n$ between the iteration t and $t + 1$, is described as follows [9]:

$$\Delta X_{t+1} = (sS_i + aA_i + cC_i + fF_i + eE_i) + w\Delta X_t \tag{2-8}$$

where s, a, c and w are the separation, alignment, cohesion and inertia weights; f and e are the food and enemy factor; S_i, A_i, C_i, F_i and E_i quantifies the separation, alignment, cohesion, food attraction and enemy distraction of the i-th individual; and ΔX_t is the step vector for the individual i in the last iteration, being t the iteration counter.

In order to obtain the step vector ΔX_{t+1}, the separation, alignment, cohesion, food attraction and enemy distraction vectors are calculated as depicted in Eq.(2-9)- Eq.(2-13), respectively. The position X_{t+1} is updated following Eq.(2-14). Still, for a better understanding of the step vector calculation, the Figure 2.6 seeks to facilitate the visualization of how the separation, alignment, cohesion, food attraction, enemy distraction and inertia vectors are constructed and the idea behind them.

$$S_i = -\sum_{j=1}^{N} X - X_j \tag{2-9}$$

$$A_i = \frac{\sum_{j=1}^{N} V_j}{N} \tag{2-10}$$

$$C_i = \frac{\sum_{j=1}^{N} X_j}{N} - X \tag{2-11}$$

$$F_i = X^+ - X \tag{2-12}$$

$$E_i = X^- + X \tag{2-13}$$

$$X_{t+1} = X_t + \Delta X_{t+1} \tag{2-14}$$

PUC-Rio - Certificação Digital Nº 1812727/CA

where X or X_t are the position of the current individual i at the iteration t; X_j and V_j indicates the position and velocity of the j-th neighbouring individual, being $j = 1, \ldots, N$, where N is the total number of neighbouring individuals; X^+ and X^- indicate the position of the food source and enemy, respectively.

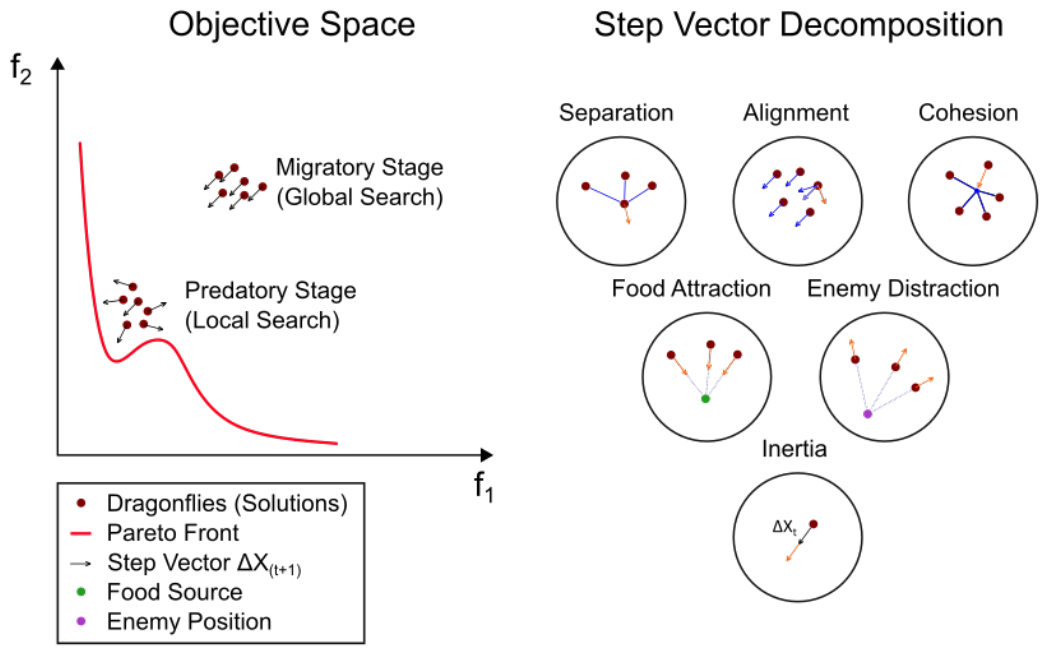

Figure 2.6: Change in the dragonflies (solutions) displacement behavior in the objective space according to the optimization stage [9].

The step vector is directly dependent on the s, a, c, f, e and w weights. Therefore, to perform the method search behavior change, characterized by the high alignment and low cohesion at the exploration stage, as well as the low alignment and high cohesion when exploiting, the mentioned weights are adaptively adjusted according to iteration number. Another important factor for the optimization process is the number of neighbors, which affects the parameters S_i, A_i and C_i calculation. So, to consider more individuals on the final optimization stage, a growing neighborhood radius is assumed, being also based on the iteration number [9].

The MODA uses an archive to store and retrieve the best solutions. Therefore, food sources and enemies position are selected from the archive, preferring the least populated region solutions for food sources and the most crowded for enemies position, for the purpose of finding a well-spread PF. In order to select solutions to add to the archive, in each iteration all dragonflies are compared with the saved solutions using the Pareto Dominance concept and if i) a dragonfly of the new population dominates one of the repository solutions, they are swapped; ii) a dragonfly dominates a set of solutions in

PUC-Rio - Certificação Digital Nº 1812727/CA

the repository, that new dragonfly is added and the set is eliminated from the repository; iii) at least one saved solution dominates a dragonfly of the new population, that last is discarded; finally iv) a dragonfly of the new population is non-dominated when compared with the other repository solutions, this dragonfly has to be added to the archive. These set of rules can guarantee that the archive always store non-dominated solutions obtained so far by the algorithm and is known as archive maintenance mechanism [9].

Still, to obtain the least crowded area, the best and worst objectives of the current archive data are found and a hyper-sphere that covers all the PF solutions is defined. With the hyper-sphere segmentation, a procedure known as 'roulette wheel mechanism' is applied ir order to define the probabilites depicted in Eq.(2-15) and Eq.(2-16). Thereby, Eq. (2-15) assigns a high probability of choosing food sources in less crowded regions, while Eq. (2-16) set a great probability of selecting as enemies position the solutions in the most crowded segments of the hyper-sphere. A summary of the MODA working structure is presented through its pseudocode in Alg. 3 [9].

$$P_i = \frac{c}{N_i} \tag{2-15}$$

$$P_i = \frac{N_i}{c} \tag{2-16}$$

where P_i is the probability and N_i is the number of Pareto optimal solutions of the ith hyper-sphere segment; c is a constant value greater than one.

2.4.4 Multi-Objective Salp Swarm Algorithm

The MSSA is a multi-objective optimization algorithms based on salp swarm displacement behavior. The salps belong to the Salpidae family, being similar to jelly fishes, due to their i) transparent body; and ii) propulsion displacement mode by pumping water through the body [122]. In deep waters, salps form chains and, although there is no consensus, some researchers believe that this swarm behavior is reproduced to improve the locomotion capability [123].

Thus, in order to mathematically abstract the salp chains behavior, the salps population is divided into two groups: leader and followers. The first individual of the chain guides the movement, being known as 'leader', while the 'followers' follow each other [55]. Therefore, considering that each salp position is a solution in a n-dimensional search space, where n is the problem dimension, the learder position depends on the food source position and is governed by [55]:

PUC-Rio - Certificação Digital Nº 1812727/CA

Algorithm 3 MODA pseudocode

Create a dragonflies population of size n, evenly distributed in the search space;
Initialize the step vector ΔX_i, $i = 1, \ldots, n$;
Define parameter N_{arc}, the number of hyper-sphere segments and initialize the generation counter $t = 1$;
while Stopping criteria is not satisfied **do**
    Evaluate the fit value of all dragonflies;
    Find the non-dominated solutions;
    if the archive is full **then**
        Run the archive maitenance mechanism to update the repository;
    end if
    if any of the new solutions added in the archive is located outside the hyper-sphere **then**
        Update and re-position all hyper spheres to cover new solutions, if these are located outside of hyper spheres;
    end if
    Select food source X^+ and an enemy X^- from archive;
    Update the step vector with Eq. (2-8);
    Update the position vector with Eq. (2-14);
    Update the generation counter $t = t + 1$;
    Check and correct the new positions based on the variable bounds.
end while

$$x_j^1 = \begin{cases} F_j + c_1\left(\left(ub_j - lb_j\right)c_2 + lb_j\right) & c_3 \geq 0 \\ F_j - c_1\left(\left(ub_j - lb_j\right)c_2 + lb_j\right) & c_3 < 0 \end{cases} \tag{2-17}$$

$$c_1 = 2e^{-\left(\frac{4l}{L}\right)^2} \tag{2-18}$$

where x_j^1 represents the salps chain leader position; F_j is the food source position; ub_j and lb_j are the upper and lower bounds, all in the jth problem dimension, being $j = 1, \ldots, n$; l and L are the current and the final/total iteration number; and finally c_1, c_2 and c_3 are MSSA parameters.

The c_1 is the most important parameter in the MSSA method, because it balances the exploration and exploitation stages. The c_2 and c_3 parameters are random numbers uniformly generated in the $[0, 1]$ interval, which define whether the new position in the jth dimension will towards to positive infinity or negative infinity, as well as the step size [55].

The formula that updates the position of follower salps is based on Newton's law of motion, which is depicted in Eq. (2-19). However, considering $v_0 = 0$ and a discrete approach of the problem, where time t is related to a single iteration, always being unitary ($t = 1$), the formula that updates the followers' position can be seen in Eq. (2-20) [55].

$$x_j^i = \frac{1}{2}at^2 + v_0 t \tag{2-19}$$

PUC-Rio - Certificação Digital Nº 1812727/CA

considering $i \geq 2$ (followers), x_j^i is the position of ith follower salp in jth dimension; t is current time; v_0 is the initial speed; and $a = \frac{v_{final}}{v_0}$, where $v = \frac{x - x_0}{t}$.

$$x_j^i = \frac{1}{2}\left(x_j^i + x_j^{i-1}\right) \tag{2-20}$$

The MSSA uses a repository of best solutions and consider an archive maintenance mechanism similar to that used in MODA (Section 2.4.3). In cases of archive fully complete, preference is given to the removal of one solution on a most populated region, focusing on finding an uniformly distributed PF, the most crowded region is defined by a neighborhood distance $\overrightarrow{d} = \frac{\overrightarrow{max} - \overrightarrow{min}}{repositorysize}$, where $\overrightarrow{max}$ and $\overrightarrow{min}$ are two vectors that store the maximum and minimum values for every objective, respectively. However, the food source is chosen from the least crowded region. For a better comprehension of the MSSA structure, the MSSA pseudocode is presented in Alg. 4 [55].

Algorithm 4 MSSA pseudocode

1: Create the salps population of size P evenly distributed in the search space, where each $X \in R^n$, being n the number of decision variables;
2: Initialize the generation counter $t = 1$;
3: **while** Stopping criteria is not satisfied **do**
4: Evaluate the fitness for all salps;
5: Find the non-dominated solutions of the salps population;
6: Update the repository based on the archive maintenance mechanism selection of non-dominated solutions;
7: **if** the repository is fully complete **then**
8: Run the archive maitenance mechanism to update the repository;
9: **end if**
10: Select a food source F from the repository;
11: Update c_1 from Eq. (2-18);
12: Update the position of the leading and followers salps from Eq. (2-17) and Eq. (2-20), respectively;
13: Amend salps population based on upper and lower bounds of return variables;
14: Update the generation counter $t = t + 1$;
15: Return the repository.
16: **end while**

PUC-Rio - Certificação Digital Nº 1812727/CA

2.4.5
Multi-Objective Heuristic Kalman Algorithm

The heuristic Kalman algorithm (HKA) was proposed by Toscano and Lyonnet [62], based on the Kalman filter philosophy, where the solution generated has its noisy corruption iteratively removed by Kalman equations, ensuring an optimal solution. The multi-objective version of the HKA (MOHKA) was proposed by Ayala, Coelho and Reynoso-Meza [63], being divided in five steps, namely as:

Step 1. Initialization: where the mean m_k and the standard deviation Σ_k is calculated based on the decision variables bounds, as follows:

$$m_k = \begin{bmatrix} \mu_1 \\ \vdots \\ \mu_n \end{bmatrix}, \mu_j = \frac{\bar{x}_j + \underline{x}_j}{2}, j = 1, 2, \ldots, n \tag{2-21}$$

$$\Sigma_k = \begin{bmatrix} \sigma_1^2 & \ldots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \ldots & \sigma_n^2 \end{bmatrix}, \sigma_j = \frac{\bar{x}_j + \underline{x}_j}{6}, j = 1, 2, \ldots, n \tag{2-22}$$

where $\underline{x}_j$ and $\bar{x}_j$ are the lower and upper bounds for the j-th decision variable $x \in \Omega$, being Ω the solution space; and k is the iteration number, initially null.

Step 2. Gaussian generator: where a set of N new solutions are generated, derived from a normal distribution above the m_k and Σ_k, thus comprising the solutions set $X(k) = \{x_1^k, x_2^k, \ldots, x_N^k\}$.

Step 3. Measurement process: where the solutions in the $X(k)$ set are evaluated, generating the $Z(k)$ set of same size, but arranged in ascending order, according to the Pareto-dominance and descending crowding distance when the solutions have the same rank. All the non-dominated solutions ($rank = 1$) are archived in a temporary repository until it reaches the total file size of N_a solutions. In case of file fully complete, the same criteria used to sort the Z_k set is considered to select the solutions that remain in the archive [8]. In the next iteration, the i-th solution from $X(k)$ that was added to the archive has its mean started as x_i^k and the same standard deviation from the solution in the archive which generated it [63]. Also, the measurement ξ^k is calculated by the mean of the N_ξ best fit solutions in the archive, and from ξ^k, the variance V^k is obtained, as described below [63]:

PUC-Rio - Certificação Digital Nº 1812727/CA

$$V^k = \frac{1}{N_\xi}\left[\sum_{i=1}^{N_\xi}(x_{i,1}^k - \xi_{1,1}^k)^2, \cdots \sum_{i=1}^{N_\xi}(x_{i,n}^k - \xi_{1,n}^k)^2\right] \tag{2-23a}$$

$$\xi^k = \frac{1}{N_\xi}\sum_{i=1}^{N_\xi} x_i^k \tag{2-23b}$$

where N_ξ is the number of candidates chosen from the gaussian generator; $x_{i,j}^k$ denotes the j-th dimension or decision variable of the i-th solution at the instant k.

Step 4. Kalman estimator: where the Kalman gain L_k and slowdown factor a_k are calculated through Kalman filter equations depicted below:

$$L_k = \Sigma_k(\Sigma_k + diag(V_k))^{-1} \tag{2-24a}$$

$$W_k = \left[vec^d\left[(I_k - L_k)\,\Sigma_k\right]\right]^{1/2} \tag{2-24b}$$

$$a_k = \frac{\alpha \min\left(1, \left(\frac{1}{n}\sum_{i=1}^n \sqrt{v_i^k}\right)^2\right)}{\min\left(1, \left(\frac{1}{n}\sum_{i=1}^n \sqrt{v_i^k}\right)^2\right) + \max_i\left(w_i^k\right)} \tag{2-24c}$$

where the gaussian generator standard deviation vector is denoted by $S_k = \left(vec^d\left(\Sigma_k\right)\right)^{1/2}$, being $vec^d(.)$ an operator that returns the diagonal components of a matrix as vector; v_i^k and w_i^k are the i-th component of V_k and W_k, respectively; finally, $\alpha \in (0, 1]$ is a scalar known as slowdown coefficient, but the typical values are comprehended between [0.4,0.9] [62].

Step 5. Mean and standard deviation update: finally, in possession of the Kalman gain L_k and slowdown factor a_k, the mean and standard deviation vectors are updated as follows:

$$m_{k+1} = m_k + L_k\left(\xi_k - m_k\right) \tag{2-25a}$$

$$S_{k+1} = S_k + a_k\left(W_k - S_k\right) \tag{2-25b}$$

After one iteration, which comprehends the resolution of Steps 2-5 for all solutions in the native archive, the temporary repository becomes the new native archive. For a better understanding of the algorithm, its pseudocode is presented in Alg. 5 [63].

PUC-Rio - Certificação Digital Nº 1812727/CA

Algorithm 5 MOHKA pseudocode

Set the parameters N, N_a, N_ξ and α;
Initialize m_0 and Σ_0 through Eq. (2-21) and Eq. (2-22);
Initialize the iteration counter $t = 1$;
Evaluate the fit value for all solutions in the population;
Identify non-dominated solutions and save them in the native archive;
while Stopping criteria is not satisfied **do**
    for each solution in the native archive **do**
        Gaussian generator: a new population $X(t)$ with gaussian distribution is set;
        Measurement: evaluation and sorting of $X(t)$ solutions and V_k calculation through Eq. 2-23a;
        Find non-dominated solutions and save them in the temporary repository;
        if the temporary repository is fully complete **then**
            Use the crowding distance concept, so that the temporary repository is always composed of $rank = 1$ solutions with greater crowding distance;
        end if
        Kalman estimator: L_k calculation through Eq. (2-24a);
        Mean and standard deviation update through Eq. (2-25a) and Eq. (2-25b);
    end for
    Native archive update, becoming the same as the temporary file;
    Update the iteration counter $t = t + 1$;
end while

PUC-Rio - Certificação Digital Nº 1812727/CA

2.5 Machine Learning

Machine learning techniques are inspired by the psychological learning theories, as introduced in Section 1. Also, the possible machine learning approaches and their applications were presented in Section 1.1. Therefore, the present section seeks to describe the binary classification decision tree and principal component analysis, which are the supervised and unsupervised learning methods used in the third work, respectively.

2.5.1 Binary Classification Decision Trees

Decision trees are a conceptually simple and interpretable supervised learning method, which are capable to perform both classification and regression tasks [10, 26]. The DT seeks to partition the feature space into a set of rectangular regions R_m by selecting a feature X_j and a split point t_{m-1} recursively, until a stopping rule is satisfied, being j the total amount of features and m the final number of regions or terminal nodes in each step [124].

In order to select the feature X_j and the cutpoint t_{m-1} for each node, it is considered the 'recursive binary splitting' procedure, where the decision tree split begins from the top to bottom, dividing the current branch in two more branches according to the split point t_{m-1}, which leads to a better tree or briefly, reduce the 'impurity' of the current node [124].

Seeking to decide the predictor X_j to split and the consequent cutpoint t_{m-1}, three methods are presented, i) the Gini index, introduced by in Eq. (2-27), ii) cross-entropy or deviance, depicted by Eq. (2-28) and iii) twoing rule presented in Eq. (2-29). The first two are impurity measures methods, therefore the goal is to reduce these indices, while the third method is a different measure of split point decision, which should have its value maximized. For the Gini index and deviance methods it is necessary to calculate $\hat{p}_{mk}$, the proportion of class k in the region m, which is depicted as follows [124]:

$$\hat{p}_{mk} = \frac{1}{n_m} \sum_{x_i \in R_m} I(y_i = k) \tag{2-26}$$

where n_m is the number of observations in the region m and y_i is the real class/label related to the observation x_i, being $i = 1, \ldots, n_m$.

$$G = \sum_{k=1}^{K} \hat{p}_{mk}(1 - \hat{p}_{mk}) \tag{2-27}$$

$$D = -\sum_{k=1}^{K} \hat{p}_{mk} \log \hat{p}_{mk} \tag{2-28}$$

$$P(L)P(R)(\sum_{k=1}^{K} \mid L(k) - R(k) \mid)^2 \tag{2-29}$$

where K is the number of classes; $P(L)$ and $P(R)$ are the fractions of observations that split to the left and right of the cutpoint, respectively; $L(k)$ and $R(k)$ denote the fraction of class k members in the left and right child or subsequent after the split, respectively.

Thus, the class associated with each region m is the one with the highest incidence in its class ($k(m) = \max_k \hat{p}_{mk}$). The growing process of a decision tree considering a bi-dimensional feature space is represented in Figure 2.7.

PUC-Rio - Certificação Digital Nº 1812727/CA

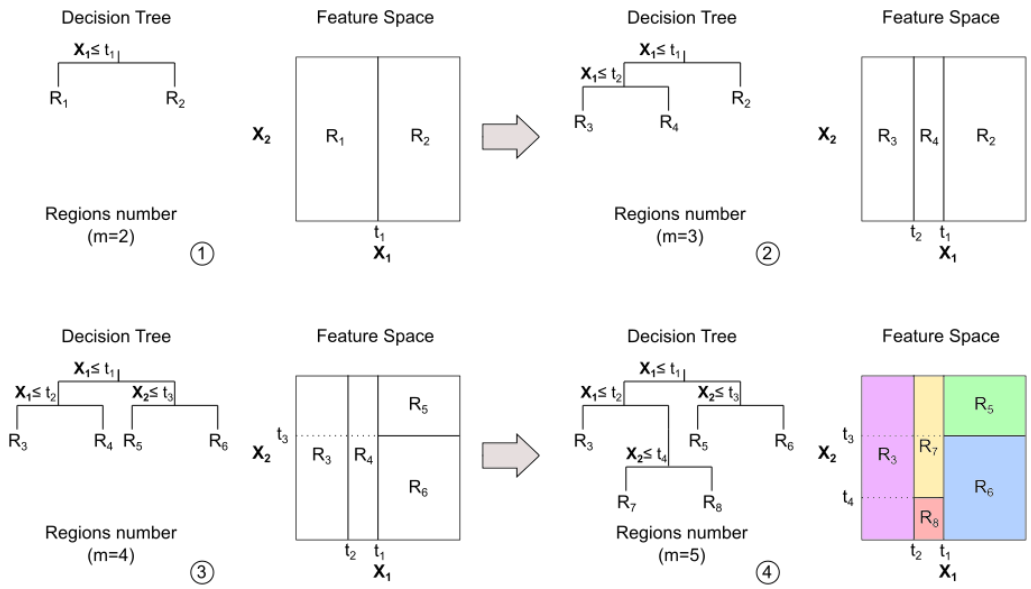

Figure 2.7: Decision tree growing process and the consequent split in the feature space [10].

PUC-Rio - Certificação Digital Nº 1812727/CA

2.5.2 Principal Component Analysis

The principal component analysis (PCA) is an unsupervised method for dimensionality reduction, first proposed by Pearson in 1901 [125]. The main idea of the PCA is to identify a lower-dimensional hyperplane close to the training data, in a way that the hyperplane preserves the maximum amount of variance, losing the minimum of information from the data set [26].

In order to perform the PCA, the singular value decomposition (SVD) may be used, being one of the main applications of this matrix factorization method [26, 126]. Therefore, given a matrix X, with i observations for each of the j features or data, being $i \leq j$, the SVD is conduced by the following expression [126, 127]:

$$\underset{i\times j}{X} = \underset{i\times i}{U} \times \underset{i\times j}{\Sigma} \times \underset{j\times j}{V}^{T} \tag{2-30}$$

where U and V are unitary matrices, being U columns composed by eigenvectors associated with the symetric matrix XX^T and V columns composed by the eigenvectors associated with the e largest eigenvalues of X^TX; and $\Sigma = diag(\sigma_1, \ldots, \sigma_e)$ is a diagonal matrix of the non-negative square roots of the eigenvalues of X^TX, called singular values and is assumed that $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_e \geq 0$.

Thus, seeking to obtain a matrix P with p principal components for each observation i in the matrix X, the PCA is made according to Eq. (2-31b), being $p < j$, since the objective is to reduce the matrix X dimension (i.e. number of features or data for one observation) [126, 127].

$$\underset{j\times p}{C} = columns_p(V) \tag{2-31a}$$

$$\underset{i\times p}{P} = (\underset{j\times p}{C}^T \times \underset{i\times j}{X}^T)^T \tag{2-31b}$$

where $columns_p(.)$ is an operator that builds a matrix from the p first columns of another one, so C is defined as a matrix composed of the p first columns of V.

PUC-Rio - Certificação Digital Nº 1812727/CA

3
Bio-Inspired Multi-objective Tuning of PID-Controlled Antilock Braking System

The present section seeks to depict the first work: 'Bio-Inspired Multi-objective Tuning of PID-Controlled Antilock Braking System' [108]. Therefore, this section is divided as follows: Section 3.1 introduces the ABS problem; Section 3.2, reviews the main contributions related to the first work, presenting the methodology associated with them; Section 3.3 addresses the obtained results; and, finally Section 3.4 summarizes the conclusions drawn and future work suggestions.

3.1
Problem Description

Seeking to reduce the vehicle speed, a braking torque is exerted on opposite way of the wheel angular velocity, which decreases the same and induces a friction force F_x between the tire and ground, that is contrary to the motion and promotes the reduction of longitudinal speed of the vehicle as shown in Figure 3.2. However, the relation of angular and longitudinal speed is not linear, since there is a physical limit of friction, which when exceeded, results in a slip rate between the tire and the ground, that during abrupt braking or on slippery surfaces, increases rapidly leading to wheel locking and loss vehicle control [128]. Thus, in order to prevent the wheel lock, the wheel slip may be controlled based on a friction model, as depicted in Section 3.1.1; the wheel model used to simulate the braking action is presented in Section 3.1.2; and the PID control law is introduced in Section 3.1.3.

3.1.1
Friction Model

Therefore, seeking to estimate the friction coefficient μ, the Burckhardt model expressed in Eq. (3-1) was used by assuming a strictly longitudinal braking, with sideslip and camber angles close to zero, the longitudinal force F_x may be defined as the product of vertical force F_z by friction coefficient μ [11].

$$\mu(\lambda, v_r) = v_{r1}\left(1 - e^{-\lambda v_{r2}}\right) - \lambda v_{r3} \tag{3-1}$$

PUC-Rio - Certificação Digital Nº 1812727/CA

where λ is the slip rate and v_{r1}, v_{r2} and v_{r3} are empirical parameters obtained for each type of soil.

In order to visualize the friction coefficient by slip value curves, i.e. the Burckhardt curves, the Figure 3.1 was generated on top of the empirical parameters v_{r1}, v_{r2} and v_{r3} available in the Table 3.1.

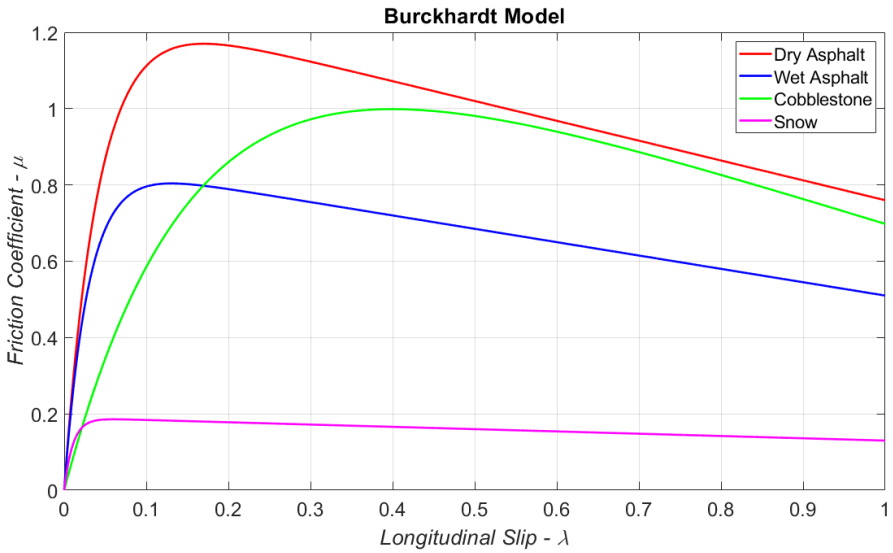

Figure 3.1: Burckhardt model curves for dry and wet asphalt, cobblestone and snow soil conditions.

Table 3.1: Burckhardt model empirical parameters values according to soil condition [11].

Soil condition	v_{r1}	v_{r2}	v_{r3}
Dry asphalt	1.28	23.99	0.52
Wet asphalt	0.86	33.82	0.35
Cobblestone	1.37	6.46	0.67
Snow	0.19	94.13	0.06

3.1.2 Single Corner Model

The single corner model consists of a single decoupled wheel dynamics, where suspension and tire deflection dynamics are disregarded, as shown in Figure 3.2. Considering that, in braking situations, the slip can be defined by Eq. (3-2) and in view of the assumption made in Section 3.1.1, the formulas that depict single corner model dynamics are defined by Eq. (3-3a) and Eq. (3-3b), respectively [11].

PUC-Rio - Certificação Digital Nº 1812727/CA

$$\lambda = \frac{v - \omega r}{v} \tag{3-2}$$

where ω and v are angular and longitudinal velocities, respectively; and r is the wheel radius.

$$J\dot{\omega} = rF_z\mu\left(\frac{v - \omega r}{v}\right) - T_b \tag{3-3a}$$

$$m\dot{v} = -F_z\mu\left(\frac{v - \omega r}{v}\right) \tag{3-3b}$$

where J is the wheel inertia; m is the proportional mass of the vehicle to which the wheel is subjected; T_b is the braking torque; F_x is the longitudinal force; F_z is the vertical load which the wheel is subjected; μ is the friction coefficient in which the value varies according to the longitudinal slip λ, already substituted, as defined by Eq. (3-2) and following the Burckhardt model expressed in Eq.(3-3a) and Eq. (3-3b).

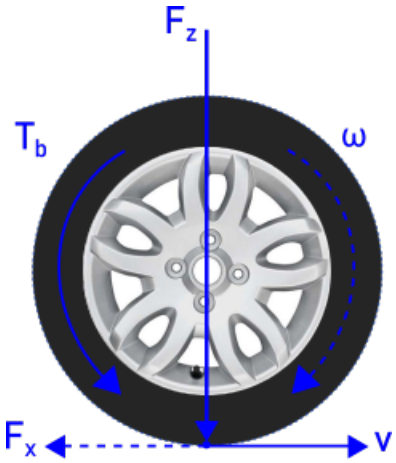

Figure 3.2: Single Corner Model [11].

PUC-Rio - Certificação Digital Nº 1812727/CA

3.1.3 Proportional-Integral-Derivative Controller for Antilock Braking System

Digital PID controllers are large-scale used in the industry due to their reliability, effectiveness and simplicity. Due to these benefits and the ease of being incorporated into already operational control meshes, the PID is researched in order to develop new design and tuning techniques (e.g. [129–131]) [132–134].

The basic description of a PID controller for the braking problem is shown in Eq. (3-4a) and Eq. (3-4b) [19]. Still, the representation in block diagrams of

a regular PID applied to the braking problem is provided in Figure 3.3 [19].

$$T_b(t) = K_p \left[e(t) + T_d \frac{de(t)}{dt} + \frac{1}{T_i} \int_0^t e(t)dt \right] \tag{3-4a}$$

$$e(t) = \lambda_{Ref}(t) - \lambda(t) \tag{3-4b}$$

where T_b is the output torque of the controller; K_p is the proportional gain; T_i is the integral time; T_d is the derivative time; $\lambda_{Ref}(t)$ and $\lambda(t)$ are the reference and obtained slip, respectively, so λ_{Ref} is the slip λ which yields the best friction coefficient in the Burckhardt curve, i.e. the greater friction coefficient in the curves depicted Figure 3.1. All terms represented at the instant t.

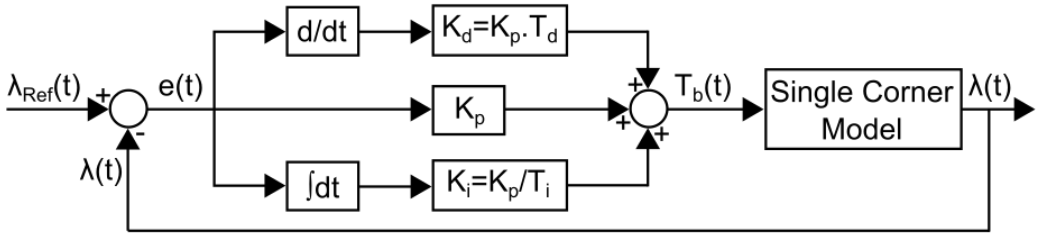

Figure 3.3: Block diagram of a regular PID applied to the ABS control.

However, the PID formulation may contain improvements such as [135]: i) derivative term filtering, which uses a low-pass filter to avoid larger derivative gains associated with high-frequency noises; ii) setpoint weight, used to avoid steady-state error and also decrease the overshoot when there is a setpoint change; and iii) anti-windup, which uses a saturation function of the actuator, seeking to avoid that the control output exceeds the actuator operating limit (i.e. if reaching the operation limit, the integrative term stops ceases to accumulate error incessantly). Thus, the digital PID version used is the same presented by Astrom [132], which provides the use of the previously mentioned features and is described as following:

$$P(t_k) = K_P \left(\beta \lambda_{Ref}(t_k) - \lambda(t_k) \right) \tag{3-5a}$$

$$D(t_k) = \frac{T_f}{T_f + t_{samp}} D(t_k - 1) - \frac{K_D}{T_f + t_{samp}} (\lambda(t_k) - \lambda(t_k - 1)) \tag{3-5b}$$

$$V(t_k) = P(t_k) + I(t_k - 1) + D(t_k) \tag{3-5c}$$

$$T_b(t_k) = sat(V(t_k), T_{b-low}, T_{b-high}) \tag{3-5d}$$

$$I(t_k) = I(t_k - 1) + t_{samp} K_I (\lambda_{Ref}(t_k) - \lambda(t_k) + \frac{t_{samp}}{T_t} (sat(V) - V) \tag{3-5e}$$

PUC-Rio - Certificação Digital Nº 1812727/CA

where t_k is discrete time; P, I and D are the proportional, integral and derivative terms, respectively; t_{samp} is the sampling time at which PID operates; V is the non-saturated output of the controller; sat is the actuator saturation function; T_{b-low} and T_{b-high} are the upper and lower operating limits of the actuator, being T_b the saturated braking torque; β is the setpoint weighting term, which was not used since an unitary value ($\beta = 1$) was considered; T_f is the filtering time, considered unitary ($T_f = 1$), as this value already guarantees a $\frac{T_f}{T_f + t_{samp}} \neq 0$ and < 1 mangnitude, ensuring a stable difference equation; and, finally, T_t is the anti-windup term, also considered unitary ($T_t = 1$), in order to multiply the difference between PID original (V) and saturated ($sat(V)$) outputs by the PID sampling time t_{samp} in the integral update [132].

In short, the actuator saturation function works in a way that torque T_b will be: i) T_{b-low}, when the output torque V is less than the lower limit of actuator operation; ii) T_{b-high}, when the output torque V is greater than the upper limit of actuator operation; and, iii) V: when the output torque V is within the operating limits of the actuator.

PUC-Rio - Certificação Digital Nº 1812727/CA

3.2 Contributions

The ABS literature review present in Section 1.3 revealed that: i) the use of a single-wheel model is sufficient and often used for the ABS development; ii) the majority of ABS controllers present a high complexity, thus the use of controllers easier to implement is fostered; iii) the use of traditional AI optimization methods or neural networks are widely used for tuning PID-based controllers, being interesting, thus, the use of new AI optimization methods for this purpose.

Still, as introduced in Section 1.5.1, the present work proposed as main contributions: i) a new multi-objective formulation for PID-controlled ABS tuning, which was introduced in Section 3.2.1; ii) a novel MSSA version considering opposite based learning initialization (OBLI-MSSA), which is presented in Section 3.2.2; and iii) in order to follow the state of the art evolution, the performance comparison between recent (i.e. MODA, MSSA and OBLI-MSSA) and traditional (i.e. NSGA-II) AI optimization techniques. The final contribution is carried out in the Section 3.3, being made through S, ED and HV comparison metrics, depicted in Section 2.3.

3.2.1
Multi-objective Tuning of a PID Controller

From the digital PID formulation described in Section 3.1.3, the multi-objective optimization of proportional K_P, integral K_I and derivative K_D gains is sought for yielding optimal relations among performance and comfort. Therefore, the metrics used to evaluate the performance [136] and comfort [22] of each solution x are respectively listed below:

$$f_1(x) = \Delta S \tag{3-6a}$$

$$f_2(x) = \frac{\sqrt{\sum_1^{N-1}\left[\left(a_{Ref_{t+1}} - a_{t+1}\right) - \left(a_{Ref_t} - a_t\right)\right]^2}}{\Delta S} \tag{3-6b}$$

where ΔS is the total distance required for the wheel to reach longitudinal velocity v equal to zero; N is the maximum number of measurements estimated or performed by the sensor; a_t and a_{Ref_t} are the current (i.e. at the instant t) and reference acceleration, respectively, being a_{t+1} and $a_{Ref_{t+1}}$ are the same for the instant $t+1$. Generally, for the same surface condition the reference acceleration a_{Ref} is constant, being equal in moments t and $t+1$, but for simulations where braking does not start at the initial moment, this distinction is necessary to not generate and accumulate errors. Also, the reference slip λ_{Ref} that yields the best friction coefficient μ_{Ref} is defined through Burckhardt curve (Eq. (3-1)). Therefore, a_{Ref} is identified by $\dot{v}$ value when replacing λ_{Ref} and μ_{Ref} with $\lambda = \frac{v-\omega r}{v}$ and μ in Eq. (3-3a) [11].

Being n the number of decision variables for the ABS multi-objective optimization, the regular MO statement ($n = 3$) for the ABS problem is similar to that described in Eq. (2-2), but with i) $F(x) = [f_1(x), f_2(x)]$; and ii) no equality and inequality constrains (Eq. (3-7)).

However, another ABS multi-objective optimization approach is proposed by considering a PID gain scheduling. In essence, the idea is to divide the PID action into 5 regions according to the slip value presented at the instant t_k, seeking to obtain a specific control law that presents a better performance for each region. The control sections are divided into four equally spaced regions from zero to the ideal slip value, and other region comprising the remaining slip values, until the unitary slip, bringing together five control laws. Thus, each solution becomes a $n = 15$ decision variables vector, every three being the set of gains for one PID controller. For a better comprehension, the MO statement for the novel approach is shown in Eq. (3-8).

PUC-Rio - Certificação Digital Nº 1812727/CA

$$\begin{cases} min \;\; F(x) = [f_1(x), f_2(x)]\,; \;\; x \in R^n \\ s.t. \\ \qquad K^l \leq x \leq K^u \end{cases} \tag{3-7}$$

$$\begin{cases} min \;\; F(x) = [f_1(x), f_2(x)]\,; \;\; x \in R^{15} \\ s.t. \\ \qquad K_P^l \leq x(3r-2) \leq K_P^u \qquad\qquad r = 1, \ldots, 5. \\ \qquad K_I^l \leq x(3r-1) \leq K_I^u \\ \qquad K_D^l \leq x(3r) \leq K_D^u \end{cases} \tag{3-8}$$

where $F(x)$ is the objective vector; x is the solution vector with n decision variables, where each one represents a proportional, integral or derivative PID gain; $f_1(x)$ and $f_2(x)$ represent performance and comfort objectives, respectively; r indicates the region to which the current slip belongs; and, finally, K^l and K^u are the lower and upper bounds vectors of the solution x, also with n terms representing proportional, integral or derivative bounds. The upper and lower bounds of each variable were defined by running optimization rounds with arbitrary boundary values until the PF yielded did not present solutions with threshold variables.

PUC-Rio - Certificação Digital Nº 1812727/CA

3.2.2 Opposite Based Learning Initialization for Multi-Objective Salp Swarm Algorithm

The opposition based learning (OBL) is a research field which has drawn a lot of attention in the last decade for having been used to improve reinforcement learning, ANN, GA, PSO, ACO, ABC and others soft computing algorithms [137].

Inspired by the opposition concept existing in different knowledge areas and cultures, the OBL is defined as [137]:

Definition 1. Opposite number: Being $x \in [a, b]$ a real number. The opposite number of x is defined as follows:

$$\breve{x} = a + b - x \tag{3-9}$$

However, the concept of opposite number may be extended for n-dimensional space, as presented below:

Definition 2. Opposite point in n space: Being $x(x_1, ...x_n)$ a point in n space and $x_i \in [a_i, b_i]$, $i = 1, 2, ..., n$. The opposite number of x is defined as follow:

$$\breve{x}_i = a_i + b_i - x_i \tag{3-10}$$

For AI optimization techniques, the OBL brought promising results in accelerating the solutions convergence on PF, when considering an opposition based initial population, an idea conceived primarily by Rahnamayan, Tizhoosh and Salama [138]. Posteriorly, Ventresca and Tizhoosh brought a mathematical proof in their work [139] that a population consisting of half of random solutions and half of opposing solutions has a greater variability than an entire population of random individuals.

Therefore, in order to obtain a new MSSA version with a more diverse initial population and greater solutions convergence potential on the PF, two approaches were considered: in a first moment i) a traditional MSSA with a 'half random-half opposite solutions' initial population, known as OBLI-MSSA; and, in a second moment ii) a MSSA with a P-sized initial population, selected from the fittest solutions in a P-sized whole random population and its opposite one (i.e. also with size P). For this purpose, the selection process is the same used by the NSGA-II, which was described in Section 2.4.2. Also, the same dynamic of 'opposite population generation' and 'selection process' occurs during the iterations, according to a 'jumping rate' (JR) probability, on top of the current population [138]. The OBLI-MASS built by the second approach are identified by the presence of the jumping rate next to the OBLI-MSSA acronym.

PUC-Rio - Certificação Digital Nº 1812727/CA

3.3 Results

The PID-controlled single corner model was built using wheel radius r, inertia J and proportional mass m of $0.30m$, $1.00kg.m^2$ and $225.00kg$, respectively, as the case study depicted in the Savaresi and Tanelli book [11]. The simulated maneuver consists of reducing the vehicle speed from $55.55m/s$ ($200km/h$) to $8.33m/s$ ($30km/h$) [22], on a dry asphalt road condition. The simulation time stipulated was $6s$, with the braking event starting at the second first and a sampling time set to $0.01s$. In addition, solutions that do not reached the desired final velocity ($\leq 8.33m/s$) at the end of the simulation were penalized, being associated with them a fit value of 10^9 for both objectives. For dry asphalt road condition, the Burckhardt parameters are $v_{r1} = 1.28$, $v_{r2} = 23.99$ and $v_{r3} = 0.52$, as depicted in Table 3.1 [11], which makes the greater friction coefficient occur when the slip is $\lambda = 0.17$. Also, a saturation limit of $1000Nm$ was considered for the actuator.

Regarding the MO algorithms configuration, for all methods the population size $P = 100$ and the maximum number of $iter_{max} = 500$ iterations were defined, i.e. 50,000 maximum number of evaluations, being executed 30 opti-

mization rounds for statistical comparison. Still, the search space was delimited by $K_P = [200, 4000]$, $K_I = [1000, 9000]$ and $K_D = [0, 1500]$. For the NSGA-II, crossover and mutation rate were defined as $PC = 0.9$ and $PM = 1/15$, respectively, the same parameter values used by Deb et al. [8] in the tested problems. About MODA, the inertia weight was defined by Eq. (3-11) and the food atraction weight was considered a random number in between zero and two, with standard uniform distribution. Also, separation s, alignment a, cohesion c and enemy distraction e weights were considered the same value y (Eq. (3-12a)), presenting a different behavior after 75% of the iterations (Eq. (3-12b)).

$$w = 0.9 - iter\left(\frac{0.7}{iter_{max}}\right) \tag{3-11}$$

$$y = 0.1 - iter\left(\frac{0.1}{\frac{iter_{max}}{2}}\right) \tag{3-12a}$$

$$y = 0.1 - \left(\frac{0.1}{\frac{iter_{max}}{2}}\right) \tag{3-12b}$$

where $iter$ is the current iteration.

As introduced in Section 3.2.1, two different PID approaches were considered. The first approach evaluated a conventional PID ($n = 3$), in which the multi-objective optimization yielded 2226, 3000, 2839 and 3000 unique solutions for MODA, MSSA, NSGA-II and OBLI-MSSA, respectively, after 30 optimization rounds. The solutions were concatenated and ranked according to Pareto dominance, generating the Figure 3.4 from the first rank solutions obtained by each method, totalizing 965, 1023, 1389 and 1104 Pareto optimal solutions, following the same order presented above. However, results found did not facilitate the methods comparison, since they reached to very close PFs, mainly for MODA and NSGA-II methods, with practically overlapping PFs.

The second approach considered a gain scheduling PID with five slip regions: four equally spaced sections, from zero to the ideal slip $\lambda = 0.17$ and another region comprising the rest of the values, until the unitary slip, totalling 5 different control laws, i.e. a solution vector x with $n = 15$ decision variables. Thus, to select the control law, at each iteration t_k the current slip is evaluated, so that the region to which the current slip belongs is recognized, causing the controller to use the gains defined for that specific region. After 30 optimization rounds, MODA, MSSA, NSGA-II and OBLI-MSSA provided 2307, 2999, 2847 and 2999 unique solutions, 155, 391, 115 and 415 of them

PUC-Rio - Certificação Digital Nº 1812727/CA

being first rank solutions, respectively, as shown in Figure 3.5.

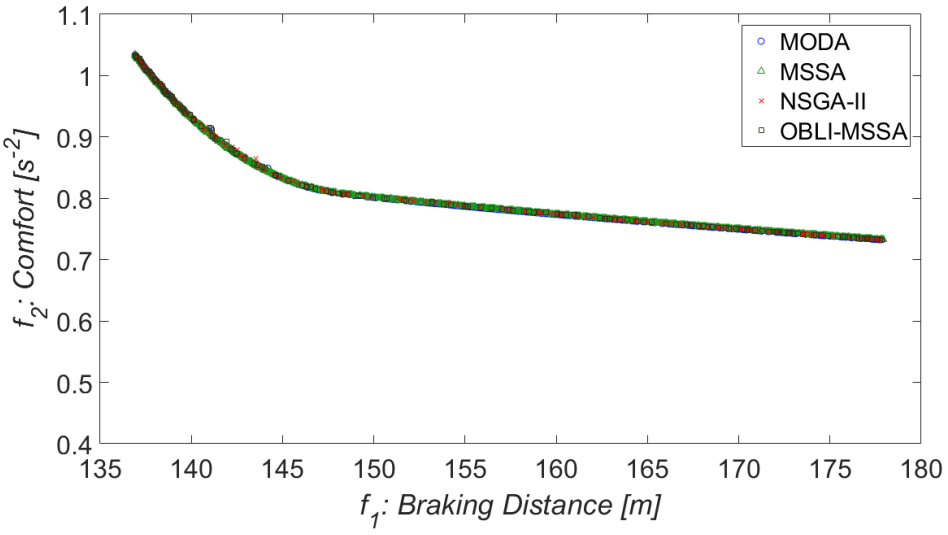

Figure 3.4: Pareto Front for each optimization algorithm, considering the PID-controller case with 3 decision variables.

PUC-Rio - Certificação Digital Nº 1812727/CA

Therefore, the Figure 3.5 indicates that the second control architecture yields better performance (f_1) and comfort (f_2) values. Also, the AI optimization methods present PFs with greater distinction among them, for that reason, the comparison between the MO algorithms is made on top of the second control architecture.

Seeking to compare the dominance of the solutions that constitute the Pareto Front of each method, the Table 3.2 was generated.

Table 3.2: Pareto Front Percentage of Domination for 15 Variables.

Algorithm		MODA	MSSA	NSGA-II	OBLI-MSSA
MODA	$\succ$	-	16.62	65.22	8.19
MSSA	$\succ$	72.90	-	61.74	25.06
NSGA-II	$\succ$	47.74	25.58	-	7.71
OBLI-MSSA	$\succ$	**84.52**	**55.50**	**80.00**	-

The results presented in Table 3.2 show that the PF obtained by OBLI-MSSA dominates more than is dominated the other PFs, as indicated by the highlighted results, suggesting a better performance of OBLI-MSSA. Following the OBLI-MSSA, MSSA dominates more than is dominated by MODA, NSGA-II. In addition, MSSA has greater dominance over OBLI-MSSA when compared with the other methods. In the third place in the dominance ranking is MODA algorithm, that more dominates than is dominated by the NSGA-II, its direct competitor, besides, also dominate a higher percentage of OBLI-MSSA, than the NSGA-II.

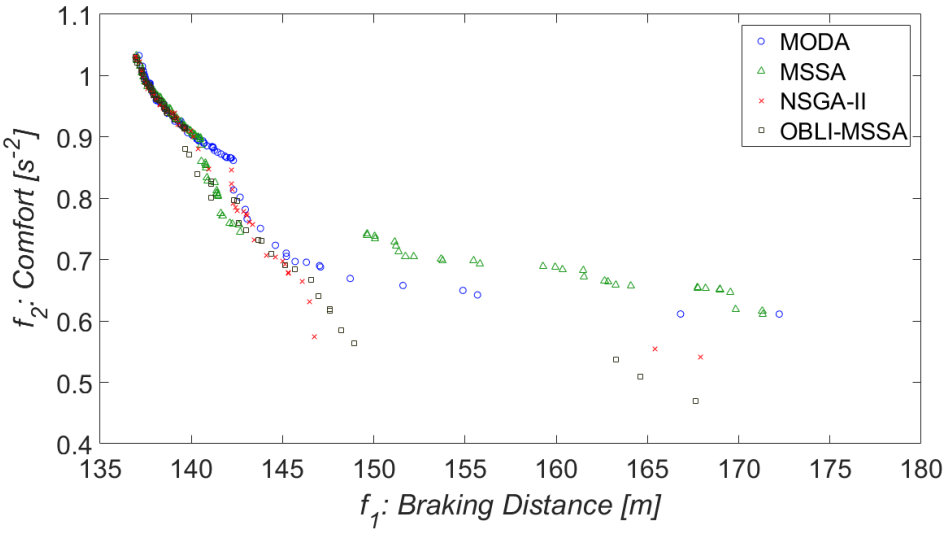

Figure 3.5: Pareto Front for each optimization algorithm, considering the PID-controller case with 15 decision variables.

Spacing (S), Euclidian Distance (ED) and Hypervolume (HV) results were also generated, being calculated base on PF solutions normalized between $[0, 1]$, for the comparison of metrics with different amplitudes. The results obtainded constitute the content of Table 3.3, 3.4 and 3.5, where 'Min' and 'Max' refer to the minimum and maximum values found in all 30 optimization rounds for the metric in question, while 'Mean' and 'Std' refers to the mean and standard deviation of these same values. Also, the best result of 'Min', 'Max', 'Mean' and 'Std' found for each comparison metric is highlighted.

Table 3.3: Spacing Metric Values for 15 Variables.

Algorithm	Min	Max	Mean	Std
MODA	9.3944E-03	1.0401E-01	3.7728E-02	2.2450E-02
MSSA	2.7521E-03	3.3467E-02	8.6814E-03	7.2686E-03
NSGA-II	3.3208E-03	**8.1461E-03**	6.9186E-03	**9.7306E-04**
OBLI-MSSA	**1.2677E-03**	1.0440E-02	**5.0483E-03**	2.4747E-03

According to Table 3.3, OBLI-MSSA presented the best average, with a result 27.03% lower than that obtained by the NSGA-II, the second lowest average. However, NSGA-II has a standard deviation of 60.68% lower than that obtained by OBLI-MSSA. Therefore, despite the best result of OBLI-MSSA, the constancy of NSGA-II regarding the distribution of solutions in PF is attractive. Following the mentioned methods is MSSA, which obtained a mean and standard deviation better than MODA, the worst method for Spacing.

PUC-Rio - Certificação Digital Nº 1812727/CA

Table 3.4: Euclidean Distance Metric Values for 15 Variables.

Algorithm	Min	Max	Mean	Std
MODA	7.0239E-01	8.1552E-01	7.5419E-01	3.0757E-02
MSSA	**6.5352E-01**	7.8156E-01	7.2929E-01	3.0475E-02
NSGA-II	7.7599E-01	8.8934E-01	8.6617E-01	**2.3143E-02**
OBLI-MSSA	6.5441E-01	**7.7184E-01**	**7.2413E-01**	3.1796E-02

Table 3.5: Hypervolume Metric Values for 15 Variables.

Algorithm	Min	Max	Mean	Std
MODA	3.2643E-01	5.7018E-01	4.4855E-01	6.7977E-02
MSSA	**5.0790E-01**	**6.8297E-01**	**5.8159E-01**	4.3971E-02
NSGA-II	3.9197E-01	5.1235E-01	4.0425E-01	**2.6100E-02**
OBLI-MSSA	4.0354E-01	6.8116E-01	5.7544E-01	5.7739E-02

From Table 3.4, OBLI-MSSA yielded the best performance in terms of mean, with 0.70% lower result than the second lower average, presented by MSSA. Although standard deviation, minimum value and maximum value presented by OBLI-MSSA are 4.33% higher, 0.14% higher and 0.12% lower than the values found for MSSA, the two algorithms are considered tied. Finally, MODA appears ahead of the NSGA-II, for having mean, minimum and maximum of 12.93%, 9.48% and 8.30% lower than presented by NSGA-II, respectively. Again, NSGA-II was the most consistent method, with a standard deviation of 24.06% lower than the second best standard deviation, obtained by MSSA.

From Table 3.5, in the study of Hypervolume, MSSA achieved the best result with mean 1.07% higher and standard deviation 23.84% lower than the OBLI-MSSA. In second place, OBLI-MSSA appears with mean 28.29% higher and standard deviation 15.06% lower than the third highest average, presented by MODA, that grant an average 10.96% higher than the lower average, obtained by the NSGA-II. Even with NSGA-II having a standard deviation of 61.60% lower than MODA and with a minimum of 20.07% higher, which allows NSGA-II to still be able to be better than MODA, its result is considered worse, since it is more constant on a worse average.

Despite the lower HV than the MSSA, OBLI-MSSA PF yield greater dominance, achieved the best results in the spacing, ensuring the best distribution of solutions on the PF, among the tested methods, and also obtained good results on euclidean distance and hypervolume metrics. For these reasons, the OBLI-MSSA was considered the best optimization method.

Thus, one solution was arbitrarily selected from the OBLI-MSSA PF as the best solution. The solution is represented in Table 3.6, while the Figure 3.6 provides the speed reduction and braking torque behavior, showing that the

PUC-Rio - Certificação Digital Nº 1812727/CA

selected solution is valid, fulfilling the objective to brake at $55.55m/s$ to reach $8.33m/s$, before the simulation time of $6s$ ends.

Table 3.6: Simulated Solutions for OBLI-MSSA.

Solution	Gain	Slip Region				
		1	2	3	4	5
Selected	K_P	2524.84	3999.67	3945.94	3869.18	3999.88
	K_I	9000.00	9000.00	8528.55	6600.07	2180.22
	K_D	0.00	58.56	0.00	1499.96	289.30

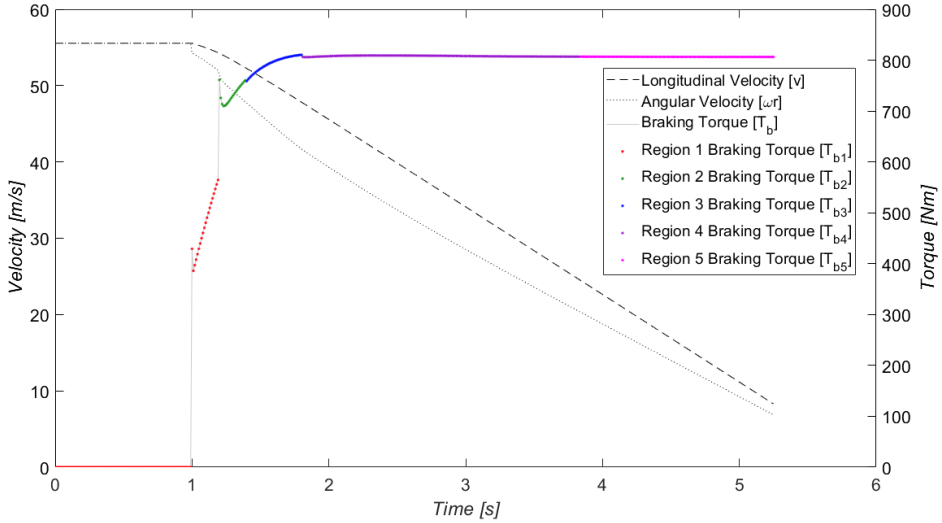

Figure 3.6: Braking torque, longitudinal and angular velocity for OBLI-MSSA solution selected as the best ($f_1(x) = 138.5370$; $f_2(x) = 0,9449$).

Finally, since MSSA yields a better performance when compared to MODA and NSGA-II, the OBL initialization presented in Section 3.2.2 was applied to MSSA. However, a second OBLI-MSSA architecture was tested, also considering the description made in Section 3.2.2. Thus, the Fig 3.7 is introduced for the comparison of the first architecture OBLI-MSSA and the second one, having in this last case a jumping rate varied among $JR = 10\%, 20\%, 30\%$ and 40%.

The Fig 3.7 indicates that the first architecture OBLI-MSSA yields a better performance, since its PF practically dominates all others.

3.4 Conclusions

The present contribution proposed the improvement of a PID-based ABS. Seeking to achieve this goal, NSGA-II, MODA, MSSA and a new proposed

PUC-Rio - Certificação Digital Nº 1812727/CA

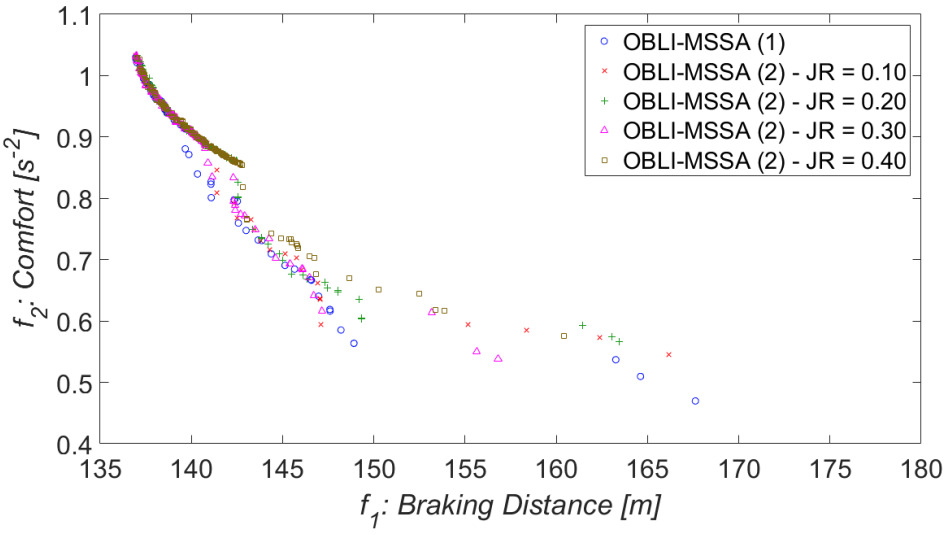

Figure 3.7: OBLI-MSSA versions comparison.

MSSA version, known as OBLI-MSSA, were applied for the PID tuning, being proposed a new multi-objective formulation for PID-controlled ABS tuning, a novel MSSA version considering opposite based learning initialization (OBLI-MSSA) and the comparison of recent (i.e. MODA, MSSA and OBLI-MSSA) and traditional (i.e. NSGA-II) AI optimization techniques.

The braking problem with 3 variables did not allow the methods comparison, due to the proximity of results obtained by the methods. For the braking problem with 15 variables, OBLI-MSSA was considered the best optimization method, because its PF yield greater dominance and achieved spacing results that ensure the best distribution of solutions on the PF, among the studied methods. Additionally, OBLI-MSSA obtained good results on the other two metrics, euclidean distance and hypervolume. The MSSA was considered to have a better performance than MODA and NSGA-II, also due to the dominance and the good results obtained in the S, ED and HV metrics. In terms of performance, MODA presented a similar behavior to NSGA-II. However, NSGA-II was the most constant method in all metrics, because it presented the smallest standard deviations in all comparison metrics.

For future works it would be interesting to: i) introduce other PID parameters as optimization decision variables, e.g. setpoint weighting term β, filtering time T_f and anti-windup term T_t, as well as the slip values that limit the gain scheduling regions; ii) evaluate if the proposed initialization method for the OBLI-MSSA would yield better results, if applied to MODA and NSGA-II; iii) seek ways to guarantee the same results stability obtained by NSGA-II in other meta-heuristics, maybe making a fine tuning of the MO

PUC-Rio - Certificação Digital Nº 1812727/CA

parameters; and iv) consider other controllers for the braking problem, e.g. a fuzzy-PID [19] or fractional order fuzzy PID [140], evaluating whether these controllers outperform the gain schedulling PID.

PUC-Rio - Certificação Digital Nº 1812727/CA

4
Multi-Objective Optimization of Heat Exchanger Design Through Heuristic Kalman Algorithm

The present section seeks to depict the second work: 'Multi-Objective Optimization of Heat Exchanger Design Through Heuristic Kalman Algorithm' [110]. Therefore, this section is divided as follows: Section 4.1 introduces the heat problem; Section 4.2 reviews the main contributions related to the second work, presenting the methodology associated with them; Section 4.3 addresses the obtained results; and, finally Section 4.4 summarizes the conclusions drawn and future work suggestions.

4.1
Problem Description

Heat exchangers are used for transfering heat between fluids or fluid and solid materials, being widely used in diverse industrial processes [23], which makes their effectiveness directly impact efficiency of other processes to which they are applied. For this purpose, the optimization of heat exchangers design draws the attention of the scientific community, highlighting the multi-objective approach, considering performance and cost metrics, as seen in Section 1.3.

Thus, the present section is divided in four parts: Section 4.1.1 introduces the objective functions used to evaluate the solutions; Sections 4.1.2 and 4.1.3 depict both plate-fin (PFHE) and shell-tube (STHE) heat exchangers design, respectively; and, Section 4.1.4 presents the ZDT1 description, a well known multi-objective optimization benchmark problem. Still, for a better comprehension, the Figures 4.2 and 4.2 presents a schematic representation of both PFHE (left) and STHE (right), respectively [12, 141].

PUC-Rio - Certificação Digital Nº 1812727/CA

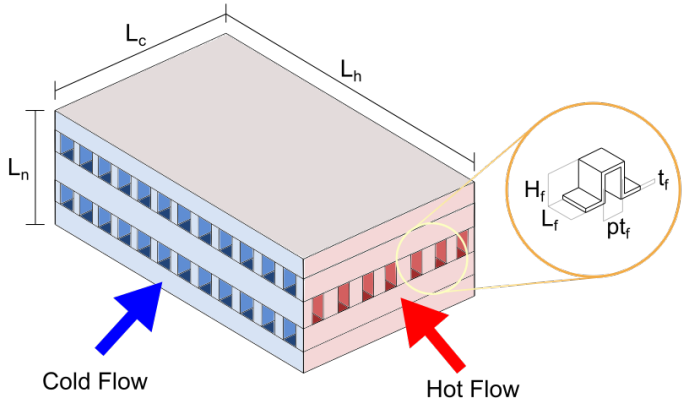

Figure 4.1: Schematic representation of a plate-fin heat exchanger [12].

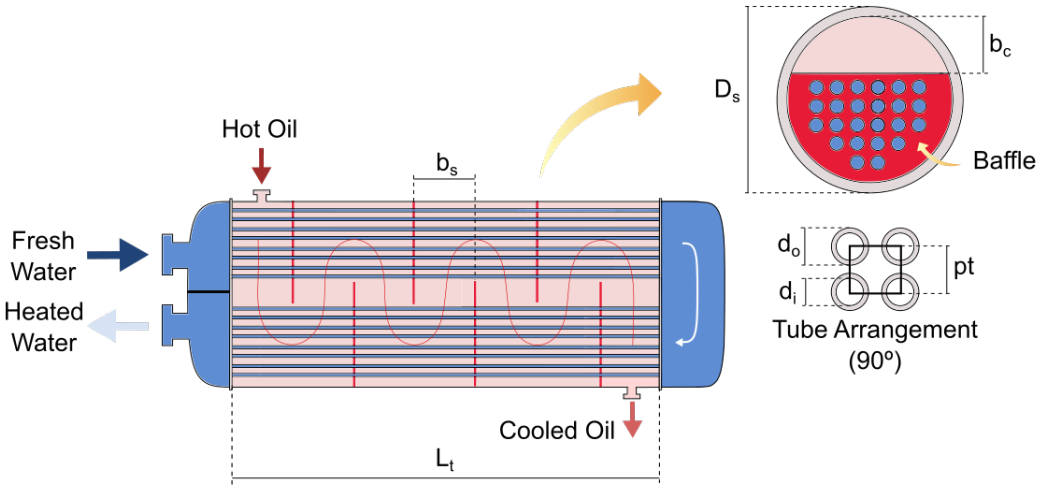

Figure 4.2: Schematic representation of a shell-tube heat exchanger [13].

4.1.1 Objective Functions

For the heat exchanger multi-objective optimization, two objectives are proposed: i) the maximization of the effectiveness ϵ; and ii) the minimization of the total annual cost C_{tot}. Considering a minimization configuration, the first

PUC-Rio - Certificação Digital Nº 1812727/CA

and second objectives, respectively f_1 and f_2, are described as follows [142]:

$$f_1(x) = 1/\left(1 + \epsilon(x)\right) \tag{4-1a}$$

$$f_2(x) = C_{tot}(x) \tag{4-1b}$$

where x is the solution, constituted of n decision variables and ϵ is effectiveness of the heat exchanger.

The effectiveness and total annual cost are distinctly calculated for PFHE and STHE. Therefore, the PFHE effectiveness and total cost are depicted in Eq. (4-2) and Eq. (4-3a)-(4-3d) [12], respectively. While, for STHE, the effectiveness and total cost are described in Eq. (4-4) and Eq. (4-5a)-(4-5d) [141], respectively.

$$\epsilon = 1 - \left[\exp\left(-NTU(1+C^*)\right)\left[\begin{array}{c} I_0(2NTU\sqrt{C^*}) + I_1\sqrt{C^*}(2NTU\sqrt{C^*}) \\ -\frac{1-C^*}{C^*}\sum_{n=2}^{\infty} I_n {C^*}^{\frac{n}{2}}(2NTU\sqrt{C^*}) \end{array}\right]\right] \tag{4-2}$$

where I is the modified Bessel function; NTU is the 'number of transfer units'; and C^* is the heat capacity ratio. Still, Sanaye and Hajabdollahi [12] and Kays and London [143] present a more detailed mathematical formulation of the above equation.

$$C_{tot} = aC_{in} + C_{op} \tag{4-3a}$$

$$C_{in} = C_A A_{tot}^n \tag{4-3b}$$

$$C_{op} = \left(k_{el}\tau\frac{\Delta PV}{\ni}\right)_c + \left(k_{el}\tau\frac{\Delta PV}{\ni}\right)_h \tag{4-3c}$$

$$a = \frac{r}{1-(1+r)^{-q}} \tag{4-3d}$$

where C_{in} and C_{op} are PFHE design and operating cost ($), respectively; a is the annual cost; subscripts c and h refer to cold and hot fluid, respectively; A_{tot}, C_A and n are the total heat transfer area (m^2), cost per unit of surface area ($/$m^2$) and a constant, respectively; k_{el} is the electricity price ($/MWh); τ is the heat exchanger operation hours per year (h/year); ΔP, V and $\ni$ are the pressure drop (kPa), volume flow rate (m^3/s) and compressor efficiency, respectively; and, finally r and q are the interest rate and depreciation time (years), respectively. Again, other details of the previous calculation may be found in [12].

PUC-Rio - Certificação Digital Nº 1812727/CA

$$\epsilon = \frac{2}{(1+C^*) + (1+C^{*2})\coth\left(\frac{NTU}{2}(1+C_*^2)^{0.5}\right)} \tag{4-4}$$

$$C_{tot} = C_{inv} + C_{op} \tag{4-5a}$$

$$C_{inv} = 8500 + 409A_{tot}^{0.85} \tag{4-5b}$$

$$C_{op} = \sum_{t=1}^{ny} \frac{C_o}{(1+i)^t} \tag{4-5c}$$

$$C_o = \frac{k_{el}\tau}{\ni_p}\left((\Delta PV)_t + (\Delta PV)_s\right) \tag{4-5d}$$

where C_{inv} and C_{op} are the STHE design and total operating cost ($), respectively; A_t is the heat transfer area (m^2); ny is the device lifespan (years); i is the annual discount rate; t is the depreciation time (years); C_o is the annual operating cost ($); subscripts t and s refer to tube and shell part, respectively; and, finally $\ni_p$ is the pump efficiency. Still, other details of the previous calculation may be found in [141].

PUC-Rio - Certificação Digital Nº 1812727/CA

4.1.2 Plate-fin heat exchanger

The PFHE design optimization problem considered a stainless steel material with thermal conductivity $k_w = 18$ W/mK and operating temperature condition of 620 K, pressure of 180 kPa and mass flow rate of 1.45 kg/s on the hot side, and operating temperature condition of 315 K, pressure of 120 kPa and mass flow rate of 1.35 kg/s for the cold side. Other constant values indispensable for the objective functions evaluation as electrical energy price $k_{el} = 20$ $/MWh, price per unit of area $C_A = 90$ $/$m^2$, expoent which provides non-linear growth relative to the area $n = 0.6$, operation time $\tau = 5000$ h/yr, compressor efficiency $\ni = 0.6$, depreciation time $q = 10$ years and interest rate $r = 0.1\%$, as in Sanaye and Hajabdollahi [12].

In order to optimize the PFHE design, seven design variables with their respective bounds were considered, namely as hot fluid flow length ($L_h = [0.2, 0.4]$ m), cold fluid flow length ($L_c = [0.2, 0.4]$ m), no flow length ($L_n = [0.7, 1.2]$ m), fin thickness to fin length ratio ($t_f/L_f = [0.012, 0.048]$), fin height ($H_f = [0.0015, 0.0080]$ m), fin length ($L_f = [0.0020, 0.0035]$ m) and fin pitch ($pt_f = [0.0010, 0.0025]$ m). All parameter bounds proposed by Sanaye and Hajabdollahi [12].

4.1.3
Shell-Tube heat exchanger

The STHE design optimization problem is the same analyzed by Sanaye and Hajabdollahi [141] using GA, where is considered an oil cooler. In that case, the oil enters the shell with mass flow rate of 8.1 kg/s and temperature of 351.45 K (78.3°C), while fresh water enters in the tubes with mass flow rate of 12.5 kg/s and temperature of 303.15 K (30.0°C). It was considered a tube arrangement of 90° and, as in [141], electrical energy price k_{el} = 0.15 $/kW h, life period ny = 10 years, rate of annual discount i = 10%, operation time τ = 7500 h/yr and pump efficiency $\ni_p$= 0.6 to evaluate the objective functions.

In order to optimize STHE design, six design variables with their respective bounds were considered, namely as the inner tube diameter (d_i = $[0.0112, 0.0153]$ m), number of tubes ($N_t = [100, 600]$), length of tube (L_t = $[3, 8]$ m), tube pitch pt to outer tube diameter d_o ($pt/d_o = [1.25, 2.00]$), baffle cut ratio ($bc/D_s = [0.19, 0.32]$), baffle spacing ratio ($bs/D_s = [0.2, 2.4]$), where D_s is the shell diameter. All parameter bounds proposed by Sanaye and Hajabdollahi [141].s

4.1.4
Zitzler-Deb-Thiele Test Problem 1

The ZDT1 is a bi-objective optimization benchmark with a convex PF, which was proposed by Zitzler, Deb and Thiele in 2000 [14]. The ZDT1 stantard statement is depicted by Eq. (4-6), while Figure 4.3 presents the ZDT1 Pareto front [14].

$$\begin{cases} min \;\; F(x) = [f_1(x), f_2(x)] & x \in R^n \\ \quad\;\; f_1(x_1) = x_1 \\ \quad\;\; g_1(x_2, \ldots, x_n) = 1 + 9\sum_{i=2}^{n} \frac{x_i}{(m-1)} \\ \quad\;\; f_2(f_1, g) = 1 - \sqrt{\frac{f_1}{g}} \end{cases} \tag{4-6}$$

where n = 30 decision variables, being all of them comprehended between $[0, 1]$. Still, the optimization is performed until the maximum number of 25,000 evaluations, as suggest Zitzler, Deb and Thiele [14].

PUC-Rio - Certificação Digital Nº 1812727/CA

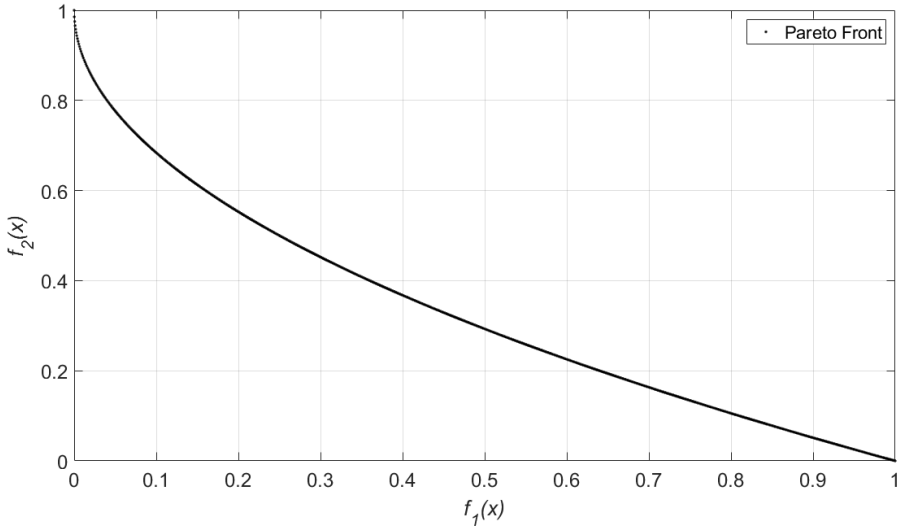

Figure 4.3: Real Pareto front for the ZDT1 problem [14]

PUC-Rio - Certificação Digital Nº 1812727/CA

4.2 Contributions

As introduced in Section 1.3, the use heat exchanger design problem for AI optimization methods comparison is a trend. Therefore, the present work proposed as main contributions: i) the MOHKA performance improvement; ii) the enhancement of MOHKA attractive features, i.e. the reduced number of parameters and simplified architecture; and iii) assess the impact of the slowdown coefficient α on the PF convergence, considering the typical values between [0.4,0.9] [62]. Thus, as seen in Section 1.5.2, seeking to accomplish these contributions, five new MOHKA versions were proposed based on i) randomness adition by N_ξ suppression; ii) a new architecture, where new populations are created only on top of the measure ξ; iii) the replacement of 'crowding distance' diversity preservation mechanism to 'niching procedure'.

However, for the development of these new MOHKA versions, only the niching procedure diversity preservation mechanism requires a more extensive explanation of its methodology, causing this section to be divided into two parts: Section 4.2.1, where the niching procedure is introduced; and Section 4.2.2, where the new MOHKA versions are described.

4.2.1 Niching Procedure Diversity Preservation Mechanism

The niching procedure diversity preservation mechanism was adapted to replace the crowding distance operator (Section 2.4.2) in some of the

new proposed MOHKA versions and may be dismembered in three steps, as follows [144]:

Step 1. (Determination of reference points on a hyper-plane): The reference points are placed in the objective space before the optimization process starts, being preferably supplied by the optimization designer. In case of lack of preference information, the reference points are allocated in a structured manner as in [144], where it is considered a systematic approach [145], which widely distributes H reference points on a $(M-1)$-dimentional normalized hyper-plane, where M is the number of objectives. Considering p niches or divisions, the number H of reference points is described by:

$$H = \binom{M+p-1}{p} \tag{4-7}$$

Step 2. (Adaptive normalization of population members): For each iteration, the population $X(k)$ is composed by N solutions, each one evaluated according to M objectives, generating the set $Z(k)$ of fit values $z_i^k = \left(z_{i,l}^k, ..., z_{i,M}^k\right)$, where i denotes the position of the fit value in the set $Z(k)$, l indicates the dimention of the i-th solution and k is the iteration number. This notation is maintained until the end of the present subsection. As the reference points are arranged in a normalized hyper-plane, the normalization of solution fit value z_i^k is important for the subsequent solution-reference point association, as well as contributing to a better Pareto front coverage [146]. In order to perform the normalization, for each objective the minimum value presented in the population (z_l^{min}) is identified and an ideal point is defined by $\bar{z}^k = (z_1^{min}, z_2^{min}, ..., z_M^{min})$. Then, the solutions of $Z(k)$ set are translated through the subtraction operation $z_{i,l}^{k'} = z_{i,l}^k - \bar{z}_l^k$ and an extreme point for each l objective, denoted by $z^{l,max}$, is identified by finding the solution which yields a minimum for the following scalarizing function [144]:

$$\zeta_l^k = \frac{z_i^{k'}}{w_l} \tag{4-8}$$

where ζ_l is the result of the scalarizing function when applied to the objective l of the translated solution $z_i^{k'}$, w_l is a weight vector close to the l-th objective axis, which can be contructed as follows $w_l = \mathbf{e}_l + \epsilon \sum_{h=1,h\neq l}^{M} \mathbf{e}_m$, being ϵ a tolerance, $\mathbf{e}_l$ a unit vector in the l-th axis direction and the same for $\mathbf{e}_m$.

The M extreme vectors defined among the $z^{l,max}$ points constitute a linear hyper-plane. For the distribution of Das and Dennis' reference points, the generated hyper-plane is coincident with that of the reference points and the interception of such hyper-plane and the l-th objective axis is a_l. Therefore, the fit values can be normalized, as follows [144]:

PUC-Rio - Certificação Digital Nº 1812727/CA

$$z^k_{i_{norm}} = \frac{z^{k'}_i}{a_l} \tag{4-9}$$

Step 3. (Association Operation): In the third step, reference lines are drawn considering the reference points and the objective space origin. Then, the perpendicular distance among the reference lines and each solution is calculated. Therefore, the solutions are associated with the reference point whose reference line yields the shortest distance to the solution. Thereby, it is counted the number ρ_h of solutions associated to each niche or reference point h and for solutions with same non-dominated rank, the preference is given to the ones associated to niches with lower ρ_h.

4.2.2 Proposed Multi-Objective Heuristic Kalman Algorithm Versions

Based on the conventional MOHKA structure described in Section 2.4.5 and on the niching procedure introduced in Section 4.2.1, five MOHKA versions were proposed. Seeking to name these novel versions, the subscripts 'r', 'p' and 'n' were used to indicate the use of the following features: randomness addition, simplified architecture and the use of niching procedure diversity preservation mechanism. Therefore, the proposed MOHKA versions are introduced below, indicating the difference in them to the conventional MOHKA.

Version 1. (Multiobjective heuristic Kalman algorithm random - MOHKAr): this version has the same architecture as the MOHKA, but the measure ξ becomes a randomly selected solution from the archive, no longer being an average of the N_ξ best solutions;

Version 2. (Multi-objective heuristic Kalman algorithm random proportional - MOHKArp): this version selects the measure ξ in the same way as MOHKAr. However, the mean and standard deviation of the solution selected as ξ is used to initialize a new population in the next iteration, making the iterations have a fixed/proportional number of evaluations;

Version 3. (Multi-objective heuristic Kalman algorithm niching - MOHKAn): similar to MOHKA, but with the adapted niching procedure being used at the locus of the crowding distance operator as diversity preservation mechanism;

Version 4. (Multi-objective heuristic Kalman algorithm random niching - MOHKArn): similar to MOHKAr, but with the crowding distance operator replaced by the niching procedure as the diversity preservation mechanism;

Version 5. (Multi-objective heuristic Kalman algorithm random proportional niching - MOHKArpn): similar to MOHKArp version, but with the niching procedure replacing the crowding distance operator as the diversity

PUC-Rio - Certificação Digital Nº 1812727/CA

preservation mechanism.

4.3 Results

In order to study the MOHKA parameter α and the MOHKA versions proposed above, the PFHE, STHE and ZDT1 problems are considered.

For the ZDT1 problem it was considered a maximum number of 25,000 evaluations, as suggest Zitzler, Deb and Thiele [14]. In order to study the ideal number of evaluation for PFHE and STHE problems it was performed 13 optimization tests using MOHKAr, which considered 30 rounds of optimization, $\alpha = 0.40$, a population size of $N = 100$, $N_\xi = 6$ and a maximum evaluations number of 1,000, 2,000, 3,000, 4,000, 5,000, 7,500, 10,000, 20,000, 30,000, 40,000, 50,000, 75,000, 100,000, respectively. As Figures 4.4 and 4.5 indicates, when the number o evaluations reached 30,000 there is practically no improvement in the generated PF for PFHE (left) and STHE (right) problems. Therefore, for all numbered tests proposed next the population size N, the maximum number of evaluation $Eval_{max}$ and the number of optimization rounds n_{seeds} were fixed in 100, 30,000 and 30, respectively, also it was considered $N_\xi = 6$ for all MOHKA N_ξ-dependent versions.

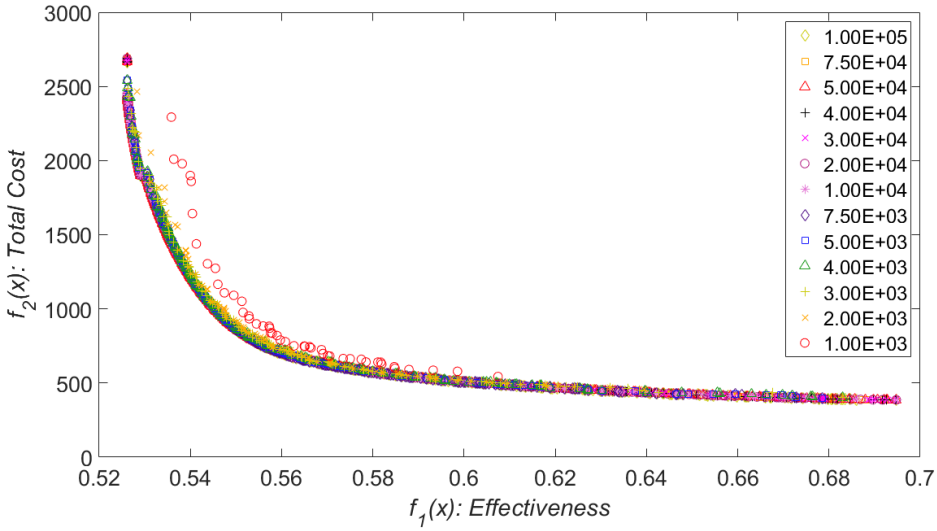

Figure 4.4: Iteration study for PFHE problem.

For all studied methods the slowdown coefficient α was varied according to the typical values 0.40, 0.50, 0.60, 0.70, 0.80 and 0.90 [62]. To facilitate the representation of results in tables, each combination of method and α value was called a test and numbered from 1 to 6 for the original MOHKA, 7 to 12 for MOHKAr, 13 to 18 for MOHKArp, 19 to 24 for MOHKAn, 25 to 30

PUC-Rio - Certificação Digital Nº 1812727/CA

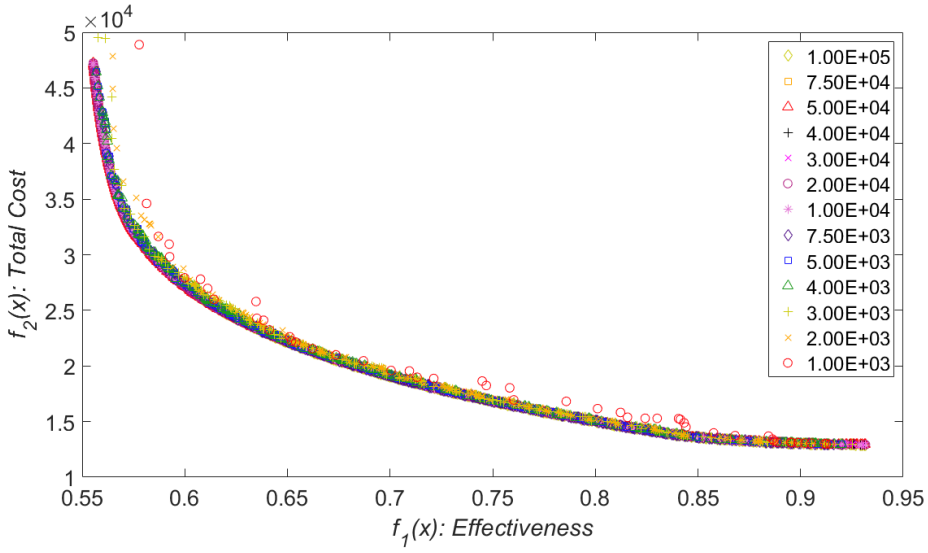

Figure 4.5: Iteration study for STHE problem.

for MOHKArn and 31 to 36 for MOHKArpn, totalizing 108 tests considering PFHE, STHE and ZDT1 problems.

Seeking to produce a PF said real, the tests from 1 to 36 were concatenated, totalling 20998 and 20997 unique solutions that generated 2452 and 1741 nondominated solution for the PFHE and STHE problems, respectively. After producing the PF said real, the tests from 1 to 36 were redone to obtain the IGD values, allowing to evaluate the IGD_a and IGD_f metrics.

Table 4.1 presents the means and standard deviations ('Std') for all methods from 1 to 36, when applied to the PFHE, STHE and ZDT1 problems. The best IGD_a and IGD_f means for each method are highlighted in bold

Seeking to assess if any of the new versions of MOHKA brought performance increase, the comparison metrics IGD_a and IGD_f are evaluated for each method and optimization round, following the methodology introduced in Section 2.3. Therefore, the violin plots for the IGD_a and IGD_f metrics obtained for all methods are presented in the Figures 4.6, 4.7 and 4.8, where the red and black points represent their mean and median, respectively. Also, the Table 4.2 was generated with the median obtained by each metric and method for the problems of PFHE, STHE and ZDT1, respectively, being the best IGD_a and IGD_f for each problem and MOHKA version highlighted in bold.

PUC-Rio - Certificação Digital Nº 1812727/CA

Table 4.1: Mean and standard deviation of the IGD_a and IGD_f metrics for all methods and problems.

	PFHE - IGD_a		PFHE - IGD_f		STHE - IGD_a		STHE - IGD_f		ZDT1 - IGD_a		ZDT1 - IGD_f	
Tests	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std
1	0.08071	0.01921	7.11178	0.58126	0.09862	0.02965	87.75893	4.19107	0.32733	0.03384	0.26669	0.03012
2	0.08092	0.01946	7.09336	0.71280	0.09862	0.02965	87.75517	4.19216	0.32729	0.03390	0.26749	0.02773
3	0.08082	0.01943	**7.07986**	0.63271	0.09862	0.02965	87.75512	4.19212	0.32684	0.03399	0.26524	0.02867
4	0.08076	0.01935	7.08948	0.68651	0.09862	0.02965	87.72650	4.20782	0.32675	0.03369	0.26432	0.02917
5	**0.08066**	0.01921	7.10559	0.58307	0.09863	0.02967	87.95238	4.21336	**0.32658**	0.03386	**0.26043**	0.02572
6	0.08084	0.01941	7.14410	0.60368	**0.09862**	0.02965	**87.69463**	4.17887	0.32702	0.03403	0.26109	0.03035
7	0.08151	0.02340	7.32560	0.51024	0.10093	0.02807	87.87557	3.47522	0.32150	0.03026	0.20438	0.03235
8	0.08150	0.02339	7.27459	0.51849	0.10093	0.02808	87.85075	2.98962	**0.32095**	0.02864	**0.19721**	0.03352
9	0.08153	0.02337	7.42823	0.58410	0.10093	0.02807	87.94513	3.50065	0.32106	0.02934	0.20621	0.03567
10	0.08154	0.02349	7.35128	0.50415	**0.10092**	0.02807	87.78711	3.31150	0.32167	0.02834	0.20905	0.03161
11	0.08155	0.02345	7.34596	0.51031	0.10093	0.02809	**87.59678**	3.20180	0.32136	0.02953	0.20746	0.03504
12	**0.08138**	0.02334	**7.19221**	0.62525	0.10093	0.02809	87.71771	3.48119	0.32182	0.02922	0.20508	0.03274
13	0.07900	0.03487	7.34952	0.68981	**0.07952**	0.02265	87.66299	4.45028	0.12920	0.00907	0.13077	0.01213
14	0.07838	0.03771	**7.33197**	0.69534	0.08048	0.02298	88.75991	3.87375	0.12838	0.01161	0.12252	0.01289
15	0.07688	0.03409	7.70273	1.31958	0.07969	0.02231	**87.01330**	5.24734	0.12462	0.01176	0.11306	0.01006
16	0.07739	0.04001	7.38552	0.59545	0.08053	0.02284	88.26105	4.26554	0.12222	0.01215	0.10558	0.01219
17	**0.07611**	0.03645	7.56501	0.67937	0.07980	0.02275	88.26542	4.74387	0.12188	0.01142	0.10082	0.01137
18	0.07765	0.03624	7.36063	0.77870	0.08007	0.02274	88.96298	4.58372	**0.12060**	0.01069	**0.09591**	0.00919
19	0.12072	0.02440	32.87390	14.33120	0.14376	0.03657	430.71277	141.55400	0.33029	0.02938	0.27892	0.04214
20	0.12146	0.02425	36.40087	13.25994	**0.14376**	0.03657	430.71523	141.55175	0.33016	0.02940	0.27940	0.04327
21	0.12085	0.02435	**31.31559**	14.11465	0.14381	0.03650	430.89771	141.35766	0.32996	0.02945	0.27621	0.03895
22	0.12122	0.02376	33.68889	13.92103	0.14389	0.03656	**428.84724**	144.11652	0.32989	0.02929	0.27514	0.04104
23	0.11933	0.02295	34.00893	16.78557	0.14376	0.03657	430.91362	141.33028	**0.32988**	0.02943	**0.27504**	0.03763
24	**0.11921**	0.02285	31.74919	14.30598	0.14376	0.03657	430.91627	141.32728	0.33014	0.02857	0.27787	0.03501
25	0.12181	0.02101	31.92999	22.56989	0.14090	0.03532	405.69336	158.81975	0.32600	0.03062	0.20397	0.03207
26	0.12293	0.02112	35.30762	25.86354	0.14101	0.03519	407.81652	158.75043	**0.32467**	0.03066	0.20190	0.03105
27	0.12431	0.02179	35.57332	32.94169	0.14101	0.03519	407.81254	158.75279	0.32578	0.03054	0.20077	0.03727
28	**0.12142**	0.02286	34.02119	24.18223	0.14101	0.03519	407.81330	158.75202	0.32597	0.03117	0.20823	0.03615
29	0.12241	0.02431	**31.13583**	17.48641	0.14101	0.03519	412.07004	158.03089	0.32577	0.03178	0.20340	0.03896
30	0.12390	0.02321	35.82697	23.96487	**0.14085**	0.03521	**398.76217**	148.53808	0.32521	0.03181	**0.19830**	0.03464
31	**0.16384**	0.05011	68.48640	51.67784	0.23234	0.05710	846.66454	297.14001	0.12977	0.01280	0.12934	0.01500
32	0.16856	0.05909	59.16842	54.37152	**0.22902**	0.05703	**833.75323**	263.03309	0.12474	0.01317	0.11770	0.01030
33	0.17072	0.04534	55.87618	39.29132	0.22940	0.05625	916.21595	314.54644	0.12451	0.01199	0.11412	0.01194
34	0.18779	0.06080	70.08573	57.04002	0.22962	0.06167	868.93923	270.80321	0.12165	0.01275	0.11008	0.00894
35	0.16634	0.05111	**49.85450**	31.74101	0.23115	0.06066	859.19644	267.32397	0.12092	0.01240	0.10345	0.00941
36	0.19156	0.07599	72.32754	69.76732	0.23123	0.05811	875.97880	223.90925	**0.11825**	0.01120	**0.09880**	0.01089

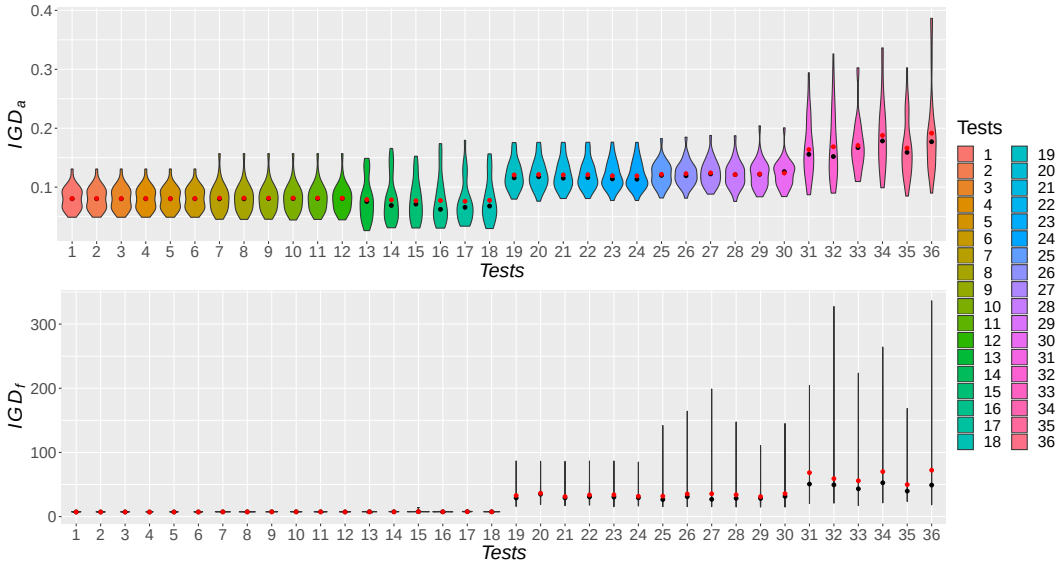

Figure 4.6: Violin plots for the IGD_a and IGD_f obtained on all tests in the PFHE problem.

PUC-Rio - Certificação Digital Nº 1812727/CA

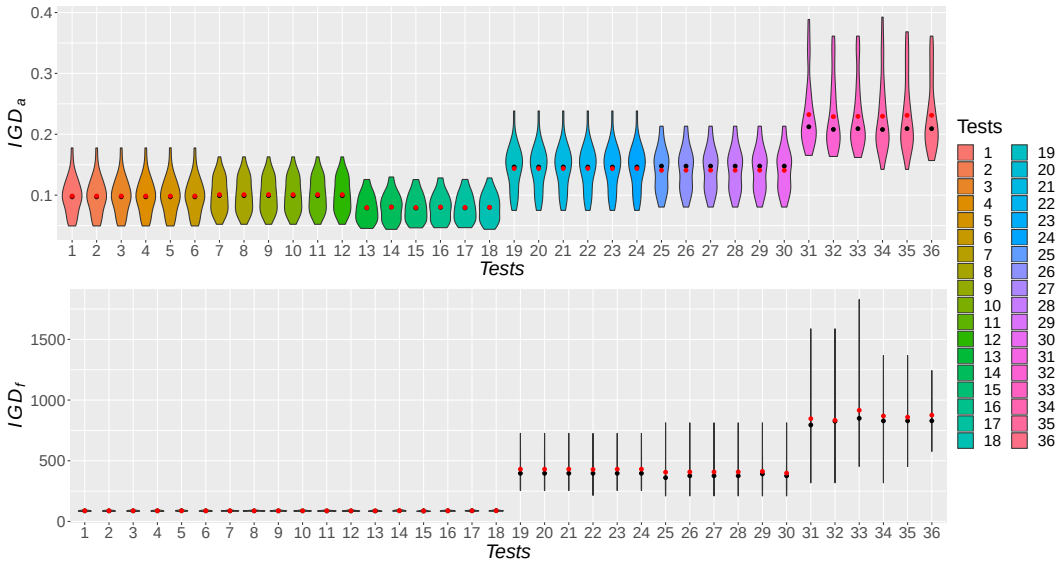

Figure 4.7: Violin plots for the IGD_a and IGD_f obtained on all tests in the STHE problem.

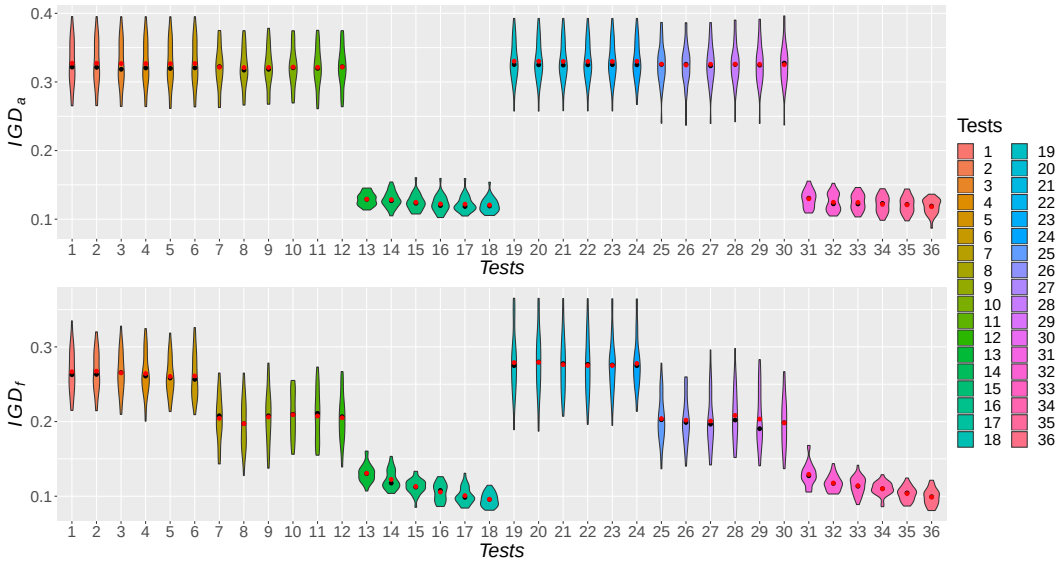

Figure 4.8: Violin plots for the IGD_a and IGD_f obtained on all tests in the ZDT1 problem.

According to Figures 4.6, 4.7 and 4.8 and with the Table 4.2, it is observed that i) the slowdown coefficient α did not have a relevant impact on the IGD_a and IGD_f metrics, which may perhaps be justified by the use of α typical values [62], that were large enough to avoid local minima (for both IGD_a and IGD_f) or because the maximum evaluation number is large enough to reach PFs very close to the real ones (for IGD_f); ii) even though MOHKArn and

PUC-Rio - Certificação Digital Nº 1812727/CA

Table 4.2: Median of the IGD_a and IGD_f metrics for all methods and problems.

	PFHE		STHE		ZDT1	
Tests	IGD_a	IGD_f	IGD_a	IGD_f	IGD_a	IGD_f
1	**0.08033**	7.27768	**0.09726**	87.39741	0.32168	0.26276
2	**0.08033**	7.24413	**0.09726**	**87.34070**	0.32137	0.26322
3	0.08034	7.30294	**0.09726**	87.34071	**0.31855**	0.26574
4	0.08034	**7.20299**	**0.09726**	87.34071	0.32040	0.26092
5	**0.08033**	7.24402	**0.09726**	88.17209	0.31973	0.25835
6	**0.08033**	7.25578	**0.09726**	87.34142	0.32061	**0.25655**
7	**0.08019**	7.45012	0.09915	87.03768	0.32230	0.20775
8	0.08026	7.44444	0.09915	87.03771	**0.31727**	**0.19718**
9	0.08068	7.50721	0.09915	87.03774	0.31849	0.20767
10	0.08060	7.49192	**0.09901**	86.93513	0.32064	0.20976
11	0.08071	7.45964	0.09915	86.78924	0.31989	0.21099
12	0.08081	**7.37480**	0.09915	**86.63860**	0.32215	0.20653
13	0.07561	7.57068	0.07896	86.55314	0.12892	0.13035
14	0.06894	7.53459	0.08044	87.85575	0.12686	0.11759
15	0.07126	7.62879	**0.07892**	**85.35071**	0.12328	0.11230
16	**0.06245**	**7.52240**	0.08003	87.19535	0.11987	0.10779
17	0.06582	7.75752	0.07904	87.77567	**0.11854**	0.09853
18	0.06791	7.57623	0.07982	88.22505	0.11924	**0.09578**
19	0.11580	**29.09568**	**0.14629**	396.80555	0.32536	0.27492
20	0.11756	34.96224	**0.14629**	**396.80160**	0.32494	0.27979
21	0.11528	29.22926	**0.14629**	396.80171	**0.32462**	0.27757
22	0.11612	30.72122	**0.14629**	396.80182	0.32499	0.27660
23	0.11401	30.40658	**0.14629**	396.80193	0.32518	0.27581
24	**0.11349**	29.24911	**0.14629**	396.80610	0.32496	**0.27483**
25	0.11996	**26.60234**	**0.14775**	**360.75719**	0.32555	0.20248
26	**0.11936**	31.06591	**0.14775**	376.04319	0.32580	0.19883
27	0.12244	26.97527	**0.14775**	376.04323	**0.32364**	0.19650
28	0.12122	28.52293	**0.14775**	376.04327	0.32552	0.20208
29	0.12180	28.31938	**0.14775**	391.16039	0.32452	**0.19044**
30	0.12631	31.52410	**0.14775**	376.04336	0.32746	0.19847
31	0.15574	50.71148	0.21227	**794.81287**	0.13059	0.12744
32	**0.15206**	49.45806	0.20804	825.65304	0.12241	0.11731
33	0.16705	43.35480	0.20921	849.79440	0.12225	0.11348
34	0.17840	52.70905	**0.20787**	828.65877	0.12290	0.11000
35	0.15911	**39.75508**	0.20921	829.27436	0.12162	0.10439
36	0.17710	49.08957	0.20921	828.95251	**0.11910**	**0.09923**

PUC-Rio - Certificação Digital Nº 1812727/CA

MOHKArpn yielded best results than the original MOHKA for ZDT1 problem, iii) the MOHKA versions that used the niching procedure (tests from 19 to 36) yielded worse results overall, when compared with the versions that used crowding distance, and finally iv) the MOHKArp method (tests from 13 to 18) presented the best performance, for providing a visually inferior median in the IGD_a metric for PFHE and STHE problems and practically tied with MOHKArpn in ZDT1 problem, and also a tied performance with the best IGD_f values for all problems.

In order to evaluate the best MOHKArp, each column interval of the Table 4.2 that corresponds to the MOHKArp IGD_a and IGD_f medians was separately scaled with zero mean and standard deviation of one, following by the sum of the row values found and the ranking of the sums, which allows us to state that the MOHKArp with $\alpha = 0.70$ (test 16) was the best method according to both IGD_a and IGD_f metrics. The same process was done with the other MOHKA versions, indicating the MOHKA with $\alpha = 0.90$ (test 6), MOHKAr with $\alpha = 0.50$ (test 8), MOHKAn with $\alpha = 0.60$ (test 21), MOHKArn with $\alpha = 0.60$ (test 27) and MOHKArpn with $\alpha = 0.80$ (test 35) as the best MOHKAs for each version.

Therefore, the Figures 4.9, 4.10 and 4.11 were generated, presenting the 30 rounds IGD median obtained during the evaluation number evolution in the PFHE, STHE and ZDT1 optimization problems, respectively.

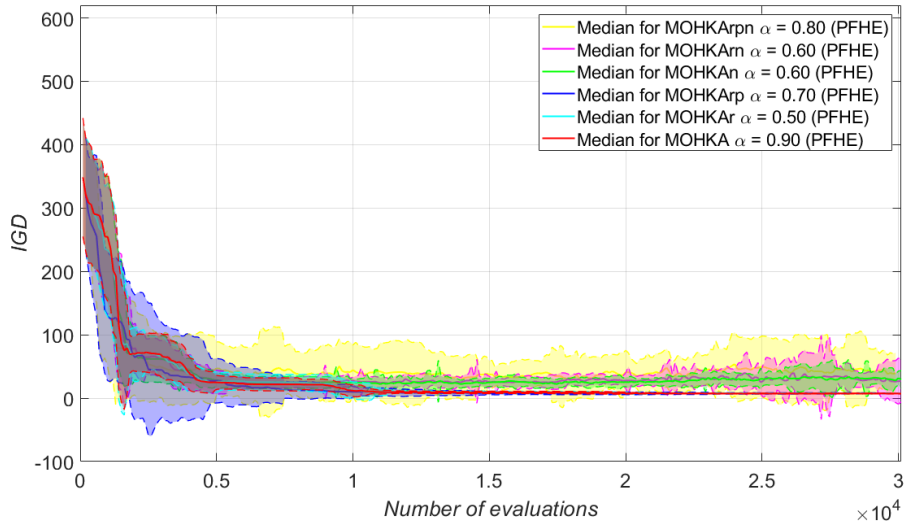

Figure 4.9: PFHE problem IGD evolution comparison for the best of each MOHKA version.

PUC-Rio - Certificação Digital Nº 1812727/CA

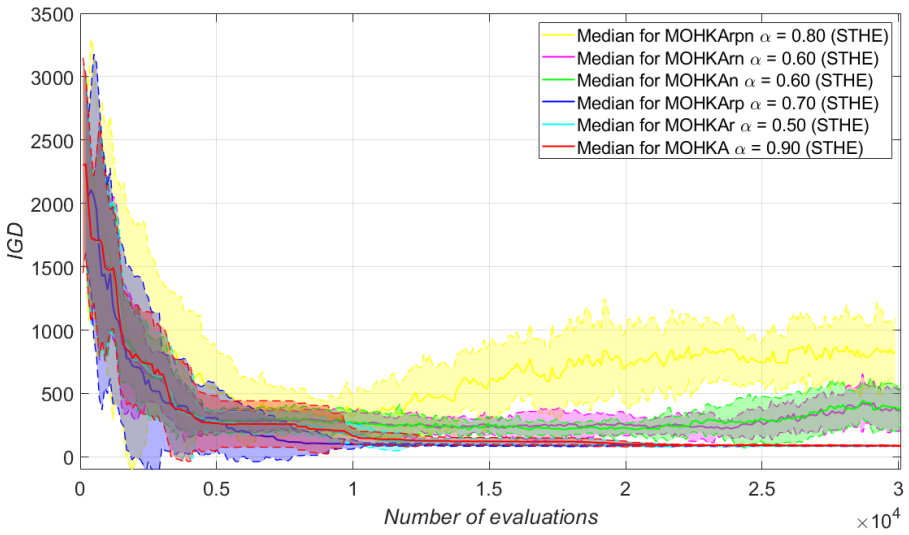

Figure 4.10: STHE problem IGD evolution comparison for the best of each MOHKA version.

The Figures 4.9 and 4.10 consider the IGD evolution of the best of each MOHKA version and confirms that for PFHE and STHE problems, the niching-procedure-dependent methods had a worse convergence in comparison with the methods that do not use this feature. This behavior changes in ZDT1 problem where MOHKA and MOHKAr lose, while MOHKArpn earn performance, being comparable to the MOHKArp IGD convergence performance, as

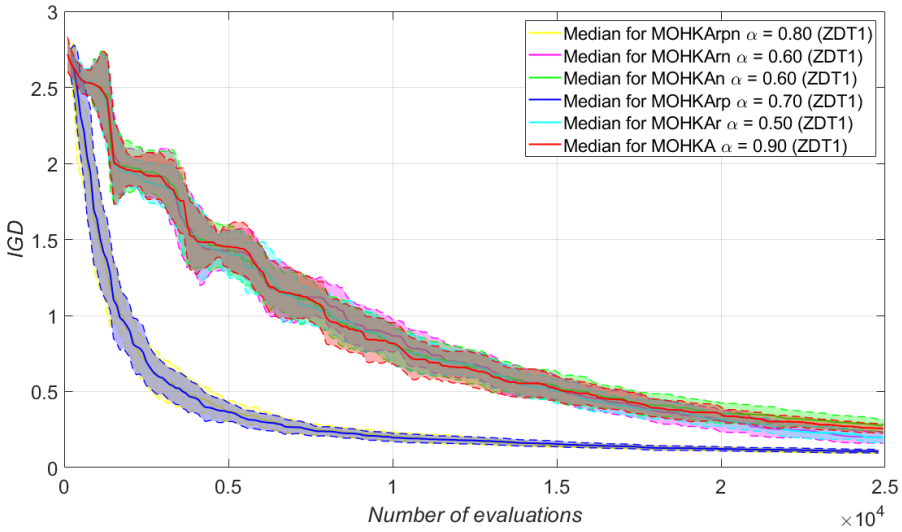

Figure 4.11: ZDT1 problem IGD evolution comparison for the best of each MOHKA version.

seen in Figure 4.11. In short, for PFHE and STHE problems, the MOHKArp has a slightly better convergence than the best MOHKA and MOHKAr, being these two tied. Nonetheless, for ZDT1, MOHKArp has a tied performance to the MOHKArpn and for all problems, MOHKArp converges faster.

Despite the best performance, Figure 4.9 reveals that the standard deviation of the MOHKArp with $\alpha = 0.70$ encompasses the MOHKA with $\alpha = 0.90$ IGD distribution for the PFHE problem, suggesting a statistically non-relevant result. The Wilcoxon rank sum test among the IGD_a and IGD_f medians obtained for tests 6 and 16 was made followinf the methodology in Section 2.3.5, yielding p-values that corroborates the suggestion (0.2366 for IGD_a and 0.1294 for IGD_f). However, for ZDT1 problem the performance enhance was significant (p-value of 1.8626E-09 for both metrics) and finally, for STHE problem the performance improvement was relevant for the IGD_a, but not for IGD_f (p-values of 0.0234 and 0.9032, respectively).

4.4 Conclusions

In this work we observed that the use of heat exchanger design problems for AI optimization techniques comparison is a common practice. Therefore, five new MOHKA versions were proposed, being compared with the traditional one through the IGD_a and IGD_f results for PFHE and STHE design problems, as well as the ZDT1 problem, seeking to enhance the general performance of the method and its attractive features, and also assess the impact of the MOHKA

PUC-Rio - Certificação Digital Nº 1812727/CA

slowdown coefficient.

The results had showed that i) the slowdown coefficient α did not impact the convergence of the methods, which may have been caused by the use of typical values that were effective in avoiding local minima or by the use of a maximum number of evaluations that allowed obtaining PFs very close to the real ones for PFHE and STHE problems, ii) with the excepetion of the ZDT1 problem, the adapted niching procedure was not effective in substitute the crowding distance operator as a diversity preservation mechanism and iii) the MOHKArp was the version that presented the fastest IGD convergence, although the median of the best MOHKArp ($\alpha = 0.70$) had presented a better performance in the IGD evolution curve when compared to the best MOHKA ($\alpha = 0.90$), the result is not statistically relevant for the PFHE problem for not having reached p-value lower than 0.05.

For future works, it would be interesting to: i) look for ways to improve the MOHKA versions, mainly MOHKArp, the best rated. One possibility would be to promote a change in the search modality as the optimization follows by randomness, thus developing a meta-heuristic based on MOHKA; and ii) evaluate the performance of niching-procedure-dependent codes in many-objectives problems (i.e. four or more objectives).

PUC-Rio - Certificação Digital Nº 1812727/CA

5
Pressure Transient Signals Feature Extraction for Illegal Tapping Detection with Supervised Learning

The present section seeks to depict the third work: 'Pressure Transient Signals Feature Extraction for Illegal Tapping Detection with Supervised Learning'. Therefore, this section is divided as follows: Section 5.1 defines the pipeline model and simulated tests; Section 5.2 reviews the main contributions related to the first work, presenting the methodology associated with them; Section 5.3 addresses the obtained results; and, finally Section 5.4 summarizes the conclusions drawn and future work suggestions.

5.1
Problem Description

The pipelines are widely used for the transport of fuel, chemicals and water, due to their economic, ecological and transport efficiency. However, when these structures fail, the potential for product spillage is high, causing economic and environmental losses [27]. As seen in Section 1, in recent years, the raise of fuel theft cases has led to an increase in the number of pipeline failures, which promotes the development of leak detection systems sensitive to theft.

Therefore, this section is divided into two parts: Section 5.1.1, which seeks to describe the pipeline structure and the properties of the product used in the simulation model, while the Section 5.1.2 details the simulated tests used to obtain the pressure transients corresponding to the pump start and stop, flow increase, outlet valve throttling and PSV actuation situations.

5.1.1
Pipeline Simulated Model

The simulated model is based on a real pipeline of 32" of diameter and 181.83 km of length. The duct thickness varied according to the length, being of 0.562", 0.500", 0.438", 0.406", 0.375", 0.344" and 0.250" for the respective stretches 0.000-8.932, 8.932-26,069, 26.069-37.466, 37.466-54.520, 54.520-69.181, 69.181-90.202 and 90.202-181.83, all expressed in kilometers (km). The model has two centrifugal booster pumps (A/B) with nominal flow

PUC-Rio - Certificação Digital Nº 1812727/CA

rate of 2,960 m^3/h and five centrifugal pumps (A/B/C/D/E) with nominal flow of 3,000 m^3/h and minimum flow of 454 m^3/h, all arranged in series. The product used in the simulation was a crude petroleum with density, viscosity, vapor pressure and Bulk modulus of 0.8986 kg/m^3, 66.36 cP, 0.50 kgf/cm^2abs and 15,751 kgf/cm^2, respectively, all magnitudes reported for 20°C. The pipeline pressure was observed at 8 different points over the simulation time, these points were called 'Sensor i', $i = 1, ..., 8$, being allocated at the positions of 0, 20, 60, 80, 100, 120, 160 and 180 km, respectively. The pipeline inlet and outlet operating pressure were 60 and 1 kgf/cm^2, respectively, while the operating flow rate (Q_{op}) was 2,568.42 m^3/h. Also, three valves were positioned at 18, 85 and 155 km, in order to simulate the pressure transients for the illegal tapping points, all with a valve opening that yields 100 m^3/h of theft flow rate, representing approximately 4% of the operating flow rate Q_{op} when openned. The simulation model is schematically presented in Figure 5.1.

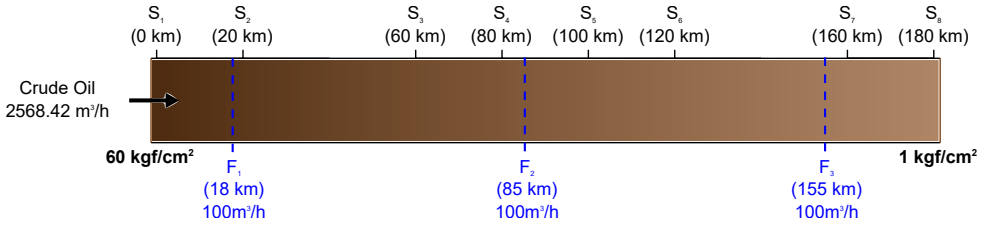

Figure 5.1: Simulated model scheme.

5.1.2 Simulated tests

Seeking to describe the simulated tests done to obtain the pressure transients, some procedures are explained next:

Procedure 1. (Start maneuver): booster A and pumps D, B and E are activated with a delay of 60s among them, totaling 180s duration;

Procedure 2. (Stop maneuver): in the same order as in the start maneuver, all booster and pumps are closed, being the respective downstream valves always closed before the upstream ones;

Procedure 3. (Flow increase): pump A is activated in order to increase the operating flow rate;

Procedure 4. (Choke): the pipeline outlet valve has it opening reduced to 2% of the initial opening;

Procedure 5. (PSV actuation) the pipeline outlet valve is completely closed in order to trigger the PSV present at the end of the line (181.83 km), which the opening set point is 20 kgf/cm^2;

PUC-Rio - Certificação Digital Nº 1812727/CA

Procedure 6. (Illegal tapping opening): the valve corresponding to the illegal tapping at beginning, middle or end regions of the pipeline (18/85/155 km) are opened.

The simulations were numbered from 1 to 7, being the simulations from 1 to 4 and from 5 to 7 performed to obtain normal pipeline operation transients and fuel theft transients, respectively. Knowing the procedures, the simulations are described in the sequence and the representation of the time pressure signals obtained for each sensor in each simulation are presented in the Figures 5.2, 5.3, 5.4 and 5.5. Also, for all simulations and sensors, 5 pressure measurements were recorded per second.

Simulation 1. Start Maneuver + 1860s + Stop Maneuver + 780s;

Simulation 2. Start Maneuver + 1860s + Flow Increase + 720s;

Simulation 3. Start Maneuver + 1860s + Choke + 1020s;

Simulation 4. Start Maneuver + 1860s + PSV Actuation + 1560s;

Simulation 5. Start Maneuver + 1620s + Illegal Tapping Opening (18km) + 240s + Stop Maneuver + 780s;

Simulation 6. Start Maneuver + 1620s + Illegal Tapping Opening (85km) + 240s + Stop Maneuver + 780s;

Simulation 7. Start Maneuver + 1620s + Illegal Tapping Opening (155km) + 240s + Stop Maneuver + 780s;

5.2 Contributions

As seen in Section 1.3, the use of: i) MLCs for the development of LDS is widespread; ii) the supervised learning approach for the MLCs training is the most common procedure for classification tasks [26]; and iii) the use of DT classifiers for LDS construction was little explored. Aware of the literature review conclusions and the importance of developing theft-sensitive LDS, the third work proposed the use of a physical pipeline based model to extract pipe leakage and regular operation pressure transient for three different scenarios. For this purpose, two DT are trained through time and PCA extracted features, respectively. The main contributions sought are to i) propose two different data-driven modeling workflows for theft sensitive LDS; ii) evaluate the use of DT classifiers for this application; and iii) evaluate which feature extraction approach generates the best DT model, according to the prediction performance in different scenarios.

Therefore, the present section is divided in two parts: Section 5.2.1 and Section 5.2.2, where the time and PCA/SVD feature extraction techniques are depicted, respectively.

PUC-Rio - Certificação Digital Nº 1812727/CA

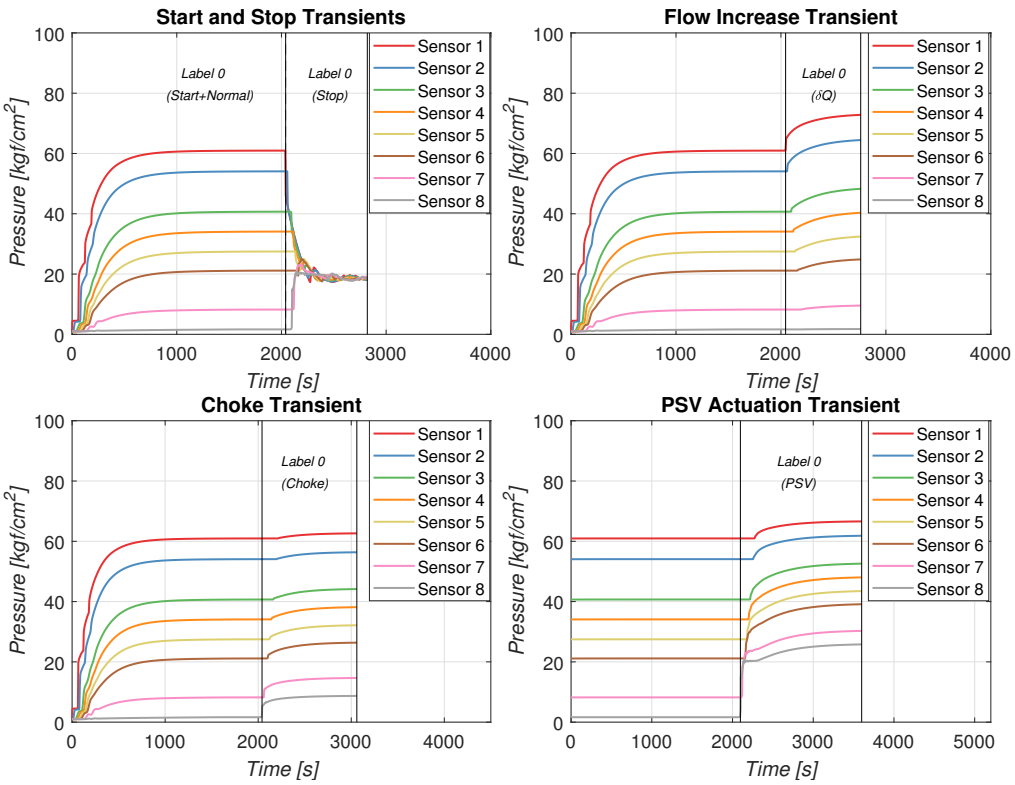

Figure 5.2: Simulations made to extract the normal duct operation transients.

Before the feature extraction procedure description, it is important to note that, for both approaches, the extraction was done on top of the data corresponding to the desirable transients (i.e. spliting the data into the exact pressure series which correspond to the desirable transient). For this purpose, the data of Simulations 2-7 went through an identification step seeking to find the first pressure signal p referent to the transients of Flow Increase, Choke, PSV Actuation and Illegal Tapping Opening at 18, 85 and 155 km, respectively, while for simulation 1 data it was considered $p = 1$ to extract features of Start and Stop Maneuver transients, as well as the steady state normal operation.

5.2.1 Time-based Feature Extraction

In the first feature extraction approach a window size $W_{sz} = 225$ data and a window step $N = 1$ were considered. For each window W_i, where $i = 1, \ldots, n$ indicates the number of windows/observations, the time-based features were calculated, i.e. 28 possible cross-correlations (XC) among the pressure data of the 8 and 72 statistical features, corresponding to Kurtosis coefficient (K), L2-energy norm (E), curve length coefficient (CL), mean (μ), median (m), standard deviation (S), maximum (Max), minimum (Min) and the difference

PUC-Rio - Certificação Digital Nº 1812727/CA

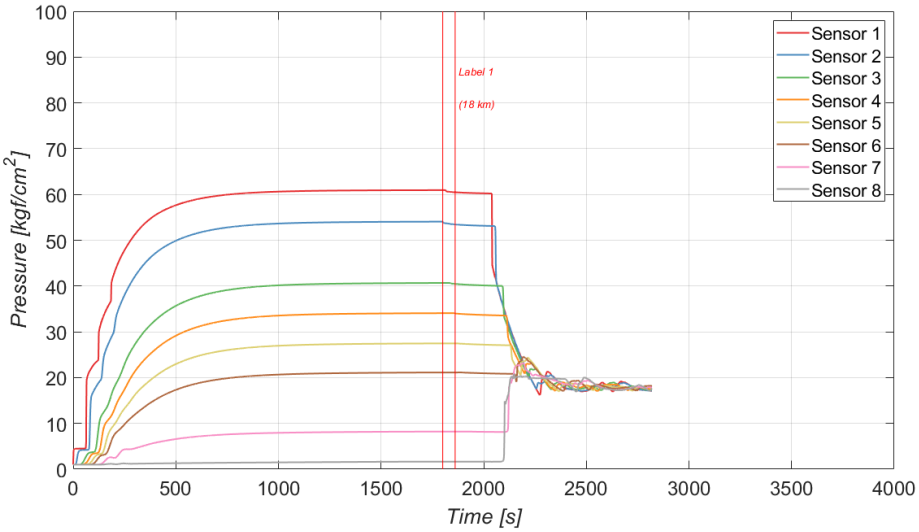

Figure 5.3: Simulations made to extract the fuel theft pressure transients at the 18 km illegal tapping.

among maximum and minimum (A), for each one of the 8 sensors' pressure series. The formulas for calculating the features are depicted below [147, 148]:

$$XC^i_{j-k} = \frac{\sum_{ii=1}^{W_{sz}}[x_j(ii) - \bar{x_j}][x_k(ii) - \bar{x_k}]}{\sqrt{\sum_{ii=1}^{W_{sz}}[x_j(ii) - \bar{x_j}]^2}\sqrt{\sum_{ii=1}^{W_{sz}}[x_k(ii) - \bar{x_k}]^2}} \tag{5-1}$$

$$K^i_k = \frac{\frac{1}{W_{sz}}\sum_{ii=1}^{W_{sz}}[x_k(ii) - \bar{x_k}]^4}{\{\frac{1}{W_{sz}}\sum_{ii=1}^{W_{sz}}[x_k(ii) - \bar{x_k}]^2\}^2} \tag{5-2}$$

$$E^i_k = \|x_k\|_2 = \sqrt{\sum_{ii=1}^{W_{sz}} x_k(ii)^2} \tag{5-3}$$

$$CL^i_k = \sum_{ii=2}^{W_{sz}} \mid x_k(ii) - x_k(ii-1) \mid \tag{5-4}$$

$$\mu^i_k = \bar{x_k} = \frac{1}{W_{sz}}\sum_{ii=1}^{W_{sz}} x_k(ii) \tag{5-5}$$

$$m^i_k = \begin{cases} x_k((W_{sz}+1)/2) & \text{if } W_{sz} \text{ is odd} \\ [x_k(W_{sz}/2) + x_k(W_{sz}+1)]/2 & \text{if } W_{sz} \text{ is even} \end{cases} \tag{5-6}$$

$$S^i_k = \sqrt{\frac{1}{W_{sz}-1}\sum_{ii=1}^{W_{sz}} \mid x_k(ii) - \bar{x_k} \mid} \tag{5-7}$$

$$Max^i_k = max(x_k) \tag{5-8}$$

$$Min^i_k = min(x_k) \tag{5-9}$$

$$A^i_k = max(x_k) - min(x_k) \tag{5-10}$$

where XC^i_{j-k}, K^i_k, E^i_k, CL^i_k, μ^i_k, m^i_k, S^i_k, Max^i_k, Min^i_k and A^i_k are the cross-

PUC-Rio - Certificação Digital Nº 1812727/CA

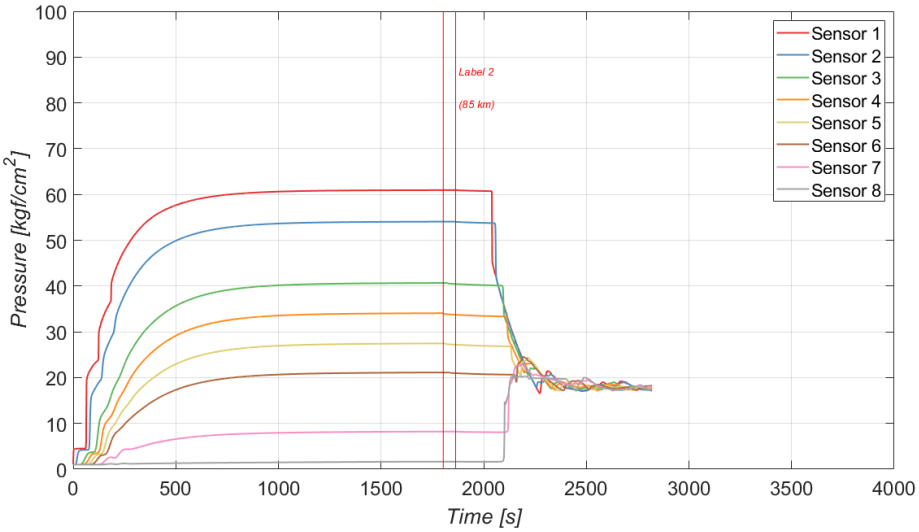

Figure 5.4: Simulations made to extract the fuel theft pressure transients at the 85 km illegal tapping.

correlation, Kurtosis, L2-energy norm, Curve length, mean, median, standard deviation, maximum, minimum and the difference between these last two, all for the pressure series x correspondent to the window W_i, considering the sensors j and k, being $j = 1, \ldots, 7$, $k = 1, \ldots, 8$ and $j < k$.

Therefore, the training matrix F_1 was constructed from the 100 features values in columns for the 62,271 windows analyzed in rows and with one extra column of classes, presenting i) label 0 if the features were extracted from the normal steady state operation or transients of Start and Stop Maneuver, Flow Increase, Choke and PSV Actuation or ii) labels 1, 2 and 3 if the features considered the pressure data of the Illegal Tapping Opening transients at 18, 85 and 155 km, respectively.

5.2.2 PCA-based Feature Extraction

For the second features extraction approach, the X matrix of observations used for DT training was constructed by concatenating in each row X_i the pressure series correspondent to the window W_i for all pressure sensors P_k, $k = 1, \ldots, 8$, where $i = 1, \ldots, n$ indicates the number of windows/observations. The window size considered was $W_{sz} = 200$, thus each row X_i has 1600 terms. Also, the window step N and the number of observations n varied according to the Table 5.1, making the observation matrix X have 1,020 observations. For a better comprehension of the observation matrix construction, the Figure 5.6 is provided.

PUC-Rio - Certificação Digital Nº 1812727/CA

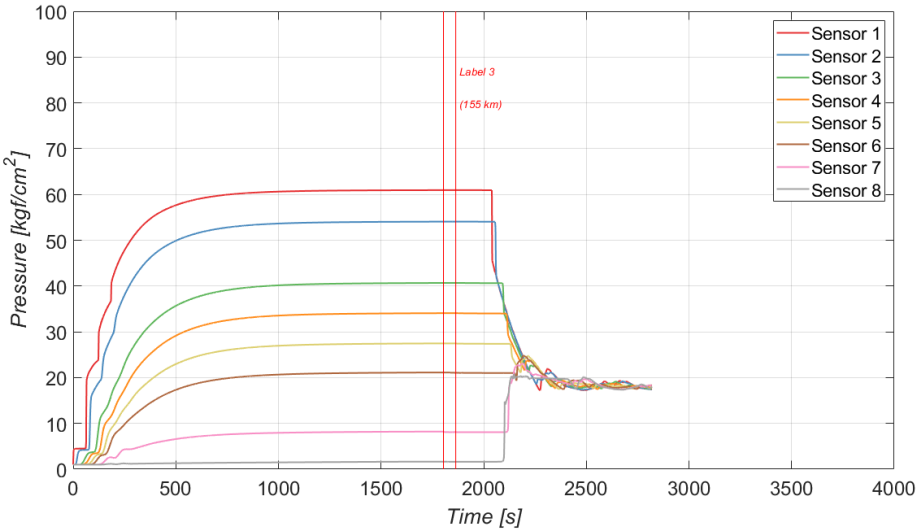

Figure 5.5: Simulations made to extract the fuel theft pressure transients at the 155 km illegal tapping.

Table 5.1: Window step and number of observations for each simulation.

Simulations	Window step (N)	Number of windows (n)
1	20	270
2-4	10	50
5-7	1	200

With the observation matrix X, the features are extracted with the PCA approach described in Section 2.5.2, yielding the matrix P of principal components. Also, in the same way made for time-based features in Section 5.2.1, seeking to train the decision tree with a supervised learning approach, the corresponding 'target'/'class' of each window is concatenated in an additional column with the principal component matrix P, generating the training matrix F_2.

5.3 Results

Firstly, all results obtained in the training and validation steps of the DT model were generated considering respectively 70% and 30% of the training matrices, i.e. F_1 for the time-based feature approach and F_2 for the PCA/SVD approach. All approaches considered Gini index as split criterion [124].

In order to evaluate if the DT model is suitable for this problem, considering the first feature extraction approach, five DT were constructed for each type of time-based features, which generated Table 5.2. The results of Table 5.2 demonstrate that the DT classifiers can achieve a good classification

PUC-Rio - Certificação Digital Nº 1812727/CA

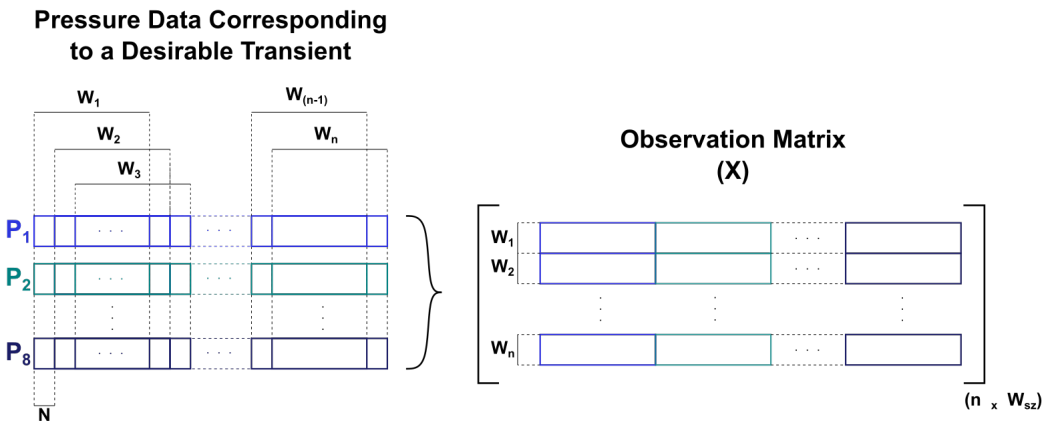

Figure 5.6: Observation matrix composition for feature extraction with SVD/PCA approach.

performance for the LDS problem, however, all DT models presented false alarms, similar to those seen in Figure 5.7.

An important note is that the number of time-based features (100) allows and suggests the use of a feature selection technique, to derive an optimized DT classifier for LDS problem. The optimization would remove non-competent features and bring some benefits, as reducing the training data size, speeding the learning process, decreasing complexity or even enhancing the DT performance [149]. Some feature selection trials were made considering a multi-objective approach with number of features and validation accuracy as the objectives, but the final results performed worse compared to those reported in the Table 5.2, even dominating these ones.

Table 5.2: Accuracy obtained for decision trees trained from a single type of time based feature and for several split numbers.

Feature		Number of splits				
Type	Number	10	20	30	40	50
XC	28	0.9951	0.9990	0.9995	0.9996	0.9996
K	8	0.9705	0.9748	0.9786	0.9802	0.9827
E	8	0.9993	0.9993	0.9993	0.9993	0.9993
CL	8	0.9960	0.9996	0.9996	0.9995	0.9995
μ	8	0.9993	0.9993	0.9993	0.9993	0.9993
m	8	0.9991	0.9990	0.9990	0.9990	0.9990
S	8	0.9993	0.9995	0.9995	0.9995	0.9995
Max	8	0.9996	0.9996	0.9996	0.9996	0.9996
Min	8	0.9992	0.9991	0.9991	0.9991	0.9991
A	8	0.9995	0.9991	0.9991	0.9991	0.9991

Therefore, we sought to eliminate the false alarms and improve the classification performance by using 2 types of time-based features for the

PUC-Rio - Certificação Digital Nº 1812727/CA

training. For considering i) the correlation among the sensor important to identify leak region and ii) the curve length property to react to changes in modal amplitudes or waves location [147], five DT with the 36 features (28 cross-correlation and 8 Curve Length coefficients) and a maximum number of splits varying from 10 to 50 was trained, generating the Table 5.3.

Table 5.3: Accuracy obtained for decision trees trained from cross-correlation and curve length features, considering several split numbers.

Feature		Number of splits				
Type	Number	10	20	30	40	50
$XC + CL$	36	0.9968	0.9998	0.9997	0.9997	0.9997

According to Table 5.2 and Table 5.3, it is concluded that the best accuracy among trained models was obtained by the DT with 36 features and maximum number of 20 splits. Thus, the proportion of classes for the time-based features approach and for the best DT model, the confusion charts of training and validation steps, as well as the validation step predictions are exposed in Figures 5.7, 5.8 and 5.9.

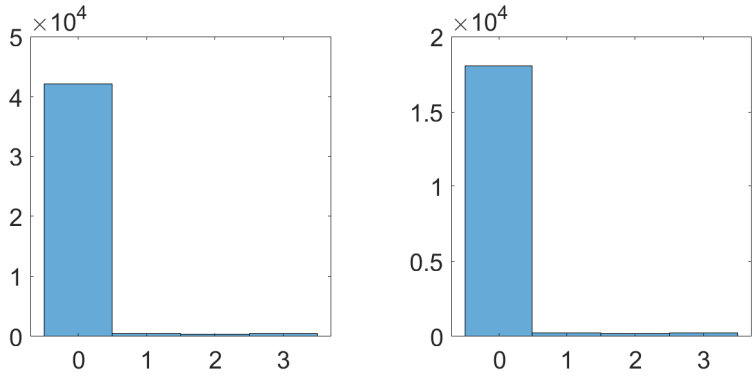

Figure 5.7: Absolute number of observations for each class for the first feature extraction approach training (left) and validation (right).

The DT generated by the first feature extraction approach considered 41 nodes and 12 splits, presenting a great classification performance for both training and validation data in general terms, according to the confusion charts presented in Figure 5.8 and with the predictions made for the validation step in Figure 5.9. Even so, despite the accuracy of 99.98% in the validation step, the presence of false positives around the number of 5000 observations invalidates the pipeline fuel theft detection model.

Also, an important note is that the different number of observations per class presented by Figure 5.7 suggests the use of a imbalance compensation

PUC-Rio - Certificação Digital Nº 1812727/CA

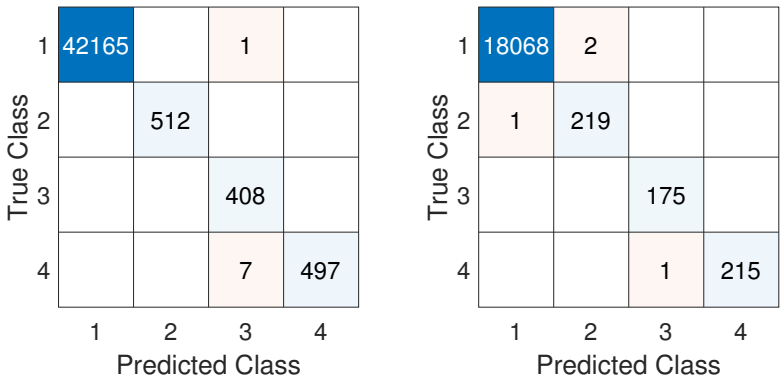

Figure 5.8: Confusion charts for training (left) and validation (right) for the decision tree obtained through first feature extraction approach.

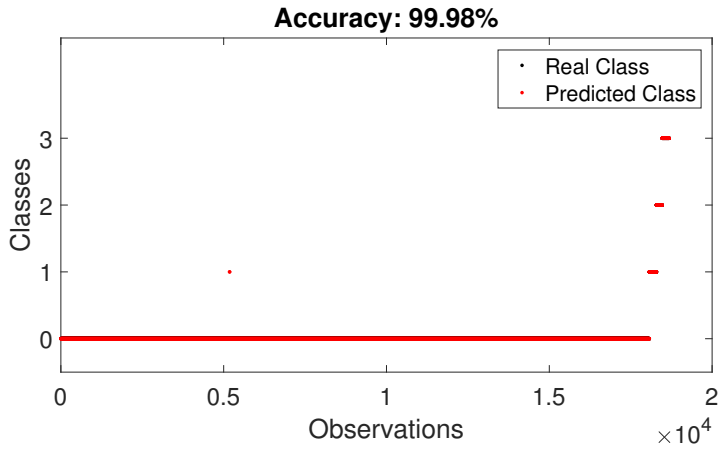

Figure 5.9: Prediction of the validation data for the decision tree obtained through first feature extraction approach.

technique. However, when a random undersampling technique [150] was applied to the training data, the DT generated had a loss of performance, making the results with imbalance kept in the study.

For the PCA/SVD feature extraction approach, the DT was trained considering a maximum number of 100 splits. The proportion of classes, the confusion charts of training and validation steps for the PCA/SVD approach and the predictions for the validation step are shown in Figures 5.10, 5.11 and 5.12.

The DT generated by the second feature extraction approach considered 13 nodes and 6 splits, presenting a perfect classification performance for both training and validation data in general terms, according to the confusion charts presented in Figure 5.11 and with the predictions made for the validation step in Figure 5.12. The accuracy of 100% in the validation predictions may indicate that the second feature extraction approach is more conducive to DT model training and although it lacks a heavier data processing, as it extracts features from SVD, the PCA/SVD approach i) generates DTs with simpler

PUC-Rio - Certificação Digital Nº 1812727/CA

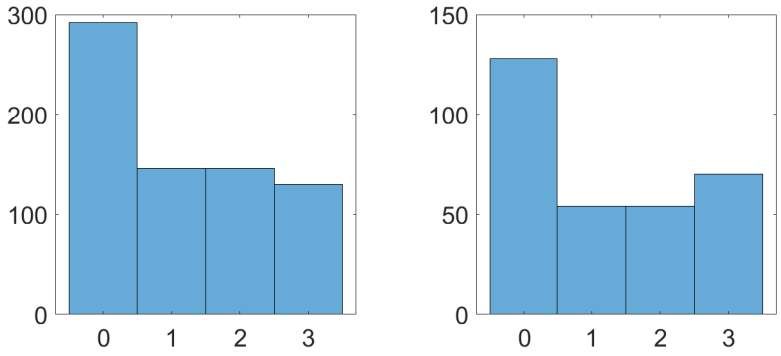

Figure 5.10: Absolute number of observations for each class for the second feature extraction approach training (left) and validation (right).

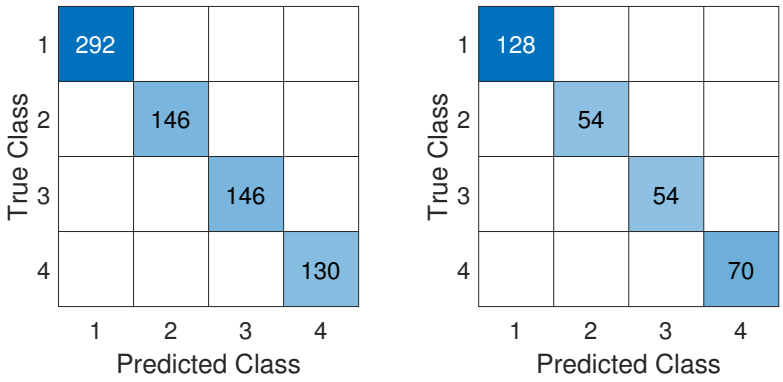

Figure 5.11: Confusion charts for training (left) and validation (right) for the decision tree obtained through second feature extraction approach.

structure, i.e. fewer nodes and splits; ii) demands less data in comparison with the first approach to train DT models with similar accuracy, i.e. while the first approach demands 62,271 observations to achieve accuracy of 99.98%, the second approach just needs 1,020 to yield 100 % accuracy performance; and iii) builds valid DT models, while the first approach generated invalid ones by the presence of false positives.

According to the confusion charts presented in Figure 5.11 and with the predictions made in the validation step in Figure 5.12, it can be inferred that, the decision tree generated from the second feature extraction approach has a perfect classification performance for both training and validation data. The accuracy of 100% in the validation predictions may indicate that the second feature extraction approach is more conducive to DT model training and although it lacks a heavier data processing, as it extracts features from SVD, the PCA/SVD approach i) generates DTs with simpler structure and ii) needs less data in comparison with the first approach to train DT models

PUC-Rio - Certificação Digital Nº 1812727/CA

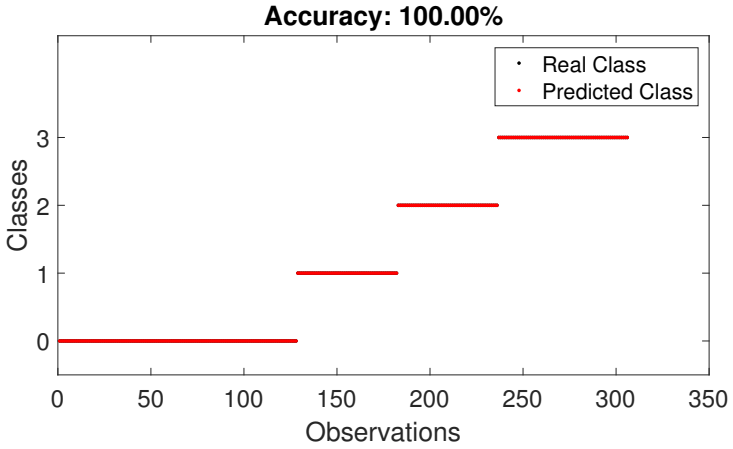

Figure 5.12: Prediction of the validation data for the decision tree obtained through second feature extraction approach.

with similar accuracy, and iii) builds valid DT models, while the first approach generated invalid models by the presence of false positives.

5.4 Conclusions

The present contribution applied the machine learning technique known as DT classifier through supervised learning to address the problem of theft sensitive LDS. Seeking to achieve this goal, two DT training approaches were considered, where training features based on time and on PCA made through SVD were employed for the first and second approach, respectively.

The DT validation results showed that the two approaches were able to identify the pipeline pressure transients of normal operation and the tapping points present in three different positions with great accuracy (99.98% and 100% for the first and second approach, respectively). However, despite the extra computational cost demanded for the feature extraction, the PCA/SVD approach is more suitable for the LDS application due to: i) building DT classifiers with simpler structure for presenting models with fewer nodes and splits number; ii) less training data need (1,020 observations) for producing DT models with similar accuracy, in comparison with the ones produced by the first approach (62,271 observations); and iii) building DT classifiers which yields valid results, in contrast with the invalid ones obtained for the first feature extraction approach, by the presence of false positives.

For future work, it is proposed for the i) first feature extraction approach: the search of metrics of model complexity and performance or maximum relevance and minimum redundancy that allow a feature selection which yields DT classifiers with simpler structure and a lower false alarm rate, in

PUC-Rio - Certificação Digital Nº 1812727/CA

addition to considering other time-based features; ii) second feature extraction approach: the DT hyperparameters optimization through a multi-objective approach, considering metrics of complexity and performance; and iii) problem complexity enhancement: evaluate situations with simultaneous occurrence of multiple illegal tapping points, develop a pressure analysis interface for an accurate location of the theft point and, finally, consider pipelines with altitude/level variation along the line or multi-product flow.

PUC-Rio - Certificação Digital Nº 1812727/CA

6
Conclusion

The present thesis sought to apply AI methods for the solution of real-world mechanical engineering problems. Thus, this work proposes: i) develop and improve AI techniques; ii) constitute an application guide for control and design multi-objective optimization problems, feature extraction process and machine learning classifier training through supervised approach; and mainly iii) demonstrate potential of AI techniques for mechanical engineering problems solving.

Seeking to achieve this goals, three mechanical engineering problems were proposed: i) the PID-based ABS control tuning through bio-inspired meta-heuristics; ii) the heat exchanger design multi-objective optimization through other AI heuristic; and iii) the development of theft-sensitive leak detection systems through machine learning classifiers. The importance of solving these problems, the contribution sought, the main conclusions and suggestions for future work are depicted in the next three paragraphs, respectively.

Originally proposed to prevent wheel lock in heavy braking and slippery road situations, avoiding accidents, the ABS may also provide comfort during braking activity. Still, the ABS literature review revealed that single-wheel models are sufficient and often used for the ABS study and development, the majority of ABS controllers present a high complexity and in case of PID-based controllers, the use of new meta-heuristics for tuning is little explored. Thus, the first work considered the MO of a PID-controlled ABS, considering both performance and comfort metrics, through NSGA-II, MODA, MSSA and OBLI-MSSA. Also, the methods PF were compared through non-dominance percentual, spacing (S), Euclidian distance (ED) and hypervolume (HV) comparison metrics. The main contributions are: i) a new multi-objective formulation for PID-controlled ABS improvement; ii) the development of OBLI-MSSA, a new MSSA version; and iii) the comparison of recent bio-inspired meta-heuristics (MODA and MSSA) with traditional AI optimization methods (NSGA-II) to follow the state of the art evolution. The results showed that the single PID controller did not allow the methods comparison, due to the proximity of results. However, for the five PID controller, the OBLI-MSSA was considered the best optimization method, since its PF

PUC-Rio - Certificação Digital Nº 1812727/CA

yielded greater dominance, achieved spacing results that ensure the best distribution of solutions on the PF and satisfatory/comparable results on Euclidian distance and hypervolume, in between the studied methods. Also due to the dominance, S, ED and HV metrics, the MSSA was considered to have a better performance than MODA and NSGA-II; MODA presented a similar performance to NSGA-II, but the last method presented the smallest standard deviation in all comparison metrics, being the most constant method for all comparison metrics. For future works it would be interesting to: i) evaluate if the proposed initialization method for the OBLI-MSSA would yield better results, if applied to MODA and NSGA-II; ii) seek ways to guarantee the same results stability obtained by NSGA-II in other meta-heuristics, maybe making a fine tuning of the MO parameters; and iii) consider other controllers for the braking problem, e.g. a fractional PID or Fuzzy-PID, which are likely to outperform the conventional PID.

Heat exchangers are common devices widely used in the industry fields, due to the ability of transfering thermal energy between fluids or fluid and solid materials. Therefore, owing to their function, the heat exchanger effectiveness directly impacts the industrial process efficiency in which they are used. The HE design literature review revealed that the MO of HE projects is a trend, considering total annual cost and performance metrics, as well as the use of these problems for AI optimization methods comparison. Thus, the second work proposed five new MOHKA versions and compared them with the traditional one through the IGD_a and IGD_f results for PFHE and STHE design problems, as well as the ZDT1 problem, seeking to enhance the general performance of the method and its attractive features; and also assess the impact of the MOHKA slowdown factor. The results had showed that i) the slowdown factor α did not impact the convergence of the methods, which may have been caused by the use of typical values that were effective in avoiding local minima or by the use of a maximum number of evaluations that allowed obtaining PFs very close to the real ones for PFHE and STHE problems, ii) with the excepetion of the ZDT1 problem, the adapted niching procedure was not effective in substitute the crowding distance operator as a diversity preservation mechanism and iii) the MOHKArp was the version that presented the fastest IGD convergence, although the median of the best MOHKArp ($\alpha = 0.70$) had presented a better performance in the IGD evolution curve when compared to the best MOHKA ($\alpha = 0.90$), the result is not statistically relevant for the PFHE problem for not having reached p-value lower than 0.05. For future works, it would be interesting to: i) look for ways to improve the MOHKA versions, mainly MOHKArp, the best rated. One possibility would

PUC-Rio - Certificação Digital Nº 1812727/CA

be to promote a change in the search modality as the optimization follows by randomness, thus developing a meta-heuristic based on MOHKA; and ii) evaluate the performance of niching-procedure-dependent codes in many-objectives problems (i.e. four or more objectives).

Pipelines are the main means of transporting fossil fuels, chemicals and water, due to the economic, ecological and transport efficiency. However, in case of failure, pipelines have a great potential for product spillage, causing economic and environmental losses. Seeking to prevent or mitigate product spillage due to pipe failures, leak detection systems were developed. In recent years, the raise of fuel theft cases has led to an increase in the number of pipeline failures with annual cost reaching billions [36, 37], and since the theft flow rate is small enough not to be detected by traditional LDS, the development of theft-sensitive LDS is fostered. The LDS literature review revaled that MLCs trained through supervised learning approach are widely used to address the LDS problem and, among the ML techniques, DT are little explored. Therefore, the third work proposed to train two DT through supervised learning, considering time-based and PCA/SVD feature extractions approaches from pressure transiente data. Thus, the main contributions are: i) the development of two different data-driven modeling workflows for theft-sensitive LDS; ii) evaluate the use of DT classifiers for LDS problem; and iii) evaluate which feature extraction approach generates the best DT model, according to the prediction performance in different scenarios. The DT validation results showed that the two training approaches were able to classify with great accuracy accuracy (99.98% and 100% for the first and second approach, respectively) the observations corresponding to the transients of normal operation and of operation with illegal tapping in three different positions. However, despite the extra computational cost demanded for the feature extraction, the most appropriate approach for the LDS application is the PCA/SVD, since the DT classifiers built through this approach present a simpler structure, demanding less training data for producing similar accuracy DT models, in comparison with the ones produced by the first approach and, mainly, for building DT classifiers that yields valid results, not showing false positives. For future work, some suggestions would be: i) the search for metrics of model complexity and performance or maximum relevance and minimum redundancy, which allow the feature selection process of the first feature extraction approach, providing DT classifiers with simpler structure and a lower false alarm rate, in addition to considering other time-based features; ii) perform hyper-parameters optmization of the second feature extraction approach DT, considering a MO approach with complexity and performance metrics; iii) to promote the problem com-

PUC-Rio - Certificação Digital Nº 1812727/CA

plexity enhancement, by evaluating situations with simultaneous occurrence of multiple illegal tapping points, developing a pressure analysis interface for an accurate location of the theft point, considering pipelines with altitude/level variation along the line or with multi-product flow.

PUC-Rio - Certificação Digital Nº 1812727/CA

Bibliography

[1] Javier Del Ser, Eneko Osaba, Daniel Molina, Xin-She Yang, Sancho Salcedo-Sanz, David Camacho, Swagatam Das, Ponnuthurai N Suganthan, Carlos A Coello Coello, and Francisco Herrera. Bio-inspired computation: where we stand and what's next. *Swarm and Evolutionary Computation*, 48:220–250, 2019.

[2] Nazmul Siddique and Hojjat Adeli. *Computational intelligence: synergies of fuzzy logic, neural networks and evolutionary computing*. John Wiley & Sons, Chichester, WS, UK, 1st edition, 2013.

[3] Gao Huang, Shiji Song, Jatinder N D Gupta, and Cheng Wu. Semi-supervised and unsupervised extreme learning machines. *IEEE transactions on cybernetics*, 44(12):2405–2417, 2014.

[4] Zengmao Wang, Bo Du, Lefei Zhang, Liangpei Zhang, and Xiuping Jia. A novel semisupervised active-learning algorithm for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 55(6):3071–3083, 2017.

[5] Richard S Sutton and Andrew G Barto. *Reinforcement learning: an introduction*, volume 1. MIT press, Mountain View, CA, USA, 2nd edition, 2018.

[6] Randy L Haupt and Sue Ellen Haupt. *Practical genetic algorithms*. Wiley Online Library, Hoboken, NJ, USA, 2nd edition, 2004.

[7] Carlos A Coello Coello, Gary B Lamont, David A Van Veldhuizen, et al. *Evolutionary algorithms for solving multi-objective problems*, volume 5. Springer US, New York, NY, USA, 2nd edition, 2007.

[8] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and T Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE transactions on evolutionary computation*, 6(2):182–197, 2002.

[9] Seyedali Mirjalili. Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. *Neural Computing and Applications*, 27(4):1053–1073, 2016.

PUC-Rio - Certificação Digital Nº 1812727/CA

[10] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An introduction to statistical learning*, volume 112. Springer-Verlag New York, 1st edition, 2013.

[11] Sergio M Savaresi and Mara Tanelli. *Active braking control systems design for vehicles*. Springer-Verlag London, London, UK, 1st edition, 2010.

[12] Sepehr Sanaye and Hassan Hajabdollahi. Thermal-economic multi-objective optimization of plate fin heat exchanger using genetic algorithm. *Applied Energy*, 87(6):1893–1902, 2010.

[13] R Venkata Rao and Vivek Patel. Multi-objective optimization of heat exchangers using a modified teaching-learning-based optimization algorithm. *Applied Mathematical Modelling*, 37(3):1147–1162, 2013.

[14] Eckart Zitzler, Kalyanmoy Deb, and Lothar Thiele. Comparison of multiobjective evolutionary algorithms: empirical results. *Evolutionary computation*, 8(2):173–195, 2000.

[15] Manuel Martínez, Sergio García-Nieto, Javier Sanchis, and X Blasco. Genetic algorithms optimization for normalized normal constraint method under pareto construction. *Advances in engineering software*, 40(4):260–267, 2009.

[16] Jürgen Branke, Jurgen Branke, Kalyanmoy Deb, Kaisa Miettinen, and Roman Slowiński. *Multiobjective optimization: interactive and evolutionary approaches*, volume 5252. Springer-Verlag Berlin Heidelberg, Berlim, DE, 1st edition, 2008.

[17] Helon Vicente Hultmann Ayala and Leandro dos Santos Coelho. Tuning of pid controller based on a multiobjective genetic algorithm applied to a robotic manipulator. *Expert Systems with Applications*, 39(10):8968–8974, 2012.

[18] Rajesh Rajamani. *Vehicle dynamics and control*. Springer US, New York, NY, USA, 2nd edition, 2011.

[19] Abdel Badie Sharkawy. Genetic fuzzy self-tuning pid controllers for antilock braking systems. *Engineering Applications of Artificial Intelligence*, 23(7):1041–1052, 2010.

[20] N Raesian, N Khajehpour, and M Yaghoobi. A new approach in antilock braking system (abs) based on adaptive neuro-fuzzy self-tuning pid

PUC-Rio - Certificação Digital Nº 1812727/CA

controller. In *The 2nd International Conference on Control, Instrumentation and Automation*, pages 530–535, Shiraz, Iran, 2011. IEEE.

[21] Yanan Qiu, Xiaogeng Liang, and Zhiyong Dai. Backstepping dynamic surface control for an anti-skid braking system. *Control Engineering Practice*, 42:140–152, 2015.

[22] Luca D'Avico, M Tanelli, SM Savaresi, M Airoldi, and G Rapicano. A deceleration-based algorithm for anti-skid control of aircraft. *IFAC-PapersOnLine*, 50(1):14168–14173, 2017.

[23] Ramesh K Shah and Dusan P Sekulic. *Fundamentals of heat exchanger design*. John Wiley & Sons, Hoboken, NJ, USA, 1st edition, 2003.

[24] Giampietro Fabbri. Heat transfer optimization in corrugated wall channels. *International Journal of Heat and Mass Transfer*, 43(23):4299–4310, 2000.

[25] Jemmy S Bintoro, Aliakbar Akbarzadeh, and Masataka Mochizuki. A closed-loop electronics cooling by implementing single phase impinging jet and mini channels heat exchanger. *Applied thermal engineering*, 25(17-18):2740–2753, 2005.

[26] Aurélien Géron. *Hands-on machine learning with scikit-learn, keras, and tensorFlow: concepts, tools, and techniques to build intelligent systems*. O'Reilly Media, Sebastopol, CA, USA, 2nd edition, 2019.

[27] Qiang Chen, Guodong Shen, JunCheng Jiang, Xu Diao, Zhirong Wang, Lei Ni, and Zhan Dou. Effect of rubber washers on leak location for assembled pressurized liquid pipeline based on negative pressure wave method. *Process Safety and Environmental Protection*, 119:181–190, 2018.

[28] Pal-Stefan Murvay and Ioan Silea. A survey on gas leak detection and localization techniques. *Journal of Loss Prevention in the Process Industries*, 25(6):966–973, 2012.

[29] Shuaiyong Li, Yumei Wen, Ping Li, Jin Yang, Xiaoxuan Dong, and Yanhua Mu. Leak location in gas pipelines using cross-time–frequency spectrum of leakage-induced acoustic vibrations. *Journal of Sound and Vibration*, 333(17):3889–3903, 2014.

[30] Cui-wei Liu, Yu-xing Li, Yu-kun Yan, Jun-tao Fu, and Yu-qian Zhang. A new leak location method based on leakage acoustic waves for oil

PUC-Rio - Certificação Digital Nº 1812727/CA

and gas pipelines. *Journal of Loss Prevention in the Process Industries*, 35:236–246, 2015.

[31] S S C Madabhushi, M Z E B Elshafie, and S K Haigh. Accuracy of distributed optical fiber temperature sensing for use in leak detection of subsea pipelines. *Journal of Pipeline Systems Engineering and Practice*, 6(2):04014014, 2015.

[32] Yue Huang, Qiang Wang, Lilian Shi, and Qihua Yang. Underwater gas pipeline leakage source localization by distributed fiber-optic sensing based on particle swarm optimization tuning of the support vector machine. *Applied optics*, 55(2):242–247, 2016.

[33] Junxiao Zhu, Liang Ren, Siu-Chun Ho, Ziguang Jia, and Gangbing Song. Gas pipeline leakage detection based on pzt sensors. *Smart Materials and Structures*, 26(2):025022, 2017.

[34] Cuiwei Liu, Yuxing Li, Liping Fang, Jinke Han, and Minghai Xu. Leakage monitoring research and design for natural gas pipelines based on dynamic pressure waves. *Journal of Process Control*, 50:66–76, 2017.

[35] Juan Li, Qiang Zheng, Zhihong Qian, and Xiaoping Yang. A novel location algorithm for pipeline leakage based on the attenuation of negative pressure wave. *Process Safety and Environmental Protection*, 123:309–316, 2019.

[36] M Cech, P Davis, F Gambardella, A Haskamp, P. Herrero González, M Spence, and J-F Larivé. *Performance of european cross-country oil pipelines: statistical summary of reported spillages in 2016 and since 1971*. CONCAWE, Brussels, BE, 2018 edition edition, 2018.

[37] Tahir Ngada and Kate Bowers. Spatial and temporal analysis of crude oil theft in the niger delta. *Security Journal*, 31(2):501–523, 2018.

[38] Marian Dudek et al. Liquid leak detection focused on theft protection. In *PSIG Annual Meeting*, San Antonio, USA, 2005. Pipeline Simulation Interest Group.

[39] Lotfi A Zadeh. Fuzzy sets. *Information and control*, 8(3):338–353, 1965.

[40] Lotfi A Zadeh. Outline of a new approach to the analysis of complex systems and decision processes. *IEEE Transactions on systems, Man, and Cybernetics*, SMC-3(1):28–44, 1973.

PUC-Rio - Certificação Digital Nº 1812727/CA

[41] Warren S McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133, 1943.

[42] Simon Haykin. *Neural networks and learning machines*. Prentice Hall, Upper Saddle River, NJ, USA, 3rd edition, 2010.

[43] Paul Erdös and Joel Spencer. *Probabilistic methods in combinatorics*, volume 17. Academic Press, New York, NY, USA, 1st edition, 1974.

[44] S Binitha, S Siva Sathya, et al. A survey of bio inspired optimization algorithms. *International journal of soft computing and engineering*, 2(2):137–151, 2012.

[45] John Henry Holland et al. *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. MIT press, Cambridge, MA, USA, 1st edition, 1975.

[46] James David Schaffer. *Some experiments in machine learning using vector evaluated genetic algorithms (artificial intelligence, optimization, adaptation, pattern recognition)*. PhD thesis, Vanderbilt University, Nashville, TN, USA, 1984.

[47] John R Koza and John R Koza. *Genetic programming: on the programming of computers by means of natural selection*, volume 1. MIT press, Cambridge, MA, USA, 1st edition, 1992.

[48] Rainer Storn and Kenneth Price. Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization*, 11(4):341–359, 1997.

[49] Nidamarthi Srinivas and Kalyanmoy Deb. Muiltiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary computation*, 2(3):221–248, 1994.

[50] Kalyanmoy Deb, Samir Agrawal, Amrit Pratap, and Tanaka Meyarivan. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: Nsga-ii. In *International conference on parallel problem solving from nature*, pages 849–858, Paris, FR, 2000. Springer.

[51] Russell Eberhart and James Kennedy. A new optimizer using particle swarm theory. In *MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, pages 39–43, Nagoya, JP, 1995. IEEE.

PUC-Rio - Certificação Digital Nº 1812727/CA

[52] Marco Dorigo and Gianni Di Caro. Ant colony optimization: a new meta-heuristic. In *Proceedings of the 1999 congress on evolutionary computation-CEC99*, volume 2, pages 1470–1477, Washington DC, USA, 1999. IEEE.

[53] Zhonghua Li, Yunong Zhang, and Hong-Zhou Tan. Ia-ais: an improved adaptive artificial immune system applied to complex optimization problems. *Applied Soft Computing*, 11(8):4692–4700, 2011.

[54] Seyedali Mirjalili, Seyed Mohammad Mirjalili, and Andrew Lewis. Grey wolf optimizer. *Advances in engineering software*, 69:46–61, 2014.

[55] Seyedali Mirjalili, Amir H Gandomi, Seyedeh Zahra Mirjalili, Shahrzad Saremi, Hossam Faris, and Seyed Mohammad Mirjalili. Salp swarm algorithm: a bio-inspired optimizer for engineering design problems. *Advances in Engineering Software*, 114:163–191, 2017.

[56] Esmat Rashedi, Hossein Nezamabadi-Pour, and Saeid Saryazdi. Gsa: a gravitational search algorithm. *Information sciences*, 179(13):2232–2248, 2009.

[57] A Kaveh and S Talatahari. A novel heuristic optimization method: charged system search. *Acta Mechanica*, 213(3-4):267–289, 2010.

[58] Tapabrata Ray and Kim-Meow Liew. Society and civilization: An optimization algorithm based on the simulation of social behavior. *IEEE Transactions on Evolutionary Computation*, 7(4):386–396, 2003.

[59] Esmaeil Atashpaz-Gargari and Caro Lucas. Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition. In *2007 IEEE congress on evolutionary computation*, pages 4661–4667, Sigapore, 2007. IEEE.

[60] Amir Ahmadi-Javid. Anarchic society optimization: A human-inspired method. In *2011 IEEE Congress of Evolutionary Computation (CEC)*, pages 2586–2592, New Orleans, USA, 2011. IEEE.

[61] Patrick R Nicolas. *Scala for machine learning*. Packt Publishing Ltd, Birminghan, UK, 1st edition, 2015.

[62] Rosario Toscano and Patrick Lyonnet. A kalman optimization approach for solving some industrial electronics problems. *IEEE Transactions on Industrial Electronics*, 59(11):4456–4464, 2011.

PUC-Rio - Certificação Digital Nº 1812727/CA

[63] Helon V Hultmann Ayala, Leandro dos Santos Coelho, and Gilberto Reynoso-Meza. Heuristic kalman algorithm for multiobjective optimization. *IFAC-PapersOnLine*, 50(1):4460–4465, 2017.

[64] Ibrahim H Osman and Gilbert Laporte. Metaheuristics: A bibliography. *Annals of Operations Research*, 63:511–623, 1996.

[65] Kenneth Sörensen and Fred Glover. Metaheuristics. *Encyclopedia of operations research and management science*, 62:960–970, 2013.

[66] Xin-She Yang and Suash Deb. Multiobjective cuckoo search for design optimization. *Computers & Operations Research*, 40(6):1616–1624, 2013.

[67] William La Cava, Kourosh Danai, Lee Spector, Paul Fleming, Alan Wright, and Matthew Lackner. Automatic identification of wind turbine models using evolutionary multiobjective optimization. *Renewable Energy*, 87:892–902, 2016.

[68] Thang Trung Nguyen and Dieu Ngoc Vo. Modified cuckoo search algorithm for multiobjective short-term hydrothermal scheduling. *Swarm and evolutionary computation*, 37:73–89, 2017.

[69] LV Pavão, CBB Costa, MASS Ravagnani, and L Jiménez. Costs and environmental impacts multi-objective heat exchanger networks synthesis using a meta-heuristic approach. *Applied Energy*, 203:304–320, 2017.

[70] Chuan Li, René-Vinicio Sanchez, Grover Zurita, Mariela Cerrada, Diego Cabrera, and Rafael E Vásquez. Gearbox fault diagnosis based on deep random forest fusion of acoustic and vibratory signals. *Mechanical Systems and Signal Processing*, 76:283–293, 2016.

[71] Mohammadreza Tahan, Masdi Muhammad, and ZA Abdul Karim. A multi-nets ann model for real-time performance-based automatic fault diagnosis of industrial gas turbine engines. *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, 39(7):2865–2876, 2017.

[72] Daniel Winkler, Markus Haltmeier, Manfred Kleidorfer, Wolfgang Rauch, and Franz Tscheikner-Gratl. Pipe failure modelling for water distribution networks using boosted decision trees. *Structure and Infrastructure Engineering*, 14(10):1402–1411, 2018.

[73] Thorsten Wuest, Daniel Weimer, Christopher Irgens, and Klaus-Dieter Thoben. Machine learning in manufacturing: advantages, challenges, and applications. *Production & Manufacturing Research*, 4(1):23–45, 2016.

PUC-Rio - Certificação Digital Nº 1812727/CA

[74] Steven L Brunton, Bernd R Noack, and Petros Koumoutsakos. Machine learning for fluid mechanics. *Annual Review of Fluid Mechanics*, 52:477–508, 2020.

[75] Jingang Guo, Xiaoping Jian, and Guangyu Lin. Performance evaluation of an anti-lock braking system for electric vehicles with a fuzzy sliding mode controller. *Energies*, 7(10):6459–6476, 2014.

[76] Mircea-Bogdan Radac and Radu-Emil Precup. Data-driven model-free slip control of anti-lock braking systems using reinforcement q-learning. *Neurocomputing*, 275:317–329, 2018.

[77] Tariq Amin Khan and Wei Li. Optimal design of plate-fin heat exchanger by combining multi-objective algorithms. *International journal of heat and mass transfer*, 108:1560–1572, 2017.

[78] Shreesh V Dhavle, Anand J Kulkarni, Apoorva Shastri, and Ishaan R Kale. Design and economic optimization of shell-and-tube heat exchanger using cohort intelligence algorithm. *Neural Computing and Applications*, 30(1):111–125, 2018.

[79] Emerson Hochsteiner de Vasconcelos Segundo, Viviana Cocco Mariani, and Leandro dos Santos Coelho. Metaheuristic inspired on owls behavior applied to heat exchangers design. *Thermal Science and Engineering Progress*, 14:100431, 2019.

[80] Zhao Yang, Zhuang Xiong, and Min Shao. A new method of leak location for the natural gas pipeline based on wavelet analysis. *Energy*, 35(9):3814–3820, 2010.

[81] Zhigang Qu, Hao Feng, Zhoumo Zeng, Jingchang Zhuge, and Shijiu Jin. A svm-based pipeline leakage detection and pre-warning system. *Measurement*, 43(4):513–519, 2010.

[82] Santosh Kumar Mandal, Felix TS Chan, and MK Tiwari. Leak detection of pipeline: An integrated approach of rough set theory and artificial bee colony trained svm. *Expert Systems with Applications*, 39(3):3071–3080, 2012.

[83] Jiedi Sun, Qiyang Xiao, Jiangtao Wen, and Fei Wang. Natural gas pipeline small leakage feature extraction and recognition based on lmd envelope spectrum entropy and svm. *Measurement*, 55:434–443, 2014.

PUC-Rio - Certificação Digital Nº 1812727/CA

[84] Kornel Rostek, Łukasz Morytko, and Anna Jankowska. Early detection and prediction of leaks in fluidized-bed boilers using artificial neural networks. *Energy*, 89:914–923, 2015.

[85] Qiyang Xiao, Jian Li, Zhiliang Bai, Jiedi Sun, Nan Zhou, and Zhoumo Zeng. A small leak detection method based on vmd adaptive denoising and ambiguity correlation classification intended for natural gas pipelines. *Sensors*, 16(12):2116, 2016.

[86] Morteza Zadkarami, Mehdi Shahbazian, and Karim Salahshoor. Pipeline leakage detection and isolation: An integrated approach of statistical and wavelet feature extraction with multi-layer perceptron neural network (mlpnn). *Journal of Loss Prevention in the Process Industries*, 43:479–487, 2016.

[87] Morteza Zadkarami, Mehdi Shahbazian, and Karim Salahshoor. Pipeline leak diagnosis based on wavelet and statistical features using dempster–shafer classifier fusion technique. *Process safety and environmental protection*, 105:156–163, 2017.

[88] Mohsen Rahmati, Honeyeh Yazdizadeh, and Alireza Yazdizadeh. Leakage detection in a gas pipeline using artificial neural networks based on wireless sensor network and internet of things. In *2017 IEEE 15th International Conference on Industrial Informatics (INDIN)*, pages 659–664, Emden, DE, 2017. IEEE.

[89] Fatih Kayaalp, Ahmet Zengin, Resul Kara, and Sultan Zavrak. Leakage detection and localization on water transportation pipelines: a multi-label classification approach. *Neural Computing and Applications*, 28(10):2905–2914, 2017.

[90] Zhenlin Li, Haifeng Zhang, Dongjie Tan, Xin Chen, and Hongxiang Lei. A novel acoustic emission detection module for leakage recognition in a gas pipeline valve. *Process Safety and Environmental Protection*, 105:32–40, 2017.

[91] Chuang Wang, Yong Zhang, Jinbo Song, Qingqiang Liu, and Hongli Dong. A novel optimized svm algorithm based on pso with saturation and mixed time-delays for classification of oil pipeline leak detection. *Systems Science & Control Engineering*, 7(1):75–88, 2019.

[92] Ziguang Jia, Siu-Chun Ho, Yang Li, Bo Kong, and Qingmin Hou. Multipoint hoop strain measurement based pipeline leakage localization

PUC-Rio - Certificação Digital Nº 1812727/CA

with an optimized support vector regression approach. *Journal of Loss Prevention in the Process Industries*, 62:103926, 2019.

[93] Guoxi He, Yansong Li, Yuanjie Huang, Liying Sun, and Kexi Liao. A framework of smart pipeline system and its application on multiproduct pipeline leakage handling. *Energy*, 188:116031, 2019.

[94] Xu Diao, Guodong Shen, Juncheng Jiang, Qiang Chen, Zhirong Wang, Lei Ni, Ahmed Mebarki, and Zhan Dou. Leak detection and location in liquid pipelines by analyzing the first transient pressure wave with unsteady friction. *Journal of Loss Prevention in the Process Industries*, 60:303–310, 2019.

[95] Jinhai Liu, Dong Zang, Chen Liu, Yanjuan Ma, and Mingrui Fu. A leak detection method for oil pipeline based on markov feature and two-stage decision scheme. *Measurement*, 138:433–445, 2019.

[96] Yuri Nagase, Yuta Sugiyama, Shiro Kubota, Tei Saburi, and Akiko Matsuo. Prediction model of the flow properties inside a tube during hydrogen leakage. *Journal of Loss Prevention in the Process Industries*, 62:103955, 2019.

[97] Junyao Xie, Xiaodong Xu, and Stevan Dubljevic. Long range pipeline leak detection and localization using discrete observer and support vector machine. *AIChE Journal*, 65(7):e16532, 2019.

[98] Aaron E Maxwell, Timothy A Warner, and Fang Fang. Implementation of machine-learning classification in remote sensing: An applied review. *International Journal of Remote Sensing*, 39(9):2784–2817, 2018.

[99] Xiongmin Li and Christine W Chan. Application of an enhanced decision tree learning approach for prediction of petroleum production. *Engineering Applications of Artificial Intelligence*, 23(1):102–109, 2010.

[100] Alexandros Bousdekis, Babis Magoutas, Dimitris Apostolou, and Gregoris Mentzas. Review, analysis and synthesis of prognostic-based decision support methods for condition based maintenance. *Journal of Intelligent Manufacturing*, 29(6):1303–1316, 2018.

[101] Mohammad Sabah, Mohsen Talebkeikhah, Farough Agin, Farzaneh Talebkeikhah, and Erfan Hasheminasab. Application of decision tree, artificial neural networks, and adaptive neuro-fuzzy inference system on predicting lost circulation: A case study from marun oil field. *Journal of Petroleum Science and Engineering*, 177:236–249, 2019.

PUC-Rio - Certificação Digital Nº 1812727/CA

[102] Mengxin Song, Xiangguang Zhou, et al. A casing damage prediction method based on principal component analysis and gradient boosting decision tree algorithm. In *SPE Middle East Oil and Gas Show and Conference*, Manama, BH, 2019. Society of Petroleum Engineers.

[103] Mauricio Barrios Castellanos, Alberto Luiz Serpa, Jorge Luiz Biazussi, William Monte Verde, and Natache do Socorro Dias Arrifano Sassim. Fault identification using a chain of decision trees in an electrical submersible pump operating in a liquid-gas flow. *Journal of Petroleum Science and Engineering*, 184:106490, 2020.

[104] Paloma S Prata, Guilherme L Alexandrino, Noroska Gabriela S Mogollón, and Fabio Augusto. Discriminating brazilian crude oils using comprehensive two-dimensional gas chromatography–mass spectrometry and multiway principal component analysis. *Journal of Chromatography A*, 1472:99–106, 2016.

[105] Zhenhua Rui, Jun Lu, Zhien Zhang, Rui Guo, Kegang Ling, Ronglei Zhang, and Shirish Patil. A quantitative oil and gas reservoir evaluation system for development. *Journal of Natural Gas Science and Engineering*, 42:31–39, 2017.

[106] Byeongcheol Kang, Hyungjun Yang, Kyungbook Lee, and Jonggeun Choe. Ensemble kalman filter with principal component analysis assisted sampling for channelized reservoir characterization. *Journal of Energy Resources Technology*, 139(3), 2017.

[107] Sin Yong Teng, Bing Shen How, Wei Dong Leong, Jun Hao Teoh, Adrian Chee Siang Cheah, Zahra Motavasel, and Hon Loong Lam. Principal component analysis-aided statistical process optimisation (paspo) for process improvement in industrial refineries. *Journal of cleaner production*, 225:359–375, 2019.

[108] Pedro H L S P Domingues, Roberto Z Freire, Leandro dos S Coelho, and Helon V H Ayala. Bio-inspired multiojective tuning of pid-controlled antilock braking systems. In *2019 IEEE Congress on Evolutionary Computation (CEC)*, pages 888–895, Wellington, NZ, 2019. IEEE.

[109] F P Da Costa, P H L S P Domingues, Roberto Z Freire, Leandro S Coelho, A R Tavakolpour-Saleh, and Helon V H Ayala. Genetic algorithm for topology optimization of an artificial neural network applied to aircraft turbojet engine identification. In *2019 IEEE Congress on*

PUC-Rio - Certificação Digital Nº 1812727/CA

Evolutionary Computation (CEC), pages 1095–1101, Wellington, NZ, 2019. IEEE.

[110] Pedro Henrique Domingues, Leandro Coelho, Roberto Zanetti Freire, and Helon Vicente Hultmann Ayala. Multiobjective optimization of heat exchanger design through heuristic kalman algorithm. In *25th International Congress of Mechanical Engineering*, Uberlandia, BR, 2019. ABCM.

[111] Victor Henrique Alves Ribeiro, Pedro Henrique Domingues, Paulo Rodrigo Cavalin, Gilberto Reynoso-Meza, Helon Vicente Hultmann Ayala, and Luis Fernando Alzuguir Azevedo. Dynamic multi-criteria classifier selection for illegal tapping detection in oil pipelines. In *2020 International Joint Conference on Neural Networks (IJCNN 2020)*, Glasgow, UK, (accepted) 2020.

[112] Eckart Zitzler. *Evolutionary algorithms for multiobjective optimization: methods and applications*. PhD thesis, Swiss Federal Institute of Technology Zurich, Zurich, SWI, 1999.

[113] W Stanley Jevons. Untitled review of mathematical psychics by fy edgeworth. *Mind*, 6:581–583, 1881.

[114] H.L. Moore. Cours d'Économie Politique. By VILFREDO PARETO, Professeur à l'Université de Lausanne. Vol. I. Pp. 430. I896. Vol. II. Pp. 426. I897. Lausanne: F. Rouge. *The ANNALS of the American Academy of Political and Social Science*, 9(3):128–131, nov 1897.

[115] Q Zhang, A Zhou, S Zhao, P N Suganthan, W Liu, and S Tiwari. Multiobjective optimization test instances for the cec 2009 special session and competition. *School Comput. Sci. Electron. Eng., Univ. Essex, Colchester, U.K., and School Electr. Electron. Eng., Nanyang Technol. Univ., Singapore, Tech. Rep. CES-487*, 2009.

[116] M R Sierra and C A C Coello. Improving pso-based multi-objective optimization using crowding, mutation and ϵ-dominance. In *Third International Conference on evolutionary multi-criterion optimization*, pages 505–519, Guanajuato, MX, 2005. Springer.

[117] Jerrold H Zar. *Biostatistical analysis*. Pearson Education India, 1999.

[118] T Pohlert. *The pairwise multiple comparison of mean ranks package (PMCMR)*, 2014. R package.

PUC-Rio - Certificação Digital Nº 1812727/CA

[119] R Core Team. *R: A language and environment for statistical computing*. R Foundation for Statistical Computing, Vienna, Austria, 2018.

[120] Panos M Pardalos, Antanas Žilinskas, and Julius Žilinskas. A brief review of non-convex single-objective optimization. In *Non-Convex Multi-Objective Optimization*, pages 33–42. Springer, 2017.

[121] David E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1st edition, 1989.

[122] Larry P Madin. Aspects of jet propulsion in salps. *Canadian Journal of Zoology*, 68(4):765–777, 1990.

[123] P A V Anderson and Quentin Bone. Communication between individuals in salp chains. ii. physiology. *Proceedings of the Royal Society of London. Series B. Biological Sciences*, 210(1181):559–574, 1980.

[124] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The elements of statistical learning: data mining, inference, and prediction*. Springer-Verlag New York, 2nd edition, 2009.

[125] Karl Pearson. Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572, 1901.

[126] Steven L Brunton and J Nathan Kutz. *Data-driven science and engineering: Machine learning, dynamical systems, and control*. Cambridge University Press, 1st edition, 2019.

[127] Gene H Golub and Christian Reinsch. Singular value decomposition and least squares solutions. In *Linear Algebra*, pages 134–151. Springer, 1971.

[128] Donghyun Kim, Chulsoo Kim, Sungho Hwang, and Hyunsoo Kim. Hardware in the loop simulation of vehicle stability control using regenerative braking and electro hydraulic brake for hybrid electric vehicle. *IFAC Proceedings Volumes*, 41(2):5664–5669, 2008.

[129] Gilberto Reynoso-Meza, Javier Sanchis, Xavier Blasco, and Miguel Martínez. Preference driven multi-objective optimization design procedure for industrial controller tuning. *Information Sciences*, 339:108–131, 2016.

PUC-Rio - Certificação Digital Nº 1812727/CA

[130] Gilberto Reynoso-Meza, Javier Sanchis, Xavier Blasco, and Roberto Z Freire. Evolutionary multi-objective optimisation with preferences for multivariable pi controller tuning. *Expert Systems with Applications*, 51:120–133, 2016.

[131] Jesús Velasco Carrau, Gilberto Reynoso-Meza, Sergio García-Nieto, and Xavier Blasco. Enhancing controller's tuning reliability with multi-objective optimisation: From model in the loop to hardware in the loop. *Engineering Applications of Artificial Intelligence*, 64:52–66, 2017.

[132] Karl Johan Åström and Richard M Murray. *Feedback systems: an introduction for scientists and engineers*. Princeton university press, Princeton, NJ, USA, 1st edition, 2010.

[133] Ramón Vilanova and Víctor M Alfaro. Control pid robusto: Una visión panorámica. *Revista Iberoamericana de Automática e Informática Industrial RIAI*, 8(3):141–158, 2011.

[134] Gilberto Reynoso-Meza, Xavier Blasco, Javier Sanchis, and Miguel Martínez. Controller tuning using evolutionary multi-objective optimisation: current trends and applications. *Control Engineering Practice*, 28:58–73, 2014.

[135] Karl Johan Åström, Tore Hägglund, and Karl J Astrom. *Advanced PID control*, volume 461. ISA, Research Triangle Park, NC, USA, 1st edition, 2006.

[136] Seibum B Choi. Antilock brake system with a continuous wheel slip control to maximize the braking performance and the ride quality. *IEEE Transactions on Control Systems Technology*, 16(5):996–1003, 2008.

[137] Sedigheh Mahdavi, Shahryar Rahnamayan, and Kalyanmoy Deb. Opposition based learning: A literature review. *Swarm and evolutionary computation*, 39:1–23, 2018.

[138] Shahryar Rahnamayan, Hamid R Tizhoosh, and Magdy MA Salama. Opposition-based differential evolution algorithms. In *2006 IEEE International Conference on Evolutionary Computation*, pages 2010–2017, Vancouver, CA, 2006. IEEE.

[139] Mario Ventresca and Hamid R Tizhoosh. A diversity maintaining population-based incremental learning algorithm. *Information Sciences*, 178(21):4038–4056, 2008.

PUC-Rio - Certificação Digital Nº 1812727/CA

[140] Fengxue Zhang, Chunhua Yang, Xiaojun Zhou, and Hongqiu Zhu. Fractional order fuzzy pid optimal control in copper removal process of zinc hydrometallurgy. *Hydrometallurgy*, 178:60–76, 2018.

[141] Sepehr Sanaye and Hassan Hajabdollahi. Multi-objective optimization of shell and tube heat exchangers. *Applied Thermal Engineering*, 30(14-15):1937–1945, 2010.

[142] Helon Vicente Hultmann Ayala, Patrick Keller, Márcia de Fátima Morais, Viviana Cocco Mariani, Leandro dos Santos Coelho, and Ravipudi Venkata Rao. Design of heat exchangers using a novel multiobjective free search differential evolution paradigm. *Applied Thermal Engineering*, 94:170–177, 2016.

[143] William Morrow Kays and Alexander Louis London. *Compact heat exchangers*. McGraw-Hill, New York, NY, USA, 1st edition, 1984.

[144] Kalyanmoy Deb and Himanshu Jain. An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: solving problems with box constraints. *IEEE transactions on evolutionary computation*, 18(4):577–601, 2013.

[145] Indraneel Das and John E Dennis. Normal-boundary intersection: A new method for generating the pareto surface in nonlinear multicriteria optimization problems. *SIAM journal on optimization*, 8(3):631–657, 1998.

[146] Jyotsna K Mandal, Somnath Mukhopadhyay, and Paramartha Dutta. *Multi-Objective Optimization: Evolutionary to Hybrid Framework*. Springer Singapore, Singapore, 1st edition, 2018.

[147] Yujie Ying, James H Garrett Jr, Irving J Oppenheim, Lucio Soibelman, Joel B Harley, Jun Shi, and Yuanwei Jin. Toward data-driven structural health monitoring: application of machine learning and signal processing to damage detection. *Journal of Computing in Civil Engineering*, 27(6):667–680, 2013.

[148] Aboozar Taherkhani, Georgina Cosma, Ali A Alani, and TM McGinnity. Activity recognition from multi-modal sensor data using a deep convolutional neural network. In *Science and Information Conference*, pages 203–218. Springer, 2018.

PUC-Rio - Certificação Digital Nº 1812727/CA

[149] Bing Xue, Mengjie Zhang, Will N Browne, and Xin Yao. A survey on evolutionary computation approaches to feature selection. *IEEE Transactions on Evolutionary Computation*, 20(4):606–626, 2015.

[150] Chris Seiffert, Taghi M Khoshgoftaar, Jason Van Hulse, and Amri Napolitano. Rusboost: a hybrid approach to alleviating class imbalance. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 40(1):185–197, 2009.

PUC-Rio - Certificação Digital Nº 1812727/CA