Multi-objective meta-heuristic optimization in intelligent control: a survey on the controller tuning problem

Alejandro Rodríguez-Molina[a,b,*], Efrén Mezura-Montes[c], Miguel G. Villarreal-Cervantes[b], Mario Aldape-Pérez[b]

[a] *Tecnológico Nacional de México/IT de Tlalnepantla, Research and Postgraduate Division, Estado de México, 54070, México.*
[b] *Posgraduate Department, Instituto Politécnico Nacional, CIDETEC, Mexico City, 07700, Mexico*
[c] *Artificial Intelligence Research Center, University of Veracruz, Xalapa Veracruz, 91000, Mexico*

Abstract

Multi-objective optimization has been adopted in many engineering problems where a set of requirements must be met to generate successful applications. Among them, there are the tuning problems from control engineering, which are focused on the correct setting of the controller parameters to properly govern complex dynamic systems to satisfy desired behaviors such as high accuracy, efficient energy consumption, low cost, among others. These requirements are stated in a multi-objective optimization problem to find the most suitable controller parameters. Nevertheless, these parameters are tough to find because of the conflicting control performance requirements (i.e., a requirement cannot be met without harming the others). Hence, the use of techniques from computational intelligence and soft computing is necessary to solve multi-objective problems and handle the trade-offs among control performance objectives. Meta-heuristics have shown to obtain outstanding results when solving complex multi-objective problems at a reasonable computational cost. In this survey, the literature related to the use of multi-objective meta-heuristics in intelligent control focused on the controller tuning problem is reviewed and discussed.

Keywords: Meta-heuristics, multi-objective optimization, controller tuning, intelligent control

Declarations of interest: none

*Corresponding author
Email addresses: alejandro.rm@tlalnepantla.tecnm.mx (Alejandro Rodríguez-Molina), emezura@uv.mx (Efrén Mezura-Montes), mvillarrealc@ipn.mx (Miguel G. Villarreal-Cervantes), maldape@ipn.mx (Mario Aldape-Pérez)

1. Introduction

Dynamic systems are found in problems from many contexts such as financial [1], medical [2], mechanical [3], chemical [4], electrical [5], and social [6]. These systems are required to have desirable behaviors to generate useful applications. These behaviors are achieved by using a suitable control strategy.

The analysis of dynamic systems to design successful controllers is concerned to control engineering. Since the emergence of PID control by Elmer Sperry in 1910, there have been more concerns about tuning procedures that guarantee

---

Abbreviations:

DE Differential Evolution

FOPID Fractional Order Proportional Integral Derivative

GA Genetic Algorithm

GM Gain Margin

HV Hyper-volume

IAE Integral Absolute Error

ISDCO Integral of the Squared Deviation of Controller Output

ISE Integral Squared Error

ITAE Integral Time-weighted Absolute Error

ITSE Integral Time-weighted Squared Error

LM Log-Modulus

MCS Maximum Complementary Sensitivity

MHO Meta-heuristic Optimizer

MIMO Multi-input Multi-output

MOMHOCTP Multi-objective Meta-heuristic Optimization Controller Tuning Process

MODE Multi-objective Differential Evolution

MOGA Multi-objective Genetic Algorithm

MOMHO Multi-objective Meta-heuristic Optimizer

MOP Multi-Objective Problem

MOPSO Multi-objective Particle Swarm Optimization

MS Maximum Sensitivity

NSGA-II Non-dominated Sorting Genetic Algorithm II

OS Overshoot

PID Proportional Integral Derivative

PM Phase Margin

PSO Particle Swarm Optimization

RT Rise time

SISO Single-input Single-output

SSE Steady State Error

ST Settling time

TV Total variation of the control signal the best performance of the PID controller. In this, the controller parameters, which compromise the overall operation and performance of a dynamic system, must be properly set. Ziegler and Nichols introduced the first tuning rules in 1942 [7], and after that, several tuning rules have been proposed. Nevertheless, some of them have been focused on stabilizing linear systems [8, 9], and they are not suitable for nonlinear dynamic systems. Other tuning approaches involve nonlinear systems [10] and aim to obtain the control parameters that fulfill the stability conditions. However, the latter approach does not guarantee specific response characteristics. Consequently, one of the main problems in this control engineering area is controller tuning.

Controller tuning has been addressed by using different methods which can be classified as in [11]:

- Analytical methods where different tools from control theory are used to find the controller parameters by analyzing the closed-loop system stability. For instance, for linear systems [8, 9] described by their frequency response, root locus method, etc. and for nonlinear systems [10] from the Lyapunov stability analysis.
- Heuristic methods in which the controller parameters are manually chosen based on empirical knowledge of an expert designer that uses the information of the controlled variable measurements to establish proper parameter-performance relationships. Ziegler-Nichols [12] and Cohen-Coon [13] are two widely used heuristic methods.
- Optimization methods in which a mathematical programming problem that takes into account different performance criteria is stated and then solved by an optimizer to find a proper set of controller parameters. The reviewed works in this study fall into this class.
- Adaptive tuning methods obtain the controller parameters online by performing an identification together with one of the above methods.

The difficulty in tuning controllers is related to the complexity of the dynamic system to be governed [14]. In many cases, it can have structures that include highly nonlinear behaviors [15], a large number of tunable variables [16], and a large number of inputs and outputs [17]. Additionally, they can be subject to environmental or operational conditions such as uncertainties and disturbances, and several real-world limitations. These characteristics make the search harder for suitable controller parameters that achieve the desired performance.

Moreover, current applications demand meeting several performance specifications at a time, and they are usually in conflict [18]. They include but are not limited to high accuracy, efficient energy consumption, low cost, among others. The above also means that different controller parameter settings imply different satisfaction levels of the demands. Therefore, the tuning process becomes a non-trivial task.

Fortunately, the above multiple-demand controller tuning problem can be treated as a multi-objective mathematical programming one, which in turn can be handled by optimization or adaptive-optimization tuning methods. This approach allows using multi-objective optimization techniques to find a set of reliable controller configurations with different trade-offs (regarding their compliance with the tuning performance specifications). In this way, based on its preferences, the designer or decision-maker can choose the configuration alternative that best suits the demands of the application for its further implementation.

Due to the complexity of the multi-objective mathematical programming problem, also known as Multi-Objective Problem (MOP), in the controller tuning task, the use of computational intelligence or soft computing techniques must be necessary [19]. The use of these techniques in control engineering (in the context of multi-objective optimization and beyond) is defined by Ruano [20] as intelligent control. It conforms to a growing area of interest for research.

Among intelligent techniques, meta-heuristic optimizers have been widely used to solve many real-world optimization problems, as observed in [21, 22, 23]. This popularity is due to their parallelizable and relatively simple operation, their capability of handling very complex problems at a reasonable computational cost, and their applicability to a wide variety of contexts [24]. The above makes meta-heuristics suitable alternatives to handle the controller tuning problem.

In [25] and [26], the tuning approaches adopted for several control problems are analyzed with a particular focus on the optimization problem formulations, the used controller structures, and the controlled dynamic systems. In all cases, meta-heuristics have had outstanding advantages in controller tuning. The variety of available dynamic systems and controller structures entail to an extensive range of different tuning MOPs whose characteristics require the use of different types of meta-heuristics with particular search mechanisms to find the most suitable controller configurations. Hence, unlike the reported in [25] and [26], the present work is concerned to the review of full steps of the controller tuning process, from abstraction to validation, analyzing the addressed dynamic systems and controller structures, paying particular attention in the formulation of the related multi-objective optimization problems, emphasizing the adopted multi-objective meta-heuristic techniques and their search mechanisms, and observing the used decision-making approaches. The present work studies research items published from 2001 to 2019.

It is important to mention that several works in the specialized literature adopt an *a priori* preference articulation approach. In this way, a single-objective is constructed as a composition of the objectives in the MOP by establishing the importance of each objective based on higher-level preference information [27, 28, 29, 30, 31, 32, 33, 34, 35]. Then, a single-objective optimizer can be used to solve the new problem and obtain a single preferred controller parameter configuration. These works are out of the scope of the present survey.

The rest of this document is organized as follows. Sections 2 and 3 give a background about multi-objective optimization through meta-heuristic techniques and multi-objective controller tuning, respectively. The general controller tuning process is described in Section 4 with the support of the surveyed research items. The conclusions and future directions derived from this survey are drawn in Section 5.

2. Multi-objective optimization - an overview

2.1. The multi-objective optimization problem

A MOP is stated as in (1), where a design variable vector $\vec{p} = [x_1, \ldots, x_d]^T$ must be found to minimize a vector $\vec{F}$ of $m > 1$ objective functions which are in conflict each other. The MOP is also known as a many-objective optimization problem when it has more than three objectives, i.e., $m > 3$. In many real-world applications, the MOP is subject to several constraints in the form of $g_i(\vec{p})$ and $h_j(\vec{p})$ that are named respectively inequality and equality constraints. Additionally, the design variables can be bounded, i.e., $p_k \in [p_k^{min}, p_k^{max}]$; all these constraints are related to the real-world limitations (e.g. energy and sizing, among others). The space of feasible solutions (where all constraints are satisfied) is denoted by Ω.

$$\begin{aligned} &\mathbf{min}\ \vec{F}(\vec{p}) = [f_1(\vec{p}), \ldots, f_m(\vec{p})]^T \\ &\text{subject to:} \\ &\quad g_i(\vec{p}) \leq 0, i = 1, \ldots, n_g \\ &\quad h_j(\vec{p}) = 0, j = 1, \ldots, n_h \\ &p_k^{min} \leq p_k \leq p_k^{max}, k = 1, \ldots, d \end{aligned} \tag{1}$$

Definition 1. *(Pareto dominance, [36]).*
A vector $\vec{F}(\vec{p}) = [f_1(\vec{p}), \ldots, f_m(\vec{p})]^T$ is said to dominate $\vec{F}(\vec{q}) = [f_1(\vec{q}), \ldots, f_m(\vec{q})]^T$ (denoted by $\vec{F}(\vec{p}) \preceq \vec{F}(\vec{q})$) if and only if $\vec{F}(\vec{p})$ is as good as $\vec{F}(\vec{q})$ for all the objectives, i.e., $f_i(\vec{p}) \leq f_i(\vec{q}), \forall i \in \{1, \ldots, m\}$, and for at least one objective $f_i(\vec{p}) < f_i(\vec{q})$.

Definition 2. *(Pareto optimality, [36]).*
A decision vector $\vec{p} \in \Omega$ is a Pareto optimal if no objective function $f_i(\vec{p})$ can be improved without worsening the rest, i.e., $\nexists \vec{q} \in \Omega$ such that $\vec{F}(\vec{q}) \preceq \vec{F}(\vec{p})$.

Definition 3. *(Pareto optimal set, [36]).*
The Pareto optimal set $\mathcal{P}^*$ contains every possible optimal decision vector $\vec{p} \in \Omega$, i.e., $\mathcal{P}^* = \left\{\vec{p} \in \Omega \mid \nexists \vec{q} \in \Omega,\ \vec{F}(\vec{q}) \preceq \vec{F}(\vec{p})\right\}$.

Definition 4. *(Pareto front, [36]).*
The Pareto front, also named true Pareto front, contains the evaluated objective values of the vectors in $\mathcal{P}^*$, i.e., $\mathcal{PF}^* = \{\vec{F}(\vec{p}) \mid \vec{p} \in \mathcal{P}^*\}$.

2.2. Multi-objective meta-heuristic optimizers

Since there is no conventional definition of Meta-heuristic Optimizer (MHO) [37], in this work, it is conceived as a stochastic computational technique that can find competitive solutions to optimization problems at a reasonable computational cost. Many MHOs are inspired by different biological processes, such as natural evolution. Among the advantages of using these techniques is their capability to solve very complex problems without requiring additional information of them, which many times is difficult to acquire (e.g., derivatives); the problem with these techniques is that they cannot guarantee the feasibility nor the optimality of solutions. Still, for many real-world problems, a solution close to the optimal one (even when such a solution cannot be found) is useful enough.

In a MOP, several solutions represent different trade-offs among the objectives. Because of this, a Multi-objective Meta-heuristic Optimizer (MOMHO) must be capable of finding as many different trade-offs as possible that allow the decision-maker to choose the most appropriate one for a given situation.

2.3. Desirable features of the Pareto front approximation

It is important to remark that a MOMHO cannot guarantee to obtain the true Pareto front $\mathcal{PF}^*$ but an approximation denoted by $\mathcal{PF}^A$. Therefore, the quality of this approximation is an important criterion to take into account when selecting a MOMHO to solve a particular problem.

Some desirable features that allow observing the quality of the Pareto front approximation obtained with a MOMHO are presented below.

1. **Capacity** [38]: This feature is related to the number of non dominated solutions in the $\mathcal{PF}^A$. The more solutions, the more possibilities are to find a proper trade-off among the objectives. Fig. 1 shows two approximated fronts in $\mathbb{R}^2$ with a different number of solutions.
2. **Convergence** [39]: It is preferred that the obtained solutions in the $\mathcal{PF}^A$ are near to $\mathcal{PF}^*$. It is important to highlight that for almost all real-world MOPs; the real Pareto front is unknown. Fig. 2 presents two fronts in $\mathbb{R}^2$ with different convergence levels.
3. **Diversity** [39]: It is highly desirable that the found trade-offs are not very similar among them. In other words, the solutions in the $\mathcal{PF}^A$ must be as well distributed as possible, i.e., they must be uniformly separated and spread along the $\mathcal{PF}^*$. In Fig. 3, two fronts in $\mathbb{R}^2$ with different spacing among solutions are presented.
4. **Pertinence** [39]: When some desired regions of the objective function space are well established (i.e., the designer *a priori* knows where the most promising solutions are located), a $\mathcal{PF}^A$ that is located within these regions is preferred. Fig. 4 shows two fronts in $\mathbb{R}^2$ and their relation with a preferred region.

Different performance metrics provide a quantitative value of an approximation quality [38]. Using these metrics allows the performance comparison among MOMHOs [40]. Some of the most popular metrics to compare MOMHOs are:

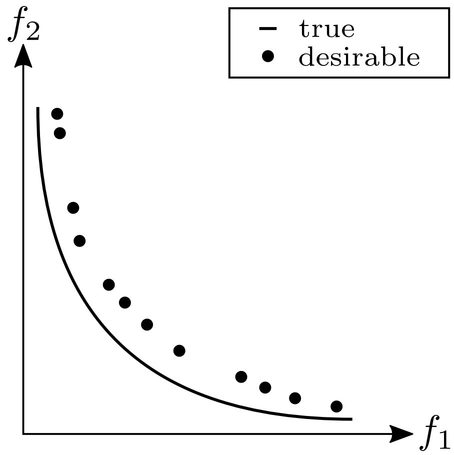

Figure 1: Comparison of two different Pareto front approximations in terms of capacity. The front on the left has more solutions than the one on the right, then the first is more desirable.

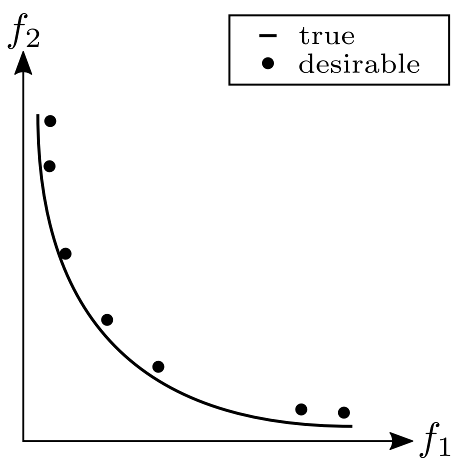

Figure 2: Comparison of two different Pareto front approximations in terms of convergence. The solutions of the front on the left are closer to the true Pareto front than the ones of the front on the right, then the first is more desirable.

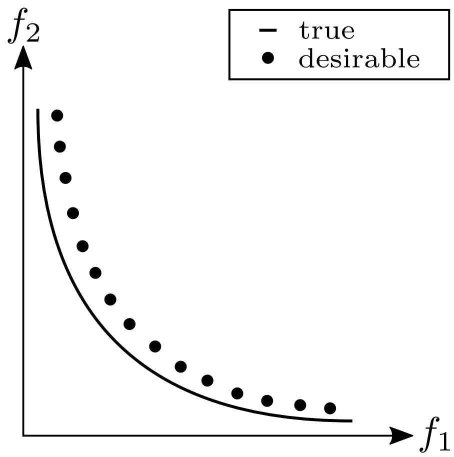

Figure 3: Comparison of two different Pareto front approximations in terms of diversity. The solutions of the front on the left are better distributed than the ones of the front on the right, then the first is more desirable.

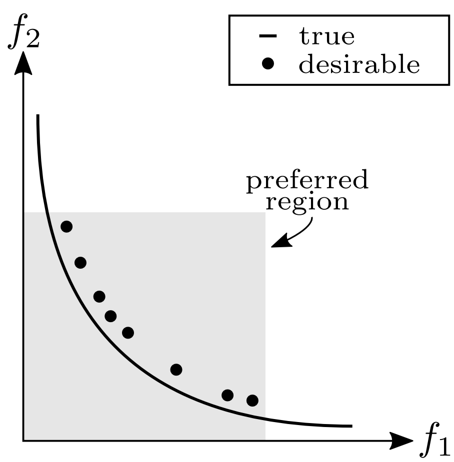

Figure 4: Comparison of two different Pareto front approximations in terms of pertinence. The solutions of the front on the left are all in the preferred region unlike the ones of the front on the right, then the first is more desirable.

- Overall non-dominated vector generation (ONVG) [41]: Gives the number of non-dominated solutions in the $\mathcal{PF}^A$.
- Generational distance (GD) [42]: Provides the degree of proximity based on the distance between the $\mathcal{PF}^A$ and a subset of the $\mathcal{PF}^*$.
- Hyper-volume (HV) [43]: Given a reference point, this metric measures both, the closeness to the $\mathcal{PF}^*$ and the diversity among the solutions in the $\mathcal{PF}^A$.
- Spacing (SP) [44]: Calculates the closest distance of pairwise solutions in the $\mathcal{PF}^A$.
- Two-set coverage (C-metric) [45]: It is a binary metric that measures the coverage proportion regarding the number of dominated solutions between two different sets $\mathcal{PF}^A_1$ and $\mathcal{PF}^A_2$.
- Metric $\triangle$ ($\triangle$) [46]: Measures the extension of the spread achieved among non-dominated solutions in the $\mathcal{PF}^A$.
- Error ratio (ER) [44]: Considers the error in the intersection of the $\mathcal{PF}^A$ and a subset of the $\mathcal{PF}^*$.
- Binary epsilon indicator (I_ϵ) [47]: Gives the value of ϵ required to translate/scale a subset of the $\mathcal{PF}^*$ to be dominated by the $\mathcal{PF}^A$.
- Averaged Hausdorff Distances [48]: Measure the distance between two approximations $\mathcal{PF}^A_1$ and $\mathcal{PF}^A_2$.
- Inverted Generational Distance (IGD) [49]: Determines the average distance from a $\mathcal{PF}^A$ to a reference set.

Although the above metrics are found recurrently in the specialized literature, many other alternatives can be useful in multi-objective optimization tasks. Some of these alternatives can be found in [50].

All the above metrics can help to know the effectiveness of a MOMHO in search of solutions for a given MOP and then could provide enough evidence for its use in a real scenario. According to the multi-objective 'No Free Lunch' and 'Free Leftovers' theorems [51], it may be necessary to compare a reasonable number of MOMHO using different metrics to opt for a single alternative.

2.4. Meta-heuristic search approaches

Some important features of the $\mathcal{PF}^A$ are described above; now it concerns to review the mechanisms adopted by the MOMHOs to obtain this approximation.

Many successful meta-heuristics have been developed over time. They differ from each other by the operators they use to interrelate the solutions from an initial set and generate improved alternatives.

There are three recurrently used meta-heuristics in controller tuning, these are the Genetic Algorithm (GA) [52], Differential Evolution (DE) [53] and Particle Swarm Optimization (PSO) [54].

Other popular and relatively recent meta-heuristics, which can be less frequently found in several controller tuning works, are the Bat Algorithm (BA) [55], the Firefly Algorithm (FA) [56], the Cuckoo Search (CS) [57] and the Harmony Search (HS) [58].

The above mentioned meta-heuristic techniques are all designed initially for solving global optimization problems. By including a multi-objective search mechanism, they can extend their operation to the solution of MOPs.

The multi-objective search mechanisms used by the MOMHOs can be classified in the following groups:

1. **Dominance-based**: The search of the $\mathcal{PF}^A$ is guided based on the Pareto dominance, i.e., Pareto optimality is used to find the best trade-off among candidate solutions. In this case, a solution that dominates another or a solution that is less dominated by a solution set is preferred and must persist. Some representative MOMHOs under this approach are the Multi-objective Genetic Algorithm (MOGA) [59], Non-dominated Sorting Genetic Algorithm II (NSGA-II) [60], Multi-objective Particle Swarm Optimization (MOPSO) [61] and Multi-objective Differential Evolution (MODE) [62].
2. **Decomposition-based**: In this approach, the MOP is decomposed into a finite number of scalar optimization sub-problems (single-objective problems) which are optimized simultaneously. MOEA/D [63] is a well known decomposition-based MOMHO.
3. **Metric-driven**: In this class of MOMHOs, a performance metric is used to select potentially good solutions from a candidate solution set. Some examples of MOMHOs are I-SIBEA [58] and HypE [64].
4. **Hybrid**: The search in this approach includes features from the above three [65]. A representative alternative from this class is the Non-dominated Sorting Genetic Algorithm III (NSGA-III) [65].

2.5. Decision making

The decision-making stage is related to the selection of the best trade-off for its later implementation, depending on the application necessities or preferences (desired or expected values of the objective functions).

There are three ways to handle these preferences [36]:

1. ***A priori***: The decision making is performed before searching for solutions. Using well-established preferences about the objectives is possible to define an aggregate objective function that indicates the pertinence degree of solutions. In some cases, the MOP can be transformed into a single-objective problem whose global solution includes the desired trade-off.

   A posteriori: The search process is performed before making decisions. Once the Pareto front approximation is obtained, the decision-maker can select/sample the best trade-off. Currently, there is a vast diversity of *a posteriori* decision-makers among which stand out those based on the position of solutions [66], on the closeness to a well-established reference point [67], on the value of a utility function that ponders the preference of each objective [68], or on the subjective analysis of a human designer through the visualization of the $\mathcal{PF}^A$ combined with his/her experience [69].

Progressive/Interactive: The preferences are incorporated during the search process (these could be altered over time using the search knowledge). The decision-maker continuously provides information about preferences to guide the search. Fuzzy rules [70] and neural networks [71] have been successfully used to progressively distinguish and promote solutions that belong to a preferable region of interest during the search. Other popular interactive approaches update a reference point automatically [72, 73]. An additional progressive approach displays the $\mathcal{PF}^A$ to allow the expert designer to update the preference criteria for each search iteration [74].

Regardless of the used way of preference handling, several mechanisms allow the decision making, and they can be summarized into two categories:

- **Empirical**: The designer experience plays the leading role in decision-making. This category is sometimes related to a human-supported decision. Therefore, the human decision-maker can use visualization methods [69], a trial and error approach, and subjective analysis of objectives to select the best trade-off solution.
- **Utility-based**: A utility function is used to determine the closeness of a solution to a preferred trade-off (or a preferred region of the objective function space). This kind of preference handling is related to automated decision making. Some well-known mechanisms are the weighted sum, goal programming, indicator-based, reference point-based, and physical programming methods [36, 75]. Position-based approaches are also considered in this category, i.e., the best trade-off solution is selected considering its position in the Pareto front approximation (e.g., knee, average and extreme solutions concerning one particular objective).

3. The controller tuning problem

3.1. The general controller tuning problem

As described before, controllers are elements used to govern the behavior of dynamic systems that generate successful engineering applications. The main goal of the controllers is then to guarantee the stability of a dynamic system response.

Although the controllers are theoretically designed to guarantee stability, they have a set of parameters that determine how systems stabilize. These parameters can be adjusted to fit different performance conditions such as speeding up the stabilization time [76], allowing recovery of stability under uncertainties

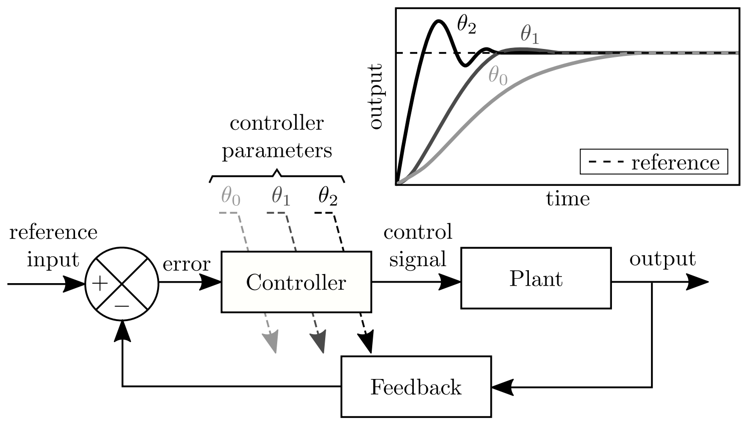

Figure 5: Control performances achieved by different controller parameters.

or disturbances [77], smoothing the transition to stability [78], to name but a few.

Fig. 5 illustrates how different parameter sets, denoted by θ_0, θ_1, and θ_2, used in the same controller structure, affect the stabilization towards a reference of the dynamic system (also known as the plant).

The correct establishment of the controller parameters is usually a difficult task, known as controller tuning, and depends on the application necessities. The general controller tuning problem refers to the search for a set of controller parameters that stabilizes the dynamic system response under an established performance criterion that fits the above necessities (performance conditions).

3.2. The multi-objective controller tuning problem

Engineering applications have become more demanding over time, in the sense that they require dynamic systems that satisfy different and conflicting necessities (i.e., one necessity cannot be fulfilled entirely without harming the fulfillment of the others). Then, the tuning of the controllers required to govern those dynamic systems becomes a harder task.

Nevertheless, this modern trend of engineering applications aligns naturally with the theory of multi-objective optimization summarized in Section 2. Without losing generality, Fig. 6 shows the usual steps for the multi-objective controller tuning, which are described next:

1. **Study the plant features:** This activity is referred to the analysis of the plant features such as the number of inputs and outputs, the type of behavior (linear or nonlinear), and the proper nature of the task that performs, the above to obtain a model for simulation.
2. **Select a controller structure:** Based on the model, it is possible to select a suitable controller structure that successfully stabilizes the plant behavior.

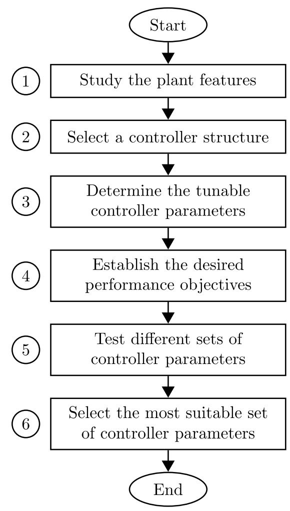

Figure 6: Typical steps for the multi-objective controller tuning.

3. **Determine the tunable controller parameters:** No matter the selected controller structure, it has a set of parameters that directly compromises the controller operation and the plant response. The designer must determine which of these parameters are used for tuning and which remain fixed to a predetermined value.
4. **Establish the desired performance objectives:** Depending on the application necessities, the designer has to establish the conflicting performance criteria that are considered for controller tuning. These criteria are related to the minimization or maximization of quantitative indicators, which are related to the system response, that formally define a multi-objective optimization problem.
5. **Test different sets of controller parameters:** Considering the difficulty in obtaining analytical solutions to engineering optimization problems, such as the controller tuning one, several alternatives (candidate sets of controller parameters) must be proposed and tested in simulation to obtain their degree of compliance with the application necessities. A way to search for suitable candidate sets is based on the use of a multi-objective meta-heuristic approach, which is described in Section 4.
6. **Select the most suitable set of controller parameters:** Using the degree of compliance of each solution, a subset of solutions with unbeatable (non-dominated) performance trade-offs is identified. Then, a single alternative is chosen from it to be implemented in the final application.

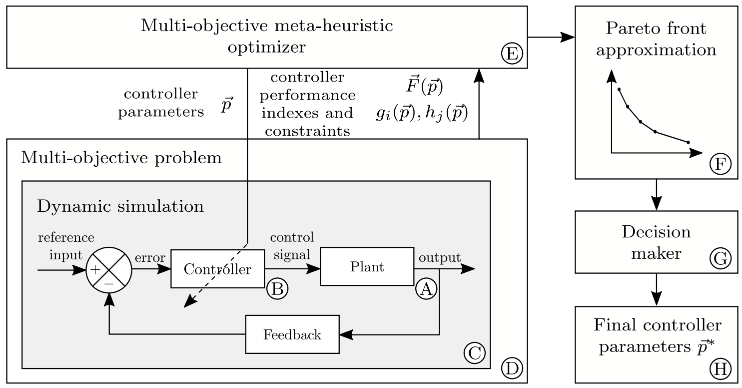

Figure 7: General Multi-objective Meta-heuristic Optimization Controller Tuning Process (MOMHOCTP).

4. Multi-objective meta-heuristic optimization in the controller tuning problem

Fig. 7 shows the general Multi-objective Meta-heuristic Optimization Controller Tuning Process (MOMHOCTP). In this, a MOP is stated based on different performance criteria, and a MOMHO is used to find the controller parameters with the best trade-offs. Each parameter configuration must be tested through a dynamic simulation to measure the controller performance and select the most promising alternatives. After the best trade-offs are found, the decision-maker is responsible for choosing a single parameter configuration to be implanted in the real controller.

At this point, it is important to highlight that the process in Fig. 7 describes the controller tuning based on an offline optimization where the best-found parameters are implanted in the controller of the final application and remain fixed. It means that there are no time limitations in the decision making, and some tests or evaluations can be performed before opting for the final configuration. On the other hand, in the controller tuning based on online optimization, which belongs to the class of adaptive controller tuning methods, the best parameters are obtained during the operation of the closed-loop system. In other words, the controller tuning process in Fig. 7 is recurrently performed to obtain different configurations along time. In this case, automated decision making becomes indispensable since it is not possible to validate all the obtained parameter configurations found by the MOMHO.

Relevant aspects of the process in Fig. 7 are discussed in detail next, based on the sustenance of relevant research works found in the specialized literature.

4.1. Plant

The plant in Fig. 7-A is a system that commonly includes, but is not limited to, the dynamics of electrical, mechanical, or pneumatic elements. Plant behavior must be adequately governed to generate useful applications.

For all works that adopt the MOMHOCTP, the plant is not handled directly, but a model is used instead. The above is to prevent damage or wear in the real plant due to tests with low-quality or preliminary controller parameter configurations.

Plants' natural behavior is always nonlinear, and nonlinear models can be obtained as state-space representations through system dynamic analysis [79] or identification [80]. Nevertheless, it is sometimes useful to assume the plant behavior to be within a certain operating region, which can be linearly modeled and represented in the form of a transfer function. An accurate plant model is highly necessary for controller tuning, but its obtaining is beyond the scope of this work.

What is interesting is the trend in usage patterns of different types of plant models and their inclusion in real-world applications. In this way, Tables 1 and 2 summarize the plant model choices in the MOMHOCTP for the reviewed works. It is observed that about of 70% of the works use linear models, and most of them correspond to benchmarked or proposed test plants. The rest of the adopted linear models are related to well-known plants such as voltage regulators, power generation systems, gas production systems, distillation systems, and aerial vehicles. On the other hand, the remaining 30% of works uses nonlinear models related to vehicles, robotic manipulators, and mechanical systems.

An important feature of a plant (and its model) which has a direct impact in the MOMHOCTP (this is discussed in depth in the MOP formulation) is its number of inputs and outputs. According to this, the plant can be Single-input Single-output (SISO) or Multi-input Multi-output (MIMO). Tables 1 and 2 also indicate that 51% of the studied plants are SISO and 49% are MIMO.

4.2. Controller

The controller in Fig. 7-B aims to govern the plant behavior to reach a reference input. It has a set of tunable parameters, denoted by $\vec{p}$, which includes variables related to a predefined control structure.

Tables 1 and 2 show trends in the selection of controller structures to govern the plants described in Section 4.1 and they are detailed next:

- PID-like controllers: This class considers the linear PI, PD, PID, and FOPID controllers. They cover 62% of the controller structure choices. This overwhelming use of PID-like controllers can be due to their simplicity and high performance that makes them to be present in most industrial applications [81].

- Fuzzy controllers: This kind of nonlinear controllers can convert a set of control rules given by expert knowledge into an automatic control strategy to handle unknown or unmodeled system dynamics [82]. 11% of the selected controllers rely on this type.

- Robust controllers: This class is referred to as those controllers that use the H_2, H_∞, H_2/H_∞, and H_2/H_2 design methods to be synthesized with high-robustness and less-sensitivity to noise or disturbances. Only 6% of the works adopt robust controllers. The above can be due to their high level of abstraction and complexity which complicate its use in practice, Nevertheless, later in Section 4.4, it is observed that any controller can be endowed with high-robustness and less-sensitivity by selecting the right objectives.

- Other controllers: The remaining 21% of works adopt more particular alternatives that by themselves do not constitute a broad enough set of studies to originate a class. These alternatives are the Linear Quadratic Regulator (LQR), the Sliding Mode Controller (SMC), the Optimal Model Predictive Controller parameterized with Laguerre functions (LOMPC), the RST controller, the inverse dynamic controller (IDC), the PD controller with Iterative Learning algorithm (PD-IL), the PID controller of the Instrument Society of America (PID-ISA), the Finite Word-Length (FWL) PID controller, nonlinear PID controller variants, and other specific purpose and state feedback controllers.

Table 1: Plants and controller structures adopted in the MOMHOCTP (Part I).

Ref.	Controller type †	Plant type ‡
[86]	Linear, PI controller	Linear, MIMO, coal gasifier
[87]	Linear, PID+FOPID controller	Linear, SISO, synchronous generator excitation system
[88]	Linear, PID+PI controller	Linear, MIMO, refrigeration system based on vapor compression
[89]	Nonlinear, Fuzzy FOPID controller / Nonlinear, Fuzzy PID controller / Linear, PID controller	Nonlinear, MIMO, 2-link robotic manipulator
[90]	Linear, Proportional Integral Derivative (PID) controller	Linear, SISO, AVR
[91]	Nonlinear, NL-PID controller / Nonlinear, NLF-PID controller / Linear, FOPID controller / Linear, PID controller	Nonlinear, MIMO, 2-link robotic manipulator
[92]	Linear, LQR-Fractional Order Proportional Integral Derivative (FOPID) controller	Linear, SISO, FO plants
[93]	Linear, PI controller	Linear, MIMO depropanizer distillation column
[94]	Linear, PID controller / Linear, FOPID controller	Linear, SISO, AVR system
[95]	Linear, PID controller / Linear, FOPID controller	Linear, SISO, AVR system
[96]	Linear, PID controller / Linear, FOPID controller	Linear, SISO, HTRS
[83]	Linear, PI controller	Linear, MIMO, coal gasifier
[45]	Linear, H_2 controller with PI structure / Linear, H_2/H_∞ controller	Linear, SISO, stable plant
[97]	Linear, PID controller	Linear, MIMO, greenhouse cultivation system
[98]	Linear, FWL-PID controller	Linear, SISO, benchmark systems
[99]	Linear, PID controller	Nonlinear, MIMO, double pendulum
[100]	Nonlinear, Fuzzy PI+D controller	Nonlinear, SISO, test process / Nonlinear, SISO, solar plant
[101]	Linear, PID-ISA	Linear, SISO, test plants
[102]	Linear, PI controller	Nonlinear, SISO, FOPDT processes
[103]	Nonlinear, Fuzzy FOPID	Linear, SISO, FO plant
[104]	Linear, PID controller	Linear, SISO, AVR
[105]	Nonlinear, Fuzzy PID controller	Linear, SISO, AMB system
[106]	Linear, FOPID controller	Linear, SISO, aircraft
[107]	Nonlinear, LOMPC	Linear, SISO, discrete-time state-space system / Linear, MIMO, discrete-time state-space system
[42]	Linear, H_∞ robust controller	Linear, MIMO, Boeing 747 aircraft
[43]	Linear, PD-IL controller	Linear, SISO wiper system
[108]	Nonlinear, Fuzzy PID controller	Nonlinear, MIMO, inverted pendulum / Nonlinear, MIMO, ball and beam system
[109]	Nonlinear, specific purpose controller	Nonlinear, SISO, vehicle
[110]	Linear, PID controller	Linear, SISO benchmark plants / Linear, MIMO, test plant
[111]	Linear, H_2/H_∞ controller / Linear, H_2/H_2 controller	Linear, SISO, test plants / Linear, MIMO, test plant
[112]	Linear, IOPID controller / Linear, FOPID controller	Linear, SISO, FO plant
[113]	Linear, PID controller	Linear, SISO, power systems
[114]	Nonlinear, Fuzzy PID controller	Linear, SISO, DC motor
[115]	Linear, PID controller / Linear, I-PD controller	Linear, SISO, patient model
[116]	Linear, PID controller	Linear, SISO, HTRS
[117]	Linear, PD controller	Nonlinear, MIMO, inverted pendulum / Nonlinear, MIMO, ball and beam system
[46]	Nonlinear, SMC	Nonlinear, MIMO, biped robot
[118]	Nonlinear, state feedback controller	Nonlinear, MIMO, biped robot
[84]	Linear, PI controller	Linear, MIMO, coal gasifier
[44]	Linear, RST controller	Linear, SISO, electrical DC drive
[119]	Linear, PI controller / Linear, PID controller	Linear, SISO, open-loop unstable system / Linear, SISO, triple pole system
[120]	Linear, PID controller	Linear, SISO, CSTR system
[41]	Linear, PD controller	Nonlinear, MIMO, double pendulum
[121]	Linear, PI controller	Nonlinear, MIMO, CAC-ZVS three phase PFC converter
[122]	Linear, PI controller	Linear, MIMO, super-heater
[85]	Linear, PID controller	Linear, MIMO, distillation column / Linear, MIMO, F18/HARV fighter aircraft
[123]	Nonlinear, Fuzzy inverse dynamics controller	Nonlinear, MIMO, parallelogram mechanism
[124]	Linear, PID controller	Linear, SISO, flexible link system
[125]	Linear, PI controller	Linear, SISO, test plants
[75]	Linear, PIDn controller	Linear, SISO, boiler system / Linear, MIMO, boiler system
[126]	Linear, PI controller	Linear, MIMO, UAV
[47]	Linear, PI controller	Linear, MIMO, Wood and Berry distillation column
[127]	Linear, PI controller	Linear, MIMO, Wood and Berry distillation column
[128]	Linear, PID controller	Linear, SISO, Cholette reactor
[129]	Nonlinear, Fuzzy controller	Nonlinear, SISO, cantilever beam system
[130]⋄	Nonlinear, inverse dynamics controller	Linear, SISO, DC motor
[131]	Linear, PI controller	Linear, MIMO, super manoeuvrable fighter aircraft
[132]	Linear, PI controller	Linear, SISO, first-order-plus-dead-time process
[133]	Linear, FOPID controller	Linear, SISO, AVR system
[134]	Linear, PID controller	Nonlinear, MIMO, goethite Process

† Controller type: Behavior, Structure.
‡ Plant type: Behavior, Inputs and outputs, Class.
⋄ Online optimization approach.

Table 2: Plants and controller structures adopted in the MOMHOCTP (Part II).

Ref.	Controller type †	Plant type ‡
[135]	Linear, PID controller / Linear, Lead Compensator / Linear PI+P controller	Nonlinear, SISO, ball and beam system / Linear, MIMO, stirring tank with heat exchanger / Linear, MIMO, distillation column
[136]	Linear, 2-DOF FOPD controller / Linear, 2-DOF PD controller / Linear, PID controller	Nonlinear, MIMO, 2-link robotic manipulator
[137]	Linear, FOPID controller	Linear, SISO, linear pump turbine governing system
[138]	Nonlinear, active disturbance rejection controller	Nonlinear, MIMO, inverted pendulum
[139]	Linear, PID	Nonlinear, MIMO, aerial manipulator
[140]	Linear, PID	Linear, SISO, DC servo motor
[141]	Nonlinear, Fuzzy controller	Nonlinear, MIMO, biped robot
[142]	Nonlinear, adaptive robust decoupled sliding mode control	Nonlinear, MIMO, ball and beam system
[143]	Linear, LQR controller	Linear, MIMO, aircraft
[144]⋄	Nonlinear, inverse dynamics controller	Nonlinear, SISO, four-bar mechanism
[145]	Nonlinear, second order sliding mode control	Linear, MIMO, grid-side converter of a doubly-fed induction generator
[146]	Linear, PID controller with state observer	Linear, MIMO, second-order oscillator
[147]	Linear, four-parameter PID controller	Linear, SISO, DC servo motor

† Controller type: Behavior, Structure.
‡ Plant type: Behavior, Inputs and outputs, Class.
⋄ Online optimization approach.

4.3. Dynamic simulation

In general terms, a closed-loop control system, such as the one in Fig. 7-C, includes a controller that governs the plant behavior to a reference input. The closed-loop control system has one or more feedback loops between its outputs and inputs. The feedback acquires information about the plant operation (e.g., the output values through sensing devices), and based on this, the controller can take proper control actions. With the above mentioned, closed-loop systems can reduce errors and achieve stability, among other advantages concerning the open-loop ones, which do not include feedback loops [148].

The dynamic simulation of the closed-loop control system refers to the modeling of its time-varying behavior. This simulation allows the evaluation of the overall performance of a given controller configuration $\vec{p}$ (e.g., the consumed energy and the set-point tracking precision during a time window).

In order to perform a dynamic simulation, the plant, the controller structure, the feedback, and a set of operational/environmental conditions (this can be related to initial conditions, parametric uncertainties, disturbances, workload changes, among others) must be well defined.

The feedback operation is negligible for all the reviewed works, i.e., the dynamic behavior of sensors or transducers is not modeled either considered in the whole closed-loop control system behavior. Similarly, this happens with the actuators and power devices, among other elements. Therefore, the closed-loop system in Fig. 7-C is described only in terms of the plant and the controller behaviors. A set of differential equations typically forms the above description.

For the surveyed works, the simulation of a closed-loop control system is performed in two ways depending on the characteristics of the plant and the controller. If both of them obey the superposition principle [148], then the system is linear, and some tools, such as the Laplace transform, are used to solve the involved differential equations to describe and simulate the system behavior using frequency-domain algebraic equations [116]. In contrast, the control system is nonlinear, and the dynamic simulation is carried out by using numerical methods due to the more complex structures of their differential equations. Numerical integration methods such as Runge-Kutta [99] and Euler [130] are adopted for this purpose. In some cases, a linearization method can be used in these kinds of systems when they operate near to a stable point, as in [94].

4.4. Multi-objective optimization problem

Using the information obtained from the dynamic simulation and from the closed-loop control system parameters, it is possible to measure the quality of a controller configuration $\vec{p}$ according to several performance indexes denoted by $\vec{F}(\vec{p}) = [f_1(\vec{p}), \ldots, f_m(\vec{p})]^T$, and determine its limitations given by $g_i(\vec{p})$ and $h_j(\vec{p})$, as observed in Fig. 7-D.

The above elements are formally stated in a MOP in the form of (1). Without loss of generality, the controller tuning MOP consists of finding a set of reliable controller parameter configurations, with different performance trade-offs.

The MOP formulation is an important aspect addressed in this review. It is focused on the selection of the design variables (tunable controller parameters), objective functions (performance requirements of the closed-loop system), and constraints (closed-loop control system limitations). Findings of those choices are described next.uch as Runge-Kutta [99] and Euler [130] are adopted for this purpose. In some cases, a linearization method can be used in these kinds of systems when they operate near to a stable point, as in [94].

4.4.1. The design variables

In the reviewed works, the design variables $\vec{p}$ are intimately related to the chosen controller structures mentioned in Section 4.2. Taking into account the information in Tables 3-6, typical choices for each controller structure are the following:

- PID-like: For PI, PD, and PID controllers, design variables include the gains k_p, k_i and k_d (in some cases the gains k_i and k_d are parametrized regarding the integral and derivative time constants τ_i and τ_d, respectively). In the case of FOPID controllers, the integro-differential orders λ and μ are also considered. Additional PID-like structures can include filter constants.

- Fuzzy controllers: for this type of controllers, design variables are used to represent the complexity of the controller (regarding the number of inputs, membership functions, fuzzy rules, etc.), the shape of the membership functions, the fuzzy rules, and the output form.

- Robust controllers: Regardless of the adopted robust design method, every robust controller has a set of tunable parameters that usually corresponds to the pole position, combined controller structure parameters (e.g., gains of a PID controller), and more particular parametrization variables (in matrix forms).

- Other controllers: Despite there is no generalization in the design variables used in this kind of controllers (beyond the fact that they represent parametrization variables), two approaches stand out from others because they do not explicitly relate the design variables with the structural parameters of the controller. In [109], the design variables correspond to the control signals applied for each sampling instant during a given time window. On the other hand, in [130] the design variables represent the dynamic parameters of the plant used for control.

As discussed in Section 4.2, a homogeneous scaling of independent controller structures is typically used to handle a MIMO plant. The above implies that the number of design parameters used in a controller for a SISO plant is scaled by the number of inputs/outputs of the MIMO one.

According to Tables 3-6, the number of the design variables involved in the reviewed works ranges between 2 and 47 indiscriminately for SISO or MIMO plants. Then, the MOPs of these works are considered moderate-size problems [149]. In the exceptional case of [109], the number of design variables is $46,000$, and the MOP belongs to the class of large-scale optimization problems [150].

Regarding the type of design variables, works that include PID-like, robust, and other controller structures, use continuous variables. For works with fuzzy controller structures, design variables are mixed continuous-discrete.

4.4.2. The objective functions

Low cost, fast response, robustness, low sensitivity, small error rates, high accuracy, and efficient energy usage are examples of common requirements for closed-loop control systems, which may conflict with each other. These features strongly depend on the settling of the controller parameters $\vec{p}$.

Each one of the closed-loop control system requirements are translated to the mathematical language to define the objective functions $\vec{F}(\vec{p}) = [f_1(\vec{p}), \ldots, f_m(\vec{p})]^T$ in (1).

Depending on the closed-loop control system representation, different performance indexes are chosen in the reviewed works as observed in Tables 3-6. For a time-domain representation, the control performance indexes, response characteristics or indexes related to the control signal behavior are used and listed below:

- Integral Absolute Error (IAE) [114]: Determines how far is the system response from the reference signal.
- Integral Time-weighted Absolute Error (ITAE) [132]: Similar to IAE, but in this case initial errors have less importance.
- Integral Squared Error (ISE) [113]: This value takes into account large errors more than small errors between the system response and the reference signal.
- Integral Time-weighted Squared Error (ITSE) [104]: Like ITAE, this index gives more importance to the new larger errors than initial smaller errors.
- Settling time (ST) [126]: It is the time required to minimize the error between the system response and the steady state response.
- Rise time (RT) [32]: It is the time required to take the system response to a steady state response.

- Overshoot (OS) [33]: It is a relation between the maximum value of the system response and the steady state response.

- Total variation of the control signal (TV) [102]: Measures the smoothness of the control signal.

- Integral of the Squared Deviation of Controller Output (ISDCO) [75]: This is another index of the control signal smoothness.

Other time-domain control performance indexes used as objective functions are related to particular error measurements [129], specific behaviors of outputs [115], or probabilistic output criteria [112].

For frequency-domain representations, the used performance measurements are the following:

- Steady State Error (SSE) [116]: It is the error obtained in the limit as time goes to infinity.

- Gain Margin (GM) and Phase Margin (PM) [94]: Both indicate the relative stability of the control system (i.e., how close is the system to be unstable).

- Maximum Sensitivity (MS) [131]: A performance measurement regarding external disturbance rejection.

- Maximum Complementary Sensitivity (MCS) [131]: A performance measurement regarding robust stability.

- Log-Modulus (LM) [151]: A large value of this measurement makes the steady state error small.

Works in [98, 107, 114] care about the closed-loop control system cost regarding the complexity of the computational structures involved in their controllers. Therefore, the computational effort is established as an objective function.

On the other hand, for problems with MIMO plants, it is common to find several objective functions considering the sums of given performance metrics (e.g., ISE, RT, and ST in [97]) over all the involved inputs/outputs. A less common approach establishes a one-to-one relationship between each objective function and a performance metric over a single input/output as observed in works like [83]. Nevertheless, this approach considerably increases the number of objectives and could also increase the complexity of the optimization problem, but the range of trade-offs can also grow.

It is also interesting to observe the number of objectives tackled by the reviewed works in Tables 3-6. Around 51% of works are related to bi-objective controller tuning problems, 29% handle MOPs with three objective functions, and only 20% include more than three objectives. It is important to remark that optimization problems with more than three objectives are considered in the particular class of many-objective problems [65]. Implications regarding the search in these kinds of problems are discussed in Section 4.5.

4.4.3. The constraints

Closed-loop control systems, and in general all systems, have practical limitations that can be technological (e.g., the computational power required to run an intelligent control strategy), economical (e.g., the budget to develop a given control application), operational (e.g., the overall system stability or the maximum allowable error rates during a set-point tracking), or environmental (e.g., the temperature in different places of a factory). Such limitations are considered in the problem (1) for controller tuning as inequality constraints $g_i(\vec{p})$, equality constraints $h_j(\vec{p})$, and boundary constraints p_k^{min} and p_k^{max}.

The boundary constraints p_k^{min} and p_k^{max} are fundamental for the controller tuning MOP since they provide the first search space delimitation. They are included in all works to limit the values of the controller structure parameters $\vec{p}$. Although the establishment of boundary constraints is not discussed deeply in the surveyed works, it is assumed that they are obtained based on the designer experience.

Regarding explicit (non-boundary) constraints (equality and inequality constraints), only 34% of the controller tuning problems include them, and Tables 3-6 summarize typical choices.

About equality constraints $h_j(\vec{p})$, the most common choice refers to the closed-loop stability (i.e., for every bounded input, the system must produce a bounded output). In the case of inequality constraints $g_i(\vec{p})$, they are commonly used to establish allowable values of the objective functions and also of the control signal taking into account the actual actuator capacity. Even in some exceptional cases, as in [97] and [115], inequality constraints are set to bound the values of other control performance indexes different from those adopted as objective functions.

As a reminder, the solution of a MOP is a Pareto optimal set $\mathcal{P}^*$, that mapped in the objective function space, generates a true Pareto front $\mathcal{PF}^*$ that contains all possible trade-offs among objectives. In the MOMHOCTP, $\mathcal{P}^*$ refers to all feasible controller parameter configurations, while $\mathcal{PF}^*$ includes all their performance trade-offs.

Table 3: Formulation of MOPs for controller tuning (Part I).

Ref.	Space ⋆	Objectives	Design variables	Constraints
[86]	(4,15,0)	IAE (for four outputs)	PID: gains k_p^j, k_i^j and k_d^j, $j = 1, \ldots, 5$	-
[87]	(3,7,6)	ISE, MS, control effort	PID+FOPID: gains k_p^j, k_i^j and k_d^j, $j = 1, 2$, integro-differential orders λ and μ	Routh stability conditions
[88]	(5,6,0)	IAE (for two outputs), TV (for two inputs), LM	PID+PI: gains k_p^j, k_i^j, $j = 1, 2$ and k_d, filter f	-
[89]	(2,16,0)	Sums of IAE and control efforts (for two outputs)	Fuzzy FOPID: gains k_p^j and k_d^j, parameters P_i^j and b_i^j, integro-differential orders λ^j and μ^j, fuzzy parameters L^j and ku^j, $j = 1, 2$	-
	(2,12,0)	Sums of IAE and control efforts (for two outputs)	Fuzzy PID: gains k_p^j and k_d^j, parameters P_i^j and b_i^j, fuzzy parameters L^j and ku^j, $j = 1, 2$	-
	(2,8,0)	Sums of IAE and control efforts (for two outputs)	PID: gains k_p^j and k_d^j, parameters P_i^j and b_i^j, $j = 1, 2$	-
[90]	(3,3,0)	ITSE, OS, ST	PID: gains k_p, k_i and k_d	-
[91]	(2,8,0)	Sums of IAE and control efforts (for two outputs)	NL-PID: gains k_p^j, k_i^j and k_d^j, nonlinear gain parameter α^j, $j = 1, 2$	-
	(2,12,0)	Sums of IAE and control efforts (for two outputs)	NLF-PID: gains k_p^j, k_i^j and k_d^j, integro-differential orders λ^j and μ^j, nonlinear gain parameter α^j, $j = 1, 2$	-
	(2,10,0)	Sums of IAE and control efforts (for two outputs)	FOPID: gains k_p^j, k_i^j and k_d^j, integro-differential orders λ^j and μ^j, $j = 1, 2$	-
	(2,6,0)	Sums of IAE and control efforts (for two outputs)	PID: gains k_p^j, k_i^j and k_d^j, $j = 1, \ldots, 2$	-
[92]	(2,5,0)	ITSE, ISDCO	LQR: weighting matrix Q, weighting factor R, FOPID: integro-differential orders λ and μ	-
[93]	(3,10,0)	Sums of IAE, OS and RT (for responses of column elements)	PID: gains k_c^j, integral times τ_i^j, $j = 1, \ldots, 5$	-
[94]	(2,3,2)	GM, PM	PID: gains k_p, k_i and k_d	Positive GM, PM
	(2,5,2)	GM, PM	FOPID: gains k_p, k_i and k_d, integro-differential orders λ and μ	Positive GM, PM
[95]	(2,3,0)	ITSE, ISDCO	PID: gains k_p, k_i and k_d	-
	(2,5,0)	ITSE, ISDCO	FOPID: gains k_p, k_i and k_d, integro-differential orders λ and μ	-
	(2,3,0)	ITSE (for set-point and disturbance rejection)	PID: gains k_p, k_i and k_d	-
	(2,5,0)	ITSE (for set-point and disturbance rejection)	FOPID: gains k_p, k_i and k_d, integro-differential orders λ and μ	-
	(3,3,0)	ITSE (for set-point and disturbance rejection), ISDCO	PID: gains k_p, k_i and k_d	-
	(3,5,0)	ITSE (for set-point and disturbance rejection), ISDCO	FOPID: gains k_p, k_i and k_d, integro-differential orders λ and μ	-
[96]	(2,3,0)	ISE and ITSE	PID: gains k_p, k_i and k_d	-
	(2,5,0)	ISE and ITSE	FOPID: gains k_p, k_i and k_d, integro-differential orders λ and μ	-
[83]	(6,8,24)	Sum of IAE (for four outputs relative to six typical operation scenarios)	PI: gains k_p^j, $j = 1, \ldots, 4$, k_i^l, $l = 1, 2, 3$ and k_f	Input and output limits for operation scenarios
[45]	(2,2,1)	MS, MCS	H_2-PI: parameters k_p and τ_i	Closed-loop stability
	(2,N,1)	MS, MCS	H_2/H_∞: place of the poles	Closed-loop stability
[97]	(3,6,9)	Sums of ISE, RT, ST	PID: gains k_p^j, k_i^j and k_d^j, $j = 1, 2$	Permissible values of OS and undershoot, real values of IAE, ITSE and ITAE, SSE bounds, minimum values of PM and GM
[98]	(2,9,0)	MS, MCS, number of bits for the FWL implementation	FWL: structural parameters	-
[99]	(2,6,0)	Sums of position errors and variation of control signals (for two outputs)	PID: gains k_c^j, integral times τ_i^j, $j = 1, \ldots, 2$	-
[100]	(3,7,0)	OS, ST, RT	Fuzzy PI+D: gains k_p, k_i, k_d, k_{uPI}, k_{uPD}, k, and membership function constant L	-

⋆ MOP definitions in terms of (objectives, variables, non-boundary constraints).

Table 4: Formulation of MOPs for controller tuning (Part II).

Ref.	Space ⋆	Objectives	Design variables	Constraints
[101]	(3,7,2)	IAE, set-point error, MS	PID-ISA: parameters T_i, T_f, T_d, N, k, b and c.	Closed-loop stability, size of the control signal
[102]	(4,3,0)	IAE (for set-point and disturbance rejection), MS, TV	PI: parameters k_c, T_i and b	-
[103]	(3,27,0)	Mean squared error, MCS, maximum control effort	Fuzzy FOPID: 15 membership functions parameters, 5 fuzzy rule parameters, and nominal FOPID parameters a, b, k_p^0, k_i^0, k_d^0, γ^0 and μ^0	-
[104]	(4,3,0)	IAE, ITAE, ISE, ITSE	PID: gains k_p, k_i and k_d	-
[105]	(2,4,0)	IAE, OS	Fuzzy PID: scaling factors GE, GCE, GU, and GCU	-
[106]	(2,5,0)	MS, MCS	FOPID: gains k_p, k_i and k_d, integro-differential orders λ and μ	-
[107]	(3,2,0)	Size of the feasibility region, performance loss, computational cost	LOMPC: number of degrees of freedom n_c and the pole of the discrete-time Laguerre network a	-
[42]	(3,4,3)	OS, RT, ST	H_∞: weights a, b, c and d	Pertinence indicators over OS, RT, and ST
[43]	(3,5,0)	IAE, RT, OS	PD-IL: gains k_p and k_d, and parameters of the iterative learning algorithm ϕ, ψ and Γ	-
[108]	(3,12,0)	IAE (for each output), control effort	Fuzzy PID: parameters of the Single Input Fuzzy Inference Motor (SIFIM) and the Preferred Fuzzy Inference Motor (PFIM)	-
[109]	(2,46000,0)	Absorbed power, mean dissipated power	46,000 control inputs applied for each sampling instant	-
[110]	(3,4,1)	Disturbance rejection, set-point error, sensitivity	Parameters of a single-loop PI	Actuator capacity
	(3,7,1)	Disturbance rejection, set-point error, sensitivity	Parameters of a single-loop PID	Actuator capacity
	(3,14,1)	Disturbance rejection, set-point error, sensitivity	Parameters of a multi-loop PID controller with decoupled structure	Actuator capacity
	(3,26,1)	Disturbance rejection, set-point error, sensitivity	Parameters of a multi-loop PID controller with coupled structure	Actuator capacity
[111]	(3,n,0)	Distance of stability, H_2 norm, H_∞ norm	H_2/H_∞: n parametrization variables	-
	(3,n,0)	Distance of stability, H_2 norm (for two outputs)	H_2/H_2: n parametrization variables	-
[112]	(5,3,0)	Probabilistic criteria (of instability, disturbance rejection, frequency response, failure of ITSE), control effort	PID: gains k_p, k_i and k_d	-
	(5,5,0)	Probabilistic criteria (of instability, disturbance rejection, frequency response, failure of ITSE), control effort	FOPID: gains k_p, k_i and k_d, integro-differential orders λ and μ	-
[113]	(2,3,0)	ISE index (for error and control signals)	PID: gains k_p, k_i and k_d	-
[114]	(4,n,0)	RT, SSE, power consumption, controller complexity	Fuzzy PID: number of inputs, number of membership functions, number of rules, and-or-ignore conjugates, type of defuzzification algorithm	-
[115]	(2,3,4)	Number of proliferating cancer cells, average level of toxicity	PID/I-PD: gains k_p, k_i and k_d	Closed-loop stability, maximum toxicity, bounds of the drug concentration
[116]	(2,3,0)	ISE, ITSE	PID: gains k_p, k_i and k_d	-
[117]	(2,4,0)	Sums of ST and OS	PD: gains k_p^j and $k_d^j, j = 1, 2$	-
[46]	(2,6,0)	Sums of errors and control efforts	SMC: three positive constants, three sliding surfaces	-
[118]	(2,6,0)	Sums of errors and control effort	Controller state feedback parameters	-
[84]	(8,8,0)	IAE and Maximum Absolute Error (MAE) (for four outputs)	PI: gains $k_p^j, j = 1, \ldots, 4$, $k_i^l, l = 1, 2, 3$ and k_f	-
[44]	(2,3,2)	IAE, OS	RTS: parameters s_1, r_0, r_1	Sensitivity bounds
[119]	(4,4,1)	ITAE (for set-point and disturbance rejection), control effort, vector margin	PI: gains k_p, k_i, and lead-lag pre-filter parameters T_{lead}, T_{lag}	Closed-loop stability
	(4,5,1)	ITAE (for set-point and disturbance rejection), control effort, vector margin	PID: parameters k_p, k_i, k_d, and lead-lag pre-filter parameters T_{lead}, T_{lag}	Closed-loop stability

⋆ MOP definitions in terms of (objectives, variables, non-boundary constraints).

Table 5: Formulation of MOPs for controller tuning (Part III).

Ref.	Space ⋆	Objectives	Design variables	Constraints
[120]	(3,5,1)	ISE, IAE, MS	PID: parameters α, k_c, T_i, T_d, T_f	Maximum control action
[41]	(2,4,0)	Sums of errors and variations of control signal (for two outputs)	PD: diagonal elements of the K_p and K_d gain matrices	-
[121]	(3,4,0)	Sums of OS, SEE, current loop function	PI: gains k_p^j and $k_i^j, j = 1, 2$	-
[122]	(2,2,1)	IAE (for set-point and disturbance rejection)	Inner PI: gains k_p^j and $k_i^j, j = 1, 2$	Upper limit of MS
	(2,2,0)	Robustness, disturbance rejection	Outer PI: gains k_p^j and $k_i^j, j = 1, 2$	-
[85]	(2,12,2)	ISE, aggregated function of robust stability and disturbance rejection indexes	PID: parametrization matrices B_0, B_1, B_2	Upper bounds of robust stability and disturbance rejection indexes
	(2,27,2)	ISE, aggregated function of robust stability and disturbance rejection indexes	PID: parametrization matrices B_0, B_1, B_2	Upper bounds of robust stability and disturbance rejection indexes
[123]	(2,10,0)	Sums of IAE and control efforts (for two outputs)	Fuzzy inverse dynamics: fuzzy rules and positive coefficients of the controller	-
[124]	(5,3,1)	OS, SSE, ST, position/tip response, peak response of tip vibration	PID: gains k_p, k_i and k_d	Closed-loop stability
[125]	(3,3,5)	Integral gain, MS, MCS	PI: gain k_p, time τ_i and weight b	Maximum control action, bounds of MS and MCS
[75]	(3,4,1)	IAE, MS, maximum value of the noise sensitivity M_u	PIDn: gain k_p, times τ_i, τ_d, and filter N	Closed-loop stability
	(5,4,1)	Stabilizing time (for two outputs), LM, MS (for two control loops)	PIDn: gain k_p, times τ_i, τ_d, and filter N	Closed-loop stability
[126]	(7,10,0)	ST (for yaw and altitude), variation of control action (for the throttle, aileron, elevator, roll, and pitch)	PI: gain k_p and time τ_i	-
	(9,10,0)	Median, maximum and median absolute deviation of time to perform a flight mission, number of successful fight missions, median variation of control action (for the throttle, aileron, elevator, roll, and pitch)	PI: gain k_p and time τ_i	-
[47]	(7,4,11)	Integral gains, MS and MCS (for two controllers), LM	PI: gains k_p^j and times $\tau_i^j, j = 1, 2$	Bounds of MS and MCS (for each controller), upper bound of LM, closed-loop stability
[127]	(7,4,11)	Integral gains, MS and MCS (for two controllers), LM	PI: gains k_p^j and times $\tau_i^j, j = 1, 2$	Bounds of MS and MCS (for each controller), upper bound of LM, closed-loop stability
[128]	(3,3,3)	ST, GM, PM	PID: gain k_p, times τ_i and τ_d	Upper bounds of ST, GM, and PM
[129]	(3,47,0)	Norm error (for displacements/rotations, velocities, and accelerations)	Fuzzy controller: membership functions, weights of fuzzy rules and the logical operators	-
[130] ⋄	(2,6,2)	ISE, ISE sensitivity concerning the model parameters (objective values are obtained within a past time window)	Inverse dynamics controller: dynamic parameters of DC motor	Bounds of control signal
[131]	(3,18,2)	MS, MCS, ISE	PI: gain matrices K_p and K_i	Upper bounds of MS and MCS
[132]	(3,2,0)	ITAE and ISDU (performance measure for the manipulated variable movements)	PI: gain k_p and time τ_i	-
[133]	(3,5,0)	IAE, SSE, ST	FOPID: gains k_p, k_i and k_d, integro-differential orders λ and μ	-
[134]	(2,24,16)	Sums of IAE and control efforts (for four outputs)	PID: gains k_p^j, k_i^j and $k_d^j, j = 1, \dots, 8$	Bounds of control signals

⋆ MOP definitions in terms of (objectives, variables, non-boundary constraints).

⋄ Online optimization approach.

4.5. Multi-objective meta-heuristic optimizer

The main goal of the tuning process is to find proper controller configurations that met all performance specifications established in the MOP. Then, the MOMHO in Fig. 7-D is responsible for solving the controller tuning MOP (Fig. 7-C) to find the best controller parameters.

As described in Section 2, MOMHOs cannot guarantee to find the optimal Pareto set $\mathcal{P}^*$ as the solution of the controller tuning MOP, which is inherently complex. Still, in exchange for a reasonable computational cost, they can find an accurate optimal Pareto set approximation $\mathcal{P}^A$ and an approximated Pareto front $\mathcal{PF}^A$ with different performance trade-offs.

4.5.1. Meta-heuristic approaches

According to Tables 7-10, MOMHOs used to solve the controller tuning MOP can be classified as follows:

- Multi-objective Genetic Algorithms: 51% of works use variants of Multi-objective Genetic Algorithms.

  In genetic algorithms [52], the fitness of individuals in the population (usually randomly initialized at the beginning) is evaluated. Next, the best individuals are selected according to their fitness. Then, the best individuals have better chances to generate the next population through crossover and mutation operations. The process is repeated until a stop condition is satisfied.

  Among the approaches found in Table 7, the Non-dominated Sorting Genetic Algorithm II (NSGA-II) [152] looks like a popular choice. This state-of-the-art MOMHO has exhibited a high performance [153]. NSGA-II includes a non-dominated sorting operator to determine the rank of each individual based on Pareto dominance, crowding of individuals and feasibility. Although a considerable number of works use the original NSGA-II, some improvements have been proposed to solve the controller tuning MOP. They include the use of chaotic numbers to enhance the diversity of solutions, the replacement of the original crossover/mutation operators, the use of a local search on individuals for fine-tuning, and the generation of the initial population based on tuning methods from control engineering.

  It is also observed in Table 7 the use of Multi-objective Genetic Algorithms from the MATLAB optimization toolkit (MATLAB MOGA), that although it has presented good results, limits the possibility of making particular improvements for the controller tuning MOP.

  The rest of alternatives are Multi-objective Genetic Algorithm (MOGA) variants that include some of the ideas introduced in [59], such as the use of secondary populations, fitness sharing, niching, mating restrictions and migration schemes, as well as the use of different crossover/mutation and selection operators, and individuals generated by controller tuning

Table 6: Formulation of MOPs for controller tuning (Part IV).

Ref.	Space ⋆	Objectives	Design variables	Constraints
[135]	(2,3,0)	IAE and sum of response properties (rise, setting, and peak times)	PID: gains k_p, k_i and k_d	-
	(2,3,0)	IAE and sum of response properties (rise, setting, and peak times)	Lead compensator: gains k_1, k_2 and k_3	-
	(2,6,0)	IAE (for two outputs) and sum of response properties (rise, setting, and peak times for two outputs)	PID: gains k_p^j, k_i^j and $k_d^j, j = 1, 2$	-
	(2,5,0)	IAE (for two outputs) and sum of response properties (rise, setting, and peak times for two outputs)	PI+P: gains k_p^j and k_i^j, $j = 1, 2$, and k_p	-
[136]	(2,10,0)	Sums of ISE and control efforts (for two outputs)	Parameters of two 2-DOF FOPD controllers	-
	(2,8,0)	Sums of ISE and control efforts (for two outputs)	Parameters of two 2-DOF PD controllers	-
	(2,6,0)	Sums of ISE and control efforts (for two outputs)	PID: gains k_p^j, k_i^j and $k_d^j, j = 1, 2$	-
[137]	(2,5,0)	ITAE (for two working conditions)	FOPID: gains k_p, k_i and k_d, integro-differential orders λ and μ	-
[138]	(8,11,0)	Tracking error and control effort (for two outputs), RT, OS, ST, SSE	Parameters of the active disturbance rejection control loops	-
[139]	(2,27,0)	Sums of ISE and control effort (for nine outputs)	PID: gains k_p^j, k_i^j and $k_d^j, j = 1, \ldots, 9$	-
[140]	(3,3,0)	OS, ST, SSE	PID: gains k_p, k_i and k_d	-
[141]	(2,6,0)	Sums of errors and control efforts	Fuzzy controller: weight constants	-
[142]	(2,8,0)	ITAE (for two outputs)	Parameters of the adaptive robust decoupled sliding mode control	-
[143]	(2,9,1)	ST, RT	LQR: weighting matrix Q	Closed-loop stability
[144]⋄	(2,12,0)	ISE, smoothness of control signal	Inverse dynamics controller: dynamic parameters of the four-bar mechanism	-
[145]	(18,6,18)	IAE per unit of time and output deviation (for nine operation stages)	Parameters of the second order sliding mode control	Upper bounds of the objectives
	(18,4,18)	IAE per unit of time and output deviation (for nine operation stages)	Parameters of the second order sliding mode control	Upper bounds of the objectives
[146]	(6,6,3)	Peak time, OS, H_∞ norm, control energy, IAE, Frobenius norm	Gains of the PID controller and the observer	Maximum OS and two eigenvalues conditions (for stability)
[147]	(4,4,2)	Peak time, OS, H_∞ norm (for load disturbance and noise rejection)	Gains of the four-parameter PID controller	Upper bound of the H_∞ norm (for load disturbance and noise rejection)

⋆ MOP definitions in terms of (objectives, variables, non-boundary constraints).
⋄ Online optimization approach.

methods. All the above mechanisms were proposed to improve the exploration and exploitation of the search space, to prevent premature and local convergence, and also enhance the solution diversity.

Another remarkable feature found in some of these MOMHOs, is the inclusion of elitism through the use of external archives. The above is used to retain the non-dominated solutions found on each iteration.

- Multi-objective Particle Swarm Optimization: The Multi-objective Particle Swarm Optimization (MOPSO) is used in the 20% of the surveyed works.

  The Particle Swarm Optimization [54] uses a swarm of particles (randomly initialized) that update their position for each iteration until a stop criterion is satisfied. Each particle position is evaluated to determine its fitness. The particle velocity (i.e., search direction) is used to update the particle position, and it is obtained based on the swarm fitness information. The information can include the best position known by the swarm (global best), the overall best position known by the particle (personal best), and the best position of a sub-swarm described by a topology (local best). The particle position can be perturbed by a turbulence operator (equivalent to the mutation in GAs) in some cases.

  All the MOMHO choices in Table 8 include several versions of the original PSO algorithm applied to the controller tuning MOP. Those versions vary in the use of different swarm topologies, leader selection mechanisms (to obtain the global, personal, and local best solutions), secondary swarms, and turbulence operators in search of the best Pareto front approximation. Most of the improvements are described in [154].

  External archives are included in most of these MOMHOs, and interactions with archived solutions and the swarm are observed, in the sense that some leader solutions are selected from the archive. Moreover, strategies to maintain the size and diversity of solutions of the external archive such as pruning methods and grid systems (i.e., discriminating solutions that share the same region of the objective function space to be removed) are frequent.

- Multi-objective Differential Evolution: About 19% of works include a variant of Multi-objective Differential Evolution (MODE).

  Differential Evolution [53] like a GA, is a MOMHO based on the process of natural evolution. This algorithm includes a population of individuals (randomly initialized), which are evaluated to determine their fitness. A mutant is generated for each original individual in the population based on the perturbation of a given base individual by the scaled differences of other individual pairs. This mutant is recombined with the original individual to generate an offspring. At the end of each generation, the fittest solution between each original individual and its offspring forms the next population. The above process is repeated until a given stop condition is met.

The MODE alternatives shown in Table 9 are based in different variants of the original DE. The variants aim to improve the explorative and exploitative capabilities of the algorithm [155]. Among them, the DE/rand/1/bin variant is the most popular for controller tuning.

The use of an external archive to store non-dominated solutions is observed in most of these MOMHO variants. A noticeable feature is the inclusion of more sophisticated pruning mechanisms to improve the diversity of elite solutions in the archive. They include spherical grids and preference information about objectives in physical programming, a reference point, or indicator-based methods to assign a fitness value to discriminate solutions based on their isolation or pertinence level to desirable regions. Interactions between the external archive and the population are also observed.

- Other multi-objective optimizers: Only 10% of surveyed works use less explored MOMHOs. They include several state-of-the-art algorithms such as the Strength Pareto Evolutionary Algorithm (SPEA) [156], the Strength Pareto Evolutionary Algorithm II (SPEA-II) [157], the Dual Population Evolutionary Algorithm (DPEA) [158], the Gravitational Search Algorithm (GSA) [159], the Multi-objective Extremal Optimization (MOEO) [160], the Approximation-Guided Evolutionary algorithm II (AGE-II) [161], and the Reference Vector Guided Evolutionary Algorithm (RVEA). The remaining works use novel multi-objective variants of meta-heuristics originally adopted for global optimization such as the Simulated Annealing algorithm, named Intelligent Multi-objective Simulated Annealing (IMOSA), the Multi-objective Cuckoo Search Algorithm (MOCSA) [136], the Multi-objective Whale Optimization Algorithm (MOWOA) [140], the Multi-objective Grey Wolf Optimizer (MOGWO)[140], and the Multi-objective Artificial Bee Colony (Multi-objective ABC) [142].

About the search mechanisms included in the MOMHOs shown in Tables 7-10, it is easy to observe that the 92% belongs to the dominance-based approach. MODE variants from the metric-driven approach form the remaining 8%.

Although 20% of the controller tuning problems are many-objective problems, the used dominance-based MOMHOs appear to have a good performance when finding suitable controller configurations. Strong evidence about the performance of decomposition-based, metric-driven, or hybrid approaches cannot be found in the specialized literature.

Large-scale problems are quite rare in the context of controller tuning. Nevertheless, for the large-scale problem in [109], the initialization of solutions based on the information of tuning methods from control engineering seems effective.

An important task that is not deeply discussed in the vast majority of the surveyed works, but crucial for all MOMHOs, is the solid parameters setting, which is an important issue for MOMHOs to get competitive results. In most revised works, the parameters are either empirically selected or taken from state-of-the-art references. The robustness of a MOMHO to changes in its parameters has been studied only in [44].

The computational complexity of the MOMHOs used for controller tuning, depends on the number of objective functions M, the size of the population or swarm N, and the size of the external archive L (if included). After analyzing the behavior of the MOMHOs in Tables 7-10, the complexity of the algorithms can be generalized for the worst-case scenario as:

- $O(MN^2)$ for alternatives that converge the original population or swarm without any archiving system.
- $O(M(N+L)^2)$ for algorithms that incorporate an external archive to store non-dominated solutions.

4.5.2. Constraint-handling techniques

Regarding the constraint-handling mechanism implemented in the MOMHOs shown in Tables 7-10, NSGA-II, and MATLAB MOGA have built-in handling methods, while for a few MOGAs it is common the use of penalization methods (they punish the value of the objective functions proportionally to the number of violated constraints).

As in MOGAs, only a few MOPSOs include a constraint-handling technique based on penalization, but the use of another strategy to deal with constraints can be observed, i.e., the epsilon-constrained method [162].

The preferred constraint-handling mechanism in MODE is the Pareto dominance generalization of the tournament selection operator in [163] (where feasible configurations are considered fitter than infeasible ones).

As described before in Section 4.4, a considerable number of controller tuning MOPs are constrained, then constraint-handling techniques are necessary to discover feasible controller configurations [164, 165]. Nevertheless, a common approach to handle constraints adopted in several reviewed works consists of filtering infeasible solutions after the $\mathcal{P}^A$ is obtained, instead of incorporating a mechanism into the MOMHO to discard these solutions during the metaheuristic optimization process. This approach can be considered risky, in the sense that solutions in the $\mathcal{P}^A$ could be all infeasible. An approach used to a lesser extent consists of the *a priori* feasible initialization of solutions.

4.6. Pareto front approximation

The result obtained by a MOMHO is a Pareto front approximation $\mathcal{PF}^A$ with the best trade-off solutions (Fig. 7-E).

Since there is no universal MOMHO whose performance is the best so far to solve all kinds of problems [51], it is worth to test different known or novel alternatives to compare them and identify their specific advantages and drawbacks [175]. The above allows the selection of the most promising alternative for solving a particular controller tuning problem, which improves the quality of results.

Table 7: Multi-objective Genetic Algorithm.

Ref.	Variant	Features or improvements
[87]	NSGA-II	The state-of-the-art Non-dominated Sorting Genetic Algorithm II (NSGA-II) [65].
[89]	NSGA-II	The state-of-the-art Non-dominated Sorting Genetic Algorithm II (NSGA-II) [65].
[90]	NSGA-II	The state-of-the-art Non-dominated Sorting Genetic Algorithm II (NSGA-II) [65].
[91]	NSGA-II	The state-of-the-art Non-dominated Sorting Genetic Algorithm II (NSGA-II) [65].
[92]	NSGA-II	The state-of-the-art Non-dominated Sorting Genetic Algorithm II (NSGA-II) [65].
[93]	NSGA-II	Blend crossover, initial population generated by the Tyreuse-Luyben controller tuning method.
[94]	Chaotic NSGA-II	Henon map random numbers in the mutation and crossover.
[95]	Chaotic NSGA-II	Henon map random numbers in the mutation and crossover.
[96]	Chaotic NSGA-II	Iterative Chaotic Map with Infinite Collapses for local searches over offspring solutions.
[83]	NSGA-II	Constraint-handling based on penalization.
[45]	EMOGA, NSGA-II	Fitness sharing.
[97]	NSGA-II	The state-of-the-art Non-dominated Sorting Genetic Algorithm II (NSGA-II) [65].
[98]	MATLAB MOGA	The Multi-objective Genetic Algorithm (MOGA) of the Global Optimization Toolbox in MATLAB.
[99]	NSGA-II	The state-of-the-art Non-dominated Sorting Genetic Algorithm II (NSGA-II) [65].
[100]	MOGA	GENOCOP genetic operators, fitness sharing.
[101]	NSGA-II	The state-of-the-art Non-dominated Sorting Genetic Algorithm II (NSGA-II) [65].
[102]	MOGA	The state-of-the-art Multi-objective Genetic Algorithm (MOGA) [59].
[103]	MATLAB MOGA	The Multi-objective Genetic Algorithm (MOGA) of the Global Optimization Toolbox in MATLAB.
[104]	NSGA-II	The state-of-the-art Non-dominated Sorting Genetic Algorithm II (NSGA-II) [65].
[105]	MOGA	Fitness sharing.
[106]	MOGA	The state-of-the-art Multi-objective Genetic Algorithm (MOGA) [59].
[107]	NSGA-II	The state-of-the-art Non-dominated Sorting Genetic Algorithm II (NSGA-II) [65].
[42]	MOGA	Solutions ranking for selection using an *a priori* preference articulation to determine pertinence levels, extended intermediate recombination operator, BGA mutation.
[43]	MOGA	Fitness sharing, multi-point crossover, two-bit mutation.
[108]	MOGA	Intermediate crossover, tournament selection, constraint-dependent mutation, sub-populations with a migration scheme.
[109]	MOGA	External archive, arithmetic crossover, uniform mutation, initial population generated by the control algorithms such as skyhook control, feedback linearization and sliding mode control.
[110]	MRCD-GA	Binary tournament, parallel island strategy with migration scheme, cut-off point crossover, external archive.
[111]	MRCD-GA	Binary tournament, parallel island strategy with migration scheme, cut-off point crossover, external archive.
[112]	MOGA	The state-of-the-art Multi-objective Genetic Algorithm (MOGA) [59].
[113]	NSGA-II	The state-of-the-art Non-dominated Sorting Genetic Algorithm II (NSGA-II) [65].
[114]	MOGA	Parallel population with a stochastic immigration mechanism.
[115]	MOGA	Stochastic Universal Sampling selection, reduced-surrogate shuffle crossover, mating restriction, uniform mutation.
[135]	NSGA-II	The state-of-the-art Non-dominated Sorting Genetic Algorithm II (NSGA-II) [65].
[137]	LCNSGA-III	A variant of the state-of-the-art algorithm Non-dominated Sorting Genetic Algorithm III (NSGA-III) [65], initialization using the latin hypercube sampling, chaos theory is used to enehance the global search and the local exploration.
[138]	RPD-NSGA-II	The state-of-the-art Reference Point-based Dominance Non-dominated Sorting Genetic Algorithm II [166].
[139]	NSGA-II	The state-of-the-art Non-dominated Sorting Genetic Algorithm II (NSGA-II) [65].
[143]	NSGA-II	The state-of-the-art Non-dominated Sorting Genetic Algorithm II (NSGA-II) [65].
[145]	ev-MOGA	External archive with pruning mechanism based on epsilon dominance.
[146]	NSGA-II	The state-of-the-art Non-dominated Sorting Genetic Algorithm II (NSGA-II) [65].
[147]	NSGA-II	The state-of-the-art Non-dominated Sorting Genetic Algorithm II (NSGA-II) [65].

Table 8: Multi-objective Particle Swarm Optimization.

Ref.	Variant	Features or improvements
[116]	AGPSO	External grid archive, global best solution obtained with roulette selection, personal best assigned using Pareto dominance, non-uniform mutation.
[117]	MOPSO	Velocity updates based on the global best solution and two random solutions, Gaussian mutation, global best solution selection based on isolation, periodical updates of the inertia weight and learning factors, finite-length external archive.
[46]	Ingenious-MOPSO	Global best selection based on isolation, personal best selection using the Sigma method, turbulence operator to replace solutions by randomly generated ones, finite-length external archive with fuzzy elimination technique.
[118]	Ingenious-MOPSO	Global best selection based on isolation, personal best solution selection using the Sigma method, turbulence operator to replace solutions by randomly generated ones, finite-length external archive with fuzzy elimination technique.
[84]	MOPSO	Finite-length external archive, personal best assigned using Pareto dominance, global best selection from the external archive using the Technique for Order of Preference by Similarity to Ideal Solution.
[44]	MOPSO	External hyper-cube grid archive, constraint-handling based on penalization, global best selection using a roulette over the hyper-cube regions isolation, personal best assigned using Pareto dominance, linear updates of the inertia weight.
[119]	MaPSO	External archive, corner archive with extremes of the Pareto front, constraint-handling based on penalization, initial population generated with stable closed-loop system configurations, global best selected as the nearest solution to two randomly selected solutions from the corner archive, personal best assigned using Pareto dominance.
[120]	MOPSO	External archive, global best selected randomly from the archive, personal best assigned using Pareto dominance.
[41]	I-MOPSO	External hyper-cube grid archive, uses the Population Spatial Diversity to establish exploitation and exploration probabilities of solutions, only the global or personal best solution is used to update the velocity depending on probability, global best selection using a roulette over the hyper-cube regions isolation, personal best assigned using Pareto dominance.
[121]	MOPSO	External archive, global guide selection method to obtain the global best solution, global best selected from the archive as the solution with the samllest Euclidean distance to the nadir vector, personal best assigned using Pareto dominance.
[122]	MOPSO	The state-of-the art Multi-objective Particle Swarm Optimization (MOPSO) [167].
[85]	2LB-MOPSO	External archive, two local best solutions used to update the velocity, first local best is the solution with the highest crowding distance, the second local best is the solution with the smallest Euclidean distance to the first local best (in the search space), local bests update based on contribution frequency of particles to the archive, epsilon-constrained method as constraint-handling technique.
[123]	MOHEPSO	External archive, pruning mechanism based on fuzzy elimination technique, dynamic inertia weight and learning factors, personal best selection from the archive using Sigma method, global best selection based on isolation, particle updates using on PSO, bee colony and multi-crossover at given probabilities.
[135]	MOPSO	External archive, personal best assigned using Pareto dominance, global best selection from the external archive based on sorting, fixed inertia weight.
[141]	MOPSO	Global best selection based on isolation, personal best selection using the Sigma method, turbulence operator to replace solutions by randomly generated ones, finite-length external archive with dynamic elimination technique based on niches.

Table 9: Multi-objective Differential Evolution.

Ref.	Variant	Features or improvements
[124]	PMODE	External archive, based on DE/rand/1/bin, isolation-based population pruning, random solutions replace pruned solutions.
[125]	spMODE	External spherical grid archive, based on DE/rand/1/bin.
[75]	spMODE-II, IB-MODE, RP-spMODE	spMODE-II: Physical programming is included in spMODE to decide which solutions must be pruned when two or more solutions share the same archive grid sector. IB-MODE: It uses a binary epsilon indicator to assign fitness values to solutions. RP-spMODE: External spherical grid archive, based on DE/rand/1/bin, uses reference point distance to decide which solutions must be pruned when two or more solutions share the same archive grid sector.
[126]	spMODE-II	Physical programming is included in spMODE to decide which solutions must be pruned when two or more solutions share the same archive grid sector.
[47]	spMODE	External spherical grid archive, based on DE/rand/1/bin.
[127]	spMODE-II, IB-MODE$_1$, IB-MODE$_2$	spMODE-II: Physical programming is included in spMODE to decide which solutions must be pruned when two or more solutions share the same archive grid sector. IB-MODE$_1$/IB-MODE$_2$: External archive, based on DE/rand/1/bin, use binary indicator in which different preference vectors determine the quality of solutions.
[128]	spMODE	External spherical grid archive, based on DE/rand/1/bin.
[129]	MODE1, MODE2, MODE3	MODE1: External archive, based on DE/rand/1/bin, target vector obtained from the archive, solutions used in the difference vector selected from the current population. MODE2: External archive, based on DE/rand/1/bin, target vector obtained from the current population, solutions used in the difference vector selected from the archive. MODE3: External archive, based on DE/rand/1/bin, target vector and solutions used in the difference vector selected from the archive.
[130]◇	MDE-IIP-NDSM	External archive, nine variants of MDE-IIP-NDSM (based on DE/rand/1/bin, DE/best/1/bin, DE/rand/1/exp, DE/best/1/exp, DE/current-to-best/1/bin, DE/current-to-rand/1/bin, DE/current-to-best/1, DE/current-to-rand/1, and DE/rand/2/dir), used in online optimization, some promising archived solutions are included in initial population after the problem updates.
[86]	spMODE-II	Physical programming is included in spMODE to decide which solutions must be pruned when two or more solutions share the same archive grid sector.
[88]	spMODEx	External spherical grid archive, physical programming in the pruning mechanism, based on DE/rand/1/bin.
[135]	MOEA/D	The state-of-the-art Multi-objective Evolutionary Algorithm based on Decomposition (MOEA/D) [168].
[143]	GDE3	The state-of-the-art third evolution step of Generalized Differential Evolution (GDE3) [169].
[144]	HV-MODE	External archive, swaps the generation strategies DE/current-to-rand and DE/current-to-pbest/1 depending on the archive size, pbest is selected based on the hypervolume contribution of solutions in the archive.

◇ Online optimization approach.

Table 10: Other Multi-objective Optimizers.

Ref.	Variant	Features or improvements
[131]	IMOSA	Generation mechanism based on orthogonal experimental solutions, Pareto based scoring function to measure the performance of candidate solutions and exploit its neighborhood.
[132]	SPEA, DPEA, GSA	SPEA: The state-of-the-art Strength Pareto Evolutionary Algorithm (SPEA) [156]. DPEA: The state-of-the-art Dual Population Evolutionary Algorithm (DPEA) [158]. GSA: The state-of-the-art Gravitational Search Algorithm (GSA) [159].
[133]	MOEO	The state-of-the-art Multi-objective Extremal Optimization [160]
[134]	MOSTA	External archive, pruning mechanism based on solution crowding, state transformations applied to a random archived solution.
[135]	AGE-II, SPEA-II, RVEA	AGE-II: The state-of-the-Approximation-Guided evolutionary algorithm II (AGE-II) [161]. SPEA-II: The state-of-the-art Strength Pareto Evolutionary Algorithm II (SPEA-II) [157]. RVEA: The state-of-the-art Reference Vector Guided Evolutionary Algorithm (RVEA)[170].
[136]	MOCSA	Multi-objective Cuckoo Search Algorithm, multi-objective variant of the state-of-the-art Cuckoo Search Algorithm (CSA) [171], uses Pareto dominance for fitness comparisons.
[140]	MOWOA, MOGWO	MOWOA: Multi-objective Whale Optimization Algorithm, multi-objective variant of the state-of-the-art Whale Optimization Algorithm (WOA) [172], external archive. MOGWO: Multi-objective Grey Wolf Optimizer, multi-objective variant of the state-of-the-art Grey Wolf Optimizer (GWO) [173], external archive, best solution from Pareto best search agents are selected from the archive using the roulette wheel.
[142]	Multi-objective ABC	Multi-objective Artificial Bee Colony, multi-objective variant of the state-of-the-art Artificial Bee Colony (ABC) [174], finite-length archive with pruning mechanism based on isolation, onlooker bees probabilistically select solutions in the archive.

The performance of a MOMHO strongly depends on the quality of its obtained Pareto front approximation. In this way, performance comparisons among different MOMHOs are carried out in this step over their obtained $\mathcal{PF}^A$s. Because of the MOMHOs stochastic nature, it is also recommended to provide descriptive [176] and inferential [177] statistical evidence about different samples of the obtained results to draw solid conclusions. Moreover, it is necessary to use different performance criteria to avoid contradictions among them [51] and ensure that a MOMHO can find Pareto front approximations with the desired features and quality levels.

About 40% of the reviewed works include this type of necessary performance comparisons in the solution of the controller tuning MOP. They can be qualitative or quantitative, as observed next:

- Qualitative comparisons: This class includes comparative analyses based on the subjective criterion of the designer. In [94, 96], the performance comparisons are among variants of the same MOMHO. On the other hand, works in [140, 141, 143, 145, 108, 118, 129, 132] compare MOMHOs from different natures.

- Quantitative comparisons: They use quality metrics over the $\mathcal{PF}^A$ to give a quantitative measure of the MOMHO performance. Due to the stochastic nature of MOMHOs, it is important to provide statistical evidence of the performance over several runs [177]. Works in [85, 75, 127, 130] opt for a single performance metric over several runs of MOMHOs from different natures. Nevertheless, it is highly recommendable to use more than one metric to ensure the reliability of results. In such a way, the works in [41, 42, 45, 46, 47, 116, 128, 135, 144] use statistical comparisons based on the performance values of several metrics over diverse runs of MOMHOs from different natures. Another additional approach, which could be a little less useful according to the theorem presented in [51], performs comparisons using benchmark MOPs instead of using the proper controller tuning MOP [44, 117, 137, 142].

By the end of this stage, a Pareto front approximation must be provided to make future decisions. This front is taken as the one with the highest quality based on the performance metrics information. Nevertheless, a useful way to take this approximation is presented in [41], where the fronts obtained for each run of the MOMHO (originally used to perform comparisons) are merged and then filtered based on Pareto dominance to generate a global front. This resulting front can include a wider variety of trade-off solutions.

4.7. Decision maker

As commented before, the Pareto front approximation $\mathcal{PF}^A$ obtained in the previous step is a finite set with a reasonably large amount of solutions. Nevertheless, only one controller configuration can be implanted in the final application at a time.

By using preference handling methods, the decision-maker in Fig. 7-F (this can be a human designer or an automated entity) selects the best trade-off solution from the Pareto front approximation.

Table 11 summarizes the preference handling methods included in the surveyed works. All works use an *a posteriori* preference handling with both empirical and utility-based decisions:

- Empirical: As a reminder, this class is based on the human experience, and it must be noticed that 41% of the works belong to it. Typical methods include trial and error approaches, visualization of the $\mathcal{PF}^A$ trade-offs (when multiple objectives are considered, the use of tools like Level Diagrams (LD) [125] can be suitable), and subjective designer analysis.

- Utility-based: This class is related to an automated decision maker, and 59% of works opt for it. In this class, a popular method is based on the choice of a solution according to its position in the $\mathcal{PF}^A$. Common alternatives include solutions near the extremes (to prioritize an objective more than the others), and the mean or knee solutions (when looking for more balanced trade-off). Another widely used method consists in the use of a tiebreaker criterion. Since all trade-offs in $\mathcal{PF}^A$ are considered equally good, the tiebreaker criterion allows the finding of a relative fittest solution regarding a quantitative evaluation of all alternatives. The use of a reference point is also an interesting method that allows designers to propose an ideal performance of the controller configuration. Based on this, the best trade-off is selected as the solution closest to the designer's wishes. Finally, other less common methods use fuzzy approaches, clustering techniques, and aggregation functions over the objective function values (e.g., weighted sums and physical programming).

4.8. Validations of the final controller configuration

In this last stage, the best controller parameters $\vec{p}^*$, selected by the decision-maker, can be implanted in the final application. Some subsequent validations are necessary to ensure that the desired controller operation is reached. Such validations include comparisons with different controller configurations from different tuning approaches, both in simulation or experimentation (laboratory tests with a real prototype):

- Simulated validations: Although all controller configurations are implicitly validated in simulation (i.e., a dynamic simulation is performed to evaluate the controller configuration performances), comparative analyses are hard to find (only in 24% of the works).

  Most of them are included in [47, 84, 85, 116, 119, 120, 121, 125, 128], where better performances are achieved with the multi-objective approach when compared with other tuning methods such as Zigler Nichols, Cohen-Coon, Murrill, global optimization methods, among others.

Table 11: Preference handling methods.

Ref.	Decision maker type †	Ref.	Decision maker type †
[87]	*A posteriori* / Utility-based / Tiebreaker criterion	[114]	*A posteriori* / Utility-based / Tiebreaker criterion
[89]	*A posteriori* / Utility-based / Solution position	[115]	*A posteriori* / Utility-based / Solution position
[90]	*A posteriori* / Empirical / Subjective analysis	[116]	*A posteriori* / Utility-based / Fuzzy
[91]	*A posteriori* / Utility-based / Solution position	[117]	*A posteriori* / Utility-based / Tiebreaker criterion
[92]	*A posteriori* / Utility-based / Solution position	[46]	*A posteriori* / Utility-based / Tiebreaker criterion
[93]	*A posteriori* / Empirical / Trial and error	[118]	*A posteriori* / Utility-based / Solution position
[94]	*A posteriori* / Utility-based / Solution position	[84]	*A posteriori* / Utility-based / Reference point
[95]	*A posteriori* / Utility-based / Solution position	[44]	*A posteriori* / Utility-based / Solution position
[96]	*A posteriori* / Utility-based / Solution position	[119]	*A posteriori* / Empirical / Subjective analysis
[83]	*A posteriori* / Utility-based / Tiebreaker criterion	[120]	*A posteriori* / Utility-based / Tiebreaker criterion
[45]	*A posteriori* / Empirical / Subjective analysis	[41]	*A posteriori* / Utility-based / Solution position
[97]	*A posteriori* / Utility-based / Clustering	[121]	*A posteriori* / Utility-based / Weighted sum
[98]	*A posteriori* / Utility-based / Reference point	[122]	*A posteriori* / Empirical / Subjective analysis
[99]	*A posteriori* / Utility-based / Solution position	[85]	*A posteriori* / Utility-based / Tiebreaker criterion
[100]	*A posteriori* / Empirical / Subjective analysis	[123]	*A posteriori* / Utility-based / Solution position
[94]	*A posteriori* / Utility-based / Solution position	[124]	*A posteriori* / Empirical / Subjective analysis
[102]	*A posteriori* / Empirical / Visualization	[125]	*A posteriori* / Empirical / Visualization
[103]	*A posteriori* / Utility-based / Reference point	[75]	*A posteriori* / Empirical / Visualization
[104]	*A posteriori* / Empirical / Subjective analysis	[126]	*A posteriori* / Empirical / Subjective analysis
[105]	*A posteriori* / Empirical / Subjective analysis	[47]	*A posteriori* / Empirical / Visualization
[106]	*A posteriori* / Empirical / Subjective analysis	[127]	*A posteriori* / Empirical / Visualization
[107]	*A posteriori* / Utility-based / Solution position	[128]	*A posteriori* / Empirical / Visualization
[42]	*A posteriori* / Utility-based / Tiebreaker criterion	[129]	*A posteriori* / Empirical / Subjective analysis
[43]	*A posteriori* / Utility-based / Solution position	[130]	*A posteriori* / Utility-based / Tiebreaker criterion
[108]	*A posteriori* / Empirical / Visualization	[86]	*A posteriori* / Utility-based / Physical programming
[109]	*A posteriori* / Utility-based / Solution position	[88]	*A posteriori* / Empirical / Visualization
[110]	*A posteriori* / Utility-based / Tiebreaker criterion	[131]	*A posteriori* / Utility-based / Tiebreaker criterion
[111]	*A posteriori* / Empirical / Subjective analysis	[132]	*A posteriori* / Utility-based / Physical programming
[112]	*A posteriori* / Utility-based / Reference point	[133]	*A posteriori* / Empirical / Subjective analysis
[113]	*A posteriori* / Utility-based / Fuzzy	[134]	*A posteriori* / Empirical / Subjective analysis
[135]	*A posteriori* / Empirical / Subjective analysis	[136]	*A posteriori* / Empirical / Subjective analysis
[137]	*A posteriori* / Utility-based / Weighted sum	[138]	*A posteriori* / Utility-based / Solution position
[139]	*A posteriori* / Empirical / Visualization	[140]	*A posteriori* / Empirical / Subjective analysis
[141]	*A posteriori* / Utility-based / Solution position	[142]	*A posteriori* / Empirical / Visualization
[143]	*A posteriori* / Utility-based / Solution position	[144]	*A posteriori* / Utility-based / Solution position
[145]	*A posteriori* / Empirical / Visualization	[146]	*A posteriori* / Utility-based / Solution position
[147]	*A posteriori* / Utility-based / Reference point		

† Decision maker type: Preference articulation / Class / Method.

On the other hand, the controller tuned in [124, 142, 144, 136] is compared with a different controller (i.e., a controller with a different structure than the one tuned using multi-objective optimization). Even this comparison could be considered unfair, it may work as a performance reference.

- Experimental validations: Experimental evidence about the performance of the obtained controller configurations is even harder to find. Only 8% of the surveyed works include such experiments.

  The works in [100, 121, 126, 145] perform comparisons with other tuning methods, while in [105, 124] controller performances are contrasted with those obtained with different controller structures. In any case, controllers from the multi-objective optimization tuning approach exhibit outstanding performance.

5. Conclusions and future directions

5.1. Conclusions

5.1.1. About the interest in this area

The annual trend of works related to the multi-objective meta-heuristic optimization-based controller tuning is shown in Fig. 8. It is observed that the research interest in this area has increased since 2011.

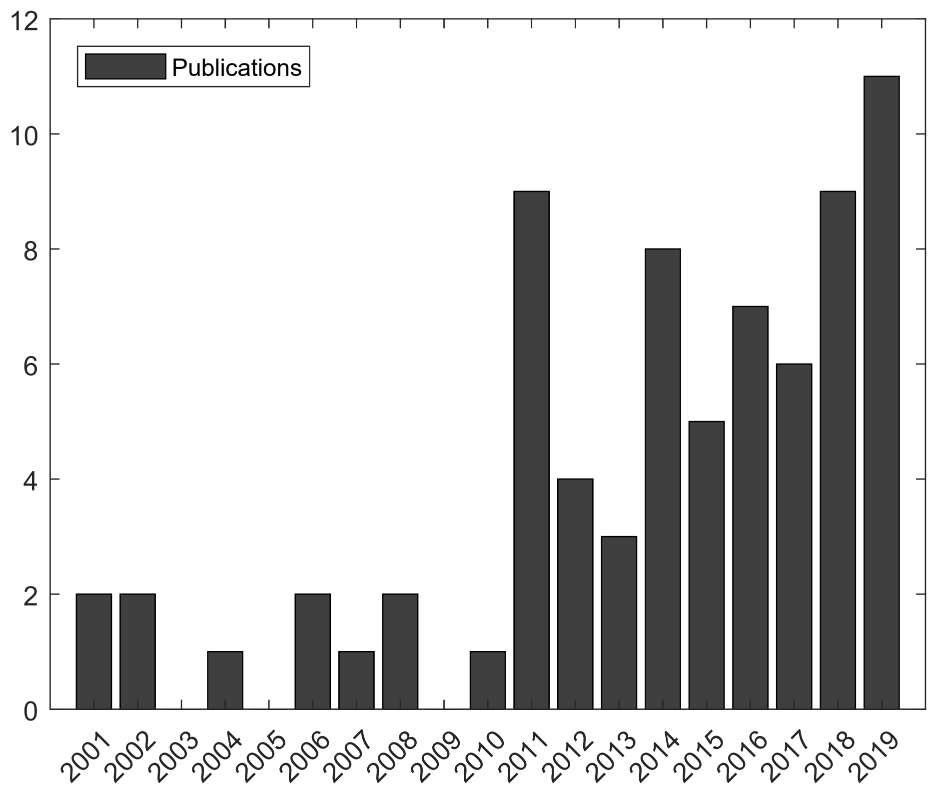

Figure 8: Distribution of the number of research items from year 2001 to 2019.

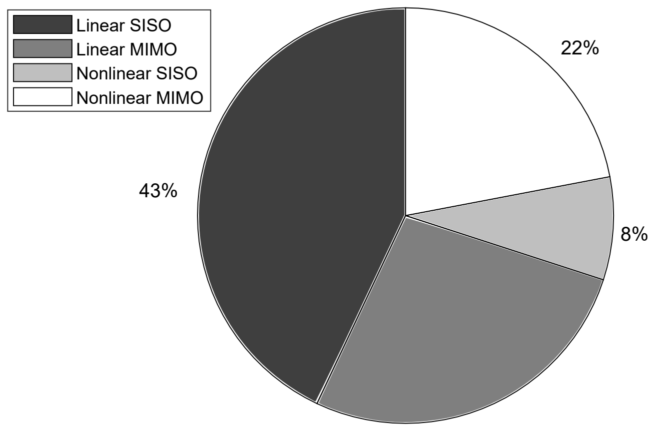

Figure 9: Percentage of linear SISO, nonlinear SISO, linear MIMO and nonlinear MIMO dynamic systems adopted for the controller tuning process.

5.1.2. Studied control systems

The reviewed works study the control of a wide variety of dynamic systems. Fig. 9 shows that the vast majority are linear systems. Therefore, it is natural that the most adopted controller structures to handle them are also linear, as observed in Fig. 10. Using linear control systems allows working with more general models that could simplify their analysis and take advantage of their frequency-domain representations. On the other hand, nonlinear systems have more complex structures, and the tools available for analysis are harder to deal with, and they are also limited. However, they can deal with an unlimited number of real-world control problems.

The tendency of works concerning the number of inputs and outputs of the controlled systems can also be observed in Fig. 9. For linear plants, the number of studied SISO systems is more than twice the number of MIMO ones. The opposite happens to nonlinear controllers since the number of MIMO plants is twice the number of SISO ones. The above may be due to the recurrent use of benchmark plants in the case of linear SISO systems, while in the linear MIMO case, the plants come from more practical applications. Concerning the nonlinear MIMO systems, most of them come from mechanical and robotics applications where dynamic systems with two or more degrees of freedom are studied. On the other hand, most of the nonlinear SISO systems correspond to test plants proposed in the specialized literature.

In Fig. 10 is observed that the PID-like structures are popular alternatives among linear controllers because of their high performance and easy implementation. In the case of nonlinear controllers, the fuzzy approach is the most preferred.

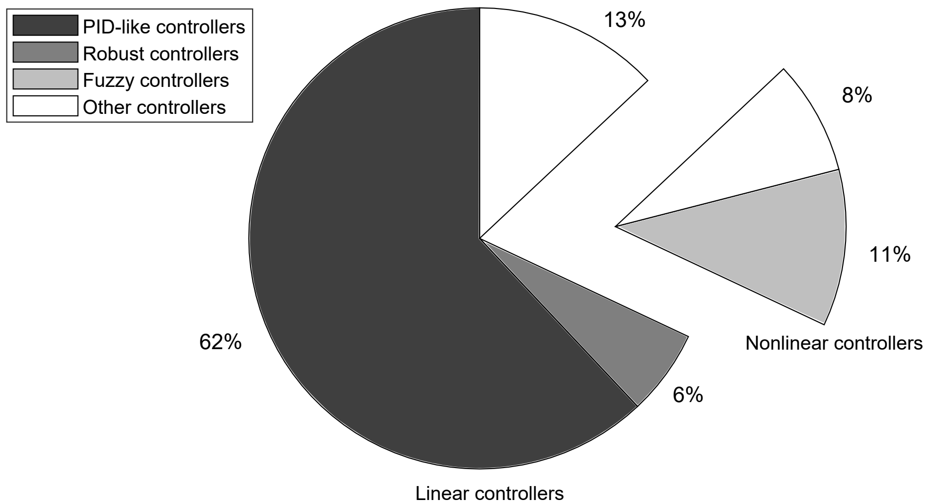

Figure 10: Percentage of linear and nonlinear controller structures adopted for the controller tuning process.

5.1.3. Multi-objective problems

The number of design criteria involved in the tuning problem definitions is variable, as observed in the reviewed works. 80% corresponds to MOP definitions, while the remaining 20% to many-objective problems.

In most cases, standard control system metrics and performance indexes in both time and frequency domains, are used as the problem objectives. It is then sought that systems are less prone to errors, more robust to disturbances, less sensitive to noise, and with better time response rates and energy efficiency. Other specific performance measurements are also adopted, and they aim to reduce the complexity and implementation cost of a given controller structure.

It is important to mention that the use of MIMO systems often implies an increasing number of tuning specifications since the number of performance measurements must be scaled to the number of inputs and outputs. In some works with MIMO systems, the performance measurements related to a group of inputs and outputs are encompassed in a single objective. Nevertheless, it can be worth to take them separately and then state a many-objective problem.

About 34% of the reviewed works contain non-boundary constraints. Most of them are related to the overall control system stability and desirable ranges of performance indexes (which are not necessarily used as design criteria). In many works, limits of performance measurements different from those established as objectives are also considered as constraints.

Concerning the number of design variables required to describe the tunable parameters of a controller, for a SISO linear controller is common to find PID-like alternatives that require between three and five variables. When a SISO nonlinear system is tuned, in most of the cases, the involved variables are less or equal than five. Still, when more complex control strategies such as fuzzy ones are adopted, the number of variables increases considerably. For a MIMO controller, the number of design variables is proportional to the number of inputs and outputs. Despite the controller type, the dimensionality of the tuning problem is low. In two cases, the number of design variables exceeds the order of thousands, as in [30] and [109]. Then, large-scale controller tuning problems are very hard to find in the specialized literature.

5.1.4. Multi-objective meta-heuristics

Fig. 11 shows the trend in the use of different meta-heuristics in the controller tuning problem. GA based multi-objective optimizers head the list of preferred meta-heuristics followed by PSO and DE based ones. This order is closely related to the appearance date of the early multi-objective versions of GA, PSO, and MODE. After them, there is a small group of other more recently proposed multi-objective optimizers.

Most of the used MOMHOs adopt a dominance-based search approach, while metric-driven, decomposition-based, and hybrid MOMHOs are presented but barely studied.

It is important to mention that dominance-based MOMHOs achieve suitable results when solving many-objective controller tuning problems despite the use of Pareto dominance, which often loses effectiveness in these kinds of problems [178]. Therefore, it might be interesting to study the decomposition-based, metric-driven, and even hybrid MOMHOs in the controller tuning process.

Although several MOPs in the studied works are constrained, there is a lack in the use of constraint-handling techniques beyond the penalization methods during the optimization process. The constraints in most of the cases are used to refine the obtained optimal solution set, but only a few works include mechanisms in the MOMHOs to guide the search to the feasible region of the search space.

5.1.5. Evaluation of the optimizer performance

Since the performance of a given optimizer depends on the problem that must be solved, it is important to test different alternatives before opting for a single one. The above is performed in about the 38% of the reviewed works. They include comparisons with several meta-heuristics regarding the Pareto front approximation quality (this is measured using well-known metrics). The compared alternatives are commonly optimizers of different natures and, to a lesser extent, variants of the main MOMHO. Nevertheless, comparisons with different types of meta-heuristics that include distinct search approaches (decomposition-based, metric-driven, or hybrid) could provide better feedback information about the most promising alternatives and their search mechanisms. About the quality metrics, it is highly recommended to opt for at least two alternatives to cover most of the desirable features of the Pareto front approximation (capacity, convergence, diversity, and pertinence), as many of the reviewed works do.

An important issue when comparing optimizers concerns the use of benchmark MOPs that are not directly related to the controller tuning problem.

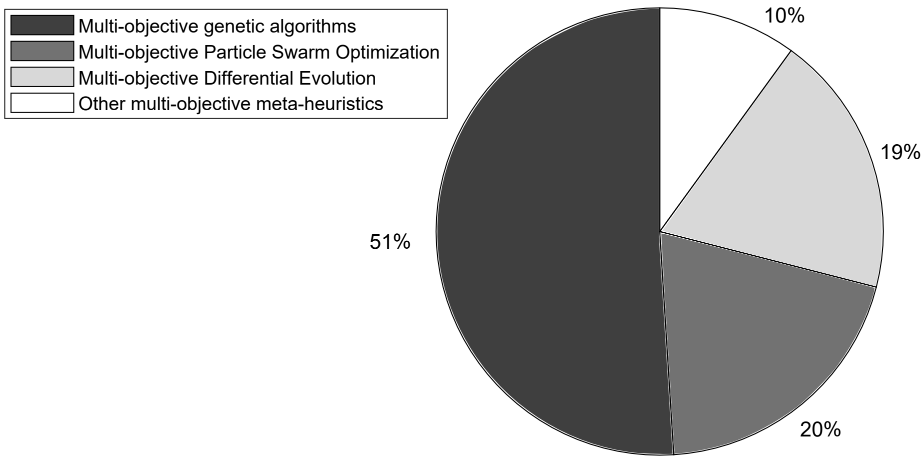

Figure 11: Percentage of research items that use multi-objective versions of GA, DE, PSO, and other types of meta-heuristics in the controller tuning problem.

Although benchmark MOPs can be used to obtain a general sense of the operation of a given MOMHO (i.e., without specifying a field of application), it can be more useful to focus on a test-bed of a particular controller tuning problem (e.g., problems that consider different operation modes, uncertainties, or disturbances). At industrial levels, the above could increase the reliability of results in the change from a simulation test to the experimental validation of the tuned controller. Alternatively, the above reliability can be increased if a Hardware-in-the-Loop simulation (a real-time simulation of a complete plant model performed in dedicated hardware), before the experimental validation, is performed as in [126].

Since MOMHOs are stochastic techniques, a descriptive and inferential statistical analysis is necessary to perform solid comparisons to draw relevant conclusions about the obtained results [177]. Only a few works provide statistical evidence that sustains the presented comparisons.

5.1.6. Decision making

Decision making is a crucial task in the controller tuning process because it determines the proper operation of the final control application.

All studied works opt for *a posteriori* preference handling. In addition to this preference handling approach, some versions of MODE include an *a priori* preference articulation to establish desirable regions of the Pareto front. Then they are used to guide the MOMHO search towards pertinent solutions (those which incorporate physical programming).

When an offline optimization approach is adopted in the controllers tuning, the decision making can be performed either through an experienced human or an automated designer. In the adaptive tuning methods based on the online optimization approach like the ones in [130, 144], such decisions must be made in an automated way.

5.1.7. Final controller configuration validations

For each of the reviewed works, the selected parameter configuration from the Pareto front approximation is validated in simulation. In 24% of the reviewed works, the tuned controller is also compared in simulation with controllers from state-of-the-art tuning methods or other control structures. Only about 8% of works include experimental laboratory tests within real environments.

eses

5.2. Future directions

The possible future directions in the multi-objective meta-heuristic controller tuning are discussed below:

5.2.1. Meta-heuristic optimizers

- Most of the used MOMHOs in the reviewed literature are all based on the dominance-based search approach. Moreover, many controller tuning problems are many-objective ones, and the dominance-based approach loses effectiveness as the number of objectives grows. Nevertheless, MOMHOs from the decomposition-based, metric-driven, and hybrid search approaches have proved to be capable of handling these kinds of problems with competitive performances [179]. Then, the exploration of these MOMHOs is suggested for both, the multi-objective and many-objective cases.

- For many applications, a fine controller tuning is required. This necessity can be satisfied by performing a local search around a given controller configuration. Memetic algorithms are techniques that incorporate local search methods in the classical meta-heuristics operation [180]. Therefore, it may be beneficial for these kinds of applications to try memetic algorithms and determine their effectiveness.

- The use of special operators, beyond the ones adopted in the presented MOMHOs, may improve the search effectiveness regarding the exploration and the exploitation of the search space in the controller tuning problem, as it has done in other problems [181]. Special operators can then help to find better and diverse trade-offs, which can be translated to a greater number of alternatives for the decision-maker. Since the use of special operators is not found in the reviewed works, their inclusion in the MOMHOs is suggested.

- Concerning the constraint-handling techniques, there is an important lack in the controller tuning process. Many of the studied works use penalization approaches or check solution feasibility after the computation of the Pareto front approximation. Nevertheless, the computational effort used to search within the whole search space can be harnessed if the search is induced to the feasible region by incorporating suitable constraint-handling techniques in the MOMHOs. Among those methods, the alternatives described in [182, 164, 165] could be considered for further implementation in the MOMHO for controller tuning problems.

- Another observed issue is related to the lack of large-scale optimization mechanisms in problems with high design space dimensionality. This lack can be related to solutions far from optimal and with a high computational effort. Although large-scale problems are rare in the controller tuning context, there are interesting approaches in [183] that can be addressed.

- The use of unsuitable MOMHO parameters may end up in poor performance when solving a controller tuning MOP. In the reviewed works, the MOMHO parameters are either empirically selected or are taken from state-of-the-art references. Still, there is not enough information about the implications of using different parameter sets. For this reason, the study of different MOMHO parameter settling approaches like the ones described in [184] is a real necessity.

- There is also a lack of information on how to handle the boundary constraints in the controller tuning MOP. An accurate establishment can reduce the overfitting and accelerate the searching for adequate controller configurations [185]. Since in the surveyed works such constraints appear to be selected based on the designer experience, a well-defined methodology to determine them could be very useful.

5.2.2. Multi-objective online controller tuning

- The controller tuning process through online optimization has been scarcely studied. Nevertheless, there is a wide variety of applications that require very high accuracy rates, which are only achievable by online controller tuning (also referred to in the specialized literature as adaptive control [186]). Then, it is recommendable to explore in a deeper way the online multi-objective controller tuning approach.

- One of the main issues in an online optimization tuning process is the required computational burden. It is necessary to find a way to make this process more efficient either by improving the search mechanism of the MOMHOs (e.g., using surrogate-assisted metaheuristics [187]) or by using high-performance computing techniques [188].

5.2.3. Decision-maker

- In offline and especially in the online optimization approach, the inclusion of problem-context-dependent interactive/progressive preference handling methods could help to increase the pertinence level of solutions on demand. It is suggested to test progressive preference handling approaches [189], which have not been studied in the multi-objective controller tuning.

- Empirical or human-based preference handling methods have been extensively used in many reviewed works. Nevertheless, It can be interesting to take advantage of the numerous automated approaches in the specialized literature [36] since they can be helpful to prevent issues when not enough preference information is provided and also to release pressure on inexperienced decision-makers.

5.2.4. Performance validations

- The statistical validation of the MOMHOs performance is absent in most of the reviewed works. Nevertheless, this is considered an important task to find potentially good MOMHO alternatives [177] for particular controller tuning problems, and it is then recommendable to be addressed in future works.

- Some important quality metrics such as the generational distance require the true Pareto front. Nevertheless, for real-world problems, this is always unknown. It can be useful to know this front for a set of controller tuning problems to aid the design and test of novel MOMHOs and to study their explorative and exploitative capabilities. Hence, it is suggested to develop a well defined multi-objective controller tuning benchmark for both offline and online optimization approaches.

- Noise and disturbances are undesirable and unavoidable characteristics of real-world environments. Therefore, it is necessary to perform experimental tests within these environments to test in-depth the sensitivity and robustness of the optimized control system under real noise, disturbances, uncertainties, and other conditions, which most of the time are not considered in the tuning process. Software-in-the-loop, processor-in-the-loop, and hardware-in-the-loop simulations can provide more realistic information for the controller tuning [126] in future works.

5.2.5. Controllers and plants

- The studied works mostly adopt linear SISO systems. There is still pending work about linear MIMO and nonlinear SISO and MIMO systems. The controller tuning problem for highly nonlinear systems is also an exceptional candidate to be solved by meta-heuristics and can exalt their usefulness.

- The use of MIMO plants requires a tuned controller to meet a broad set of tuning specifications that entails to a many-objective problem definition. It can be interesting to study large-scale MIMO systems (e.g., redundant robotic systems) to observe the behavior of the available meta-heuristics.

- The low-level dynamic elements (e.g., actuators, power electronics elements, and sensing devices) are, in most cases, less relevant in the plant modeling. Including them in the controller tuning process can improve the achieved performance in a real scenario.

- Among the nonlinear controllers, the fuzzy ones are presented in several research items. Works that include the multi-objective optimization of other artificial intelligence techniques (e.g., neural networks [190]) are not yet developed and can be addressed in the future.
- Given a controller structure and a plant, the controller tuning process can find suitable control parameters in all the reviewed works. Nevertheless, this controller structure has always been established *a priori*. By using combinatorial optimization elements, the controller structure could also be selected from a finite set of alternatives (e.g., PID-like, sliding mode, and model predictive). The above could reveal information about which structure is better to control a given dynamic system, especially when the last is subject to uncertainties or disturbances.

Acknowledgements

The first author acknowledges the support from the Consejo Nacional de Ciencia y Tecnología (CONACyT) through the scholarship to pursue his graduate studies at CIDETEC-IPN. The second author acknowledges the support from the CONACyT through project No. 220522. The third and fourth authors acknowledge the support of the Secretaría de Investigación y Posgrado (SIP) through project No. SIP-20200150.

References

[1] N. Minato, R. Morimoto, Dynamically interdependent business model for airline-airport coexistence, Journal of Air Transport Management 64 (2017) 161 – 172, business model innovation in air transport management. `doi:10.1016/j.jairtraman.2016.08.002`.

[2] A. Oroji, M. Omar, S. Yarahmadian, An ǐto stochastic differential equations model for the dynamics of the mcf-7 breast cancer cell line treated by radiotherapy, Journal of Theoretical Biology 407 (2016) 128 – 137. `doi:10.1016/j.jtbi.2016.07.035`.

[3] Y. Wang, C. Wu, L. Yu, Y. Mei, Dynamics of a rolling robot of closed five-arc-shaped-bar linkage, Mechanism and Machine Theory 121 (2018) 75 – 91. `doi:10.1016/j.mechmachtheory.2017.10.010`.

[4] N. Pachauri, A. Rani, V. Singh, Bioreactor temperature control using modified fractional order imc-pid for ethanol production, Chemical Engineering Research and Design 122 (2017) 97 – 112. `doi:10.1016/j.cherd.2017.03.031`.

[5] P. Prommee, K. Angkeaw, High performance electronically tunable log-domain current-mode pid controller, Microelectronics Journal 72 (2018) 126 – 137. `doi:10.1016/j.mejo.2017.09.008`.

[6] L. Zhu, Y. Wang, Rumor spreading model with noise interference in complex social networks, Physica A: Statistical Mechanics and its Applications 469 (2017) 750 – 760. doi:10.1016/j.physa.2016.11.119.

[7] J. G. Ziegler, N. B. Nichols, Optimum settings for automatic controllers, Trans. ASME 64 (1942) 759–768. doi:10.1115/1.2899060.

[8] K. Astrom, T. Hagglund, PID Controllers: Theory, Design and Tuning, Isa, 1995.

[9] K. J. Astrom, T. Hagglund, Advanced PID Control, ISA Press, 2006.

[10] R. Kelly, V. Santibáñez, A. Loria, Control of robot manipulators in joint space, Springer, 2006.

[11] M. G. Villarreal-Cervantes, J. Alvarez-Gallegos, Off-line PID control tuning for a planar parallel robot using DE variants, Expert Syst. Appl. 64 (C) (2016) 444–454. doi:10.1016/j.eswa.2016.08.013.

[12] R. Kumar, S. Singla, V. Chopra, Comparison among some well known control schemes with different tuning methods, Journal of Applied Research and Technology 13 (3) (2015) 409 – 415. doi:10.1016/j.jart.2015.07.007.

[13] W. K. Ho, O. P. Gan, E. B. Tay, E. L. Ang, Performance and gain and phase margins of well-known pid tuning formulas, IEEE Transactions on Control Systems Technology 4 (4) (1996) 473–477. doi:10.1109/87.508897.

[14] D. S. Bernstein, What makes some control problems hard?, IEEE Control Systems 22 (4) (2002) 8–19. doi:10.1109/MCS.2002.1021640.

[15] N. Prabhu, M. D. Anand, S. Satish, C. K. Akhil, Dynamic modeling of scorbot-er vu plus industrial robot manipulator using labview, Indian Journal of Science and Technology 9 (13). doi:10.17485/ijst/2016/v9i13/90581.

[16] H. Sira-Ramírez, A. Luviano-Juárez, J. Cortés-Romero, Robust input–output sliding mode control of the buck converter, Control Engineering Practice 21 (5) (2013) 671 – 678. doi:10.1016/j.conengprac.2012.03.008.

[17] H. Fasih, S. Tavakoli, J. Sadeghi, H. Torabi, Kalman filter-based centralized controller design for non-square multi-input multi-output processes, Chemical Engineering Research and Design 132 (2018) 187 – 198. doi:10.1016/j.cherd.2018.01.008.

[18] S. Boyd, C. Baratt, S. Norman, Linear controller design: limits of performance via convex optimization, Proceedings of the IEEE 78 (3) (1990) 529–574. doi:10.1109/5.52229.

[19] A. E. Ruano, S. S. Ge, T. M. Guerra, F. L. Lewis, J. C. Principe, M. Colnarič, Computational intelligence in control, Annual Reviews in Control 38 (2) (2014) 233 – 242. doi:10.1016/j.arcontrol.2014.09.006.

[20] A. E. Ruano (Ed.), Intelligent Control Systems using Computational Intelligence Techniques, Control, Robotics & Sensors, Institution of Engineering and Technology, 2005. doi:10.1049/PBCE070E.

[21] R. R. Bulatović, S. R. Dordević, V. S. Dordević, Cuckoo search algorithm: A metaheuristic approach to solving the problem of optimum synthesis of a six-bar double dwell linkage, Mechanism and Machine Theory 61 (2013) 1 – 13. doi:10.1016/j.mechmachtheory.2012.10.010.

[22] S. Jovic, B. Nedeljkovic, Z. Golubovic, N. Kostic, Evolutionary algorithm for reference evapotranspiration analysis, Computers and Electronics in Agriculture 150 (2018) 1 – 4. doi:10.1016/j.compag.2018.04.003.

[23] M. Deveci, N. Ç. Demirel, Evolutionary algorithms for solving the airline crew pairing problem, Computers & Industrial Engineering 115 (2018) 389 – 406. doi:10.1016/j.cie.2017.11.022.

[24] E. Talbi, Metaheuristics: From Design to Implementation, Wiley Publishing, 2009.

[25] P. Fleming, R. Purshouse, Evolutionary algorithms in control systems engineering: a survey, Control Engineering Practice 10 (11) (2002) 1223 – 1241. doi:10.1016/S0967-0661(02)00081-3.

[26] G. Reynoso-Meza, X. Blasco, J. Sanchis, M. Martínez, Controller tuning using evolutionary multi-objective optimisation: Current trends and applications, Control Engineering Practice 28 (2014) 58 – 73. doi:10.1016/j.conengprac.2014.03.003.

[27] D. S. Pires, G. L. d. O. Serra, A new methodology for model based robust fuzzy digital pid controller design, in: 2012 Third Global Congress on Intelligent Systems, 2012, pp. 264–267. doi:10.1109/GCIS.2012.109.

[28] M. A. Martínez, J. Sanchis, X. Blasco, Multiobjective controller design handling human preferences, Engineering Applications of Artificial Intelligence 19 (8) (2006) 927 – 938. doi:10.1016/j.engappai.2006.01.018.

[29] I. Chiha, N. Liouane, P. Borne, Tuning PID controller using multiobjective ant colony optimization, pplied Computational Intelligence and Soft Computing 2012. doi:10.1155/2012/536326.

[30] C. J. Wu, T. L. Lee, Y. Y. Fu, L. C. Lai, Auto-tuning fuzzy pid control of a pendubot system, in: 2007 IEEE International Conference on Mechatronics, 2007, pp. 1–6. doi:10.1109/ICMECH.2007.4280045.

[31] M. Hussain, V. Apama, U. S. Banu, S. K. Lakshmanaprabu, Automation of two interacting conical tank process with multiobjective optimization of fractional pid controller with a metaheuristic optimizer, the cuckoo search algorithm, in: 2017 Trends in Industrial Measurement and Automation (TIMA), 2017, pp. 1–6. doi:10.1109/TIMA.2017.8064809.

[32] M. A. Sahib, B. S. Ahmed, A new multiobjective performance criterion used in pid tuning optimization algorithms, Journal of Advanced Research 7 (1) (2016) 125 – 134. doi:10.1016/j.jare.2015.03.004.

[33] Z. Wang, Q. Su, X. Luo, A novel htd-cs based pid controller tuning method for time delay continuous systems with multi-objective and multi-constraint optimization, Chemical Engineering Research and Design 115 (2016) 98 – 106. doi:10.1016/j.cherd.2016.09.025.

[34] M. Singh, R. Patel, D. Neema, Robust tuning of excitation controller for stability enhancement using multi-objective metaheuristic firefly algorithm, Swarm and Evolutionary Computation (2018) –doi:10.1016/j.swevo.2018.01.010.

[35] O. Serrano-Pérez, M. G. Villarreal-Cervantes, J. C. González-Robles, A. Rodríguez-Molina, Meta-heuristic algorithms for the control tuning of omnidirectional mobile robots, Engineering Optimization 0 (0) (2019) 1–18. doi:10.1080/0305215X.2019.1585834.

[36] C. A. C. Coello, G. B. Lamont, D. A. V. Veldhuizen, Evolutionary Algorithms for Solving Multi-Objective Problems (Genetic and Evolutionary Computation), Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006. doi:10.1007/978-0-387-36797-2.

[37] S. Ç. Onar, B. Öztayşi, C. Kahraman, S. Yanık, Ö. Şenvar, A Literature Survey on Metaheuristics in Production Systems, Springer International Publishing, Cham, 2016, pp. 1–24. doi:10.1007/978-3-319-23350-5_1.

[38] S. Jiang, Y. S. Ong, J. Zhang, L. Feng, Consistencies and contradictions of performance metrics in multiobjective optimization, IEEE Transactions on Cybernetics 44 (12) (2014) 2391–2404. doi:10.1109/TCYB.2014.2307319.

[39] F. Mallor-Gímenez, R. Blanco, C. Azcárate, Combining linear programming and multiobjective evolutionary computation for solving a type of stochastic knapsack problem, in: S. Obayashi, K. Deb, C. Poloni, T. Hiroyasu, T. Murata (Eds.), Evolutionary Multi-Criterion Optimization, Springer Berlin Heidelberg, Berlin, Heidelberg, 2007, pp. 531–545. doi:10.1007/978-3-540-70928-2_41.

[40] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, V. G. da Fonseca, Performance assessment of multiobjective optimizers: an analysis and review, IEEE Transactions on Evolutionary Computation 7 (2) (2003) 117–132. doi:10.1109/TEVC.2003.810758.

[41] J. Pierezan, H. H. V. Ayala, L. F. da Cruz, R. Z. Freire, L. dos S. Coelho, Improved multiobjective particle swarm optimization for designing pid controllers applied to robotic manipulator, in: 2014 IEEE Symposium on Computational Intelligence in Control and Automation (CICA), 2014, pp. 1–8. doi:10.1109/CICA.2014.7013255.

[42] A. Shenfield, P. J. Fleming, Multi-objective evolutionary design of robust controllers on the grid, IFAC Proceedings Volumes 44 (1) (2011) 14711 – 14716, 18th IFAC World Congress. doi:10.3182/20110828-6-IT-1002.01384.

[43] A. Zolfagharian, A. Noshadi, M. Z. M. Zain, A. R. A. Bakar, Practical multi-objective controller for preventing noise and vibration in an automobile wiper system, Swarm and Evolutionary Computation 8 (2013) 54 – 68. doi:10.1016/j.swevo.2012.08.004.

[44] R. Madiouni, S. Bouallègue, J. Haggège, P. Siarry, Robust RST control design based on multi-objective particle swarm optimization approach, International Journal of Control, Automation and Systems 14 (6) (2016) 1607–1617. doi:10.1007/s12555-015-0173-1.

[45] G. Sánchez, M. Villasana, M. Strefezza, Solving multi-objective linear control design problems using genetic algorithms, IFAC Proceedings Volumes 41 (2) (2008) 12324 – 12329, 17th IFAC World Congress. doi:10.3182/20080706-5-KR-1001.02086.

[46] M. Mahmoodabadi, M. Taherkhorsandi, A. Bagheri, Optimal robust sliding mode tracking control of a biped robot based on ingenious multi-objective pso, Neurocomputing 124 (2014) 194 – 209. doi:10.1016/j.neucom.2013.07.009.

[47] G. Reynoso-Meza, J. Sanchis, X. Blasco, J. M. Herrero, Multiobjective evolutionary algorithms for multivariable pi controller design, Expert Systems with Applications 39 (9) (2012) 7895 – 7907. doi:10.1016/j.eswa.2012.01.111.

[48] J. M. Bogoya, A. Vargas, O. Schütze, The averaged hausdorff distances in multi-objective optimization: A review, Mathematics 7 (10) (2019) 894. doi:10.3390/math7100894.

[49] C. A. C. Coello, N. C. Cortés, Solving multiobjective optimization problems using an artificial immune system, Genetic programming and evolvable machines 6 (2) (2005) 163–190. doi:10.1007/s10710-005-6164-x.

[50] N. Riquelme, C. Von Lücken, B. Baran, Performance metrics in multi-objective optimization, in: 2015 Latin American Computing Conference (CLEI), 2015, pp. 1–11. doi:10.1109/CLEI.2015.7360024.

[51] D. W. Corne, J. D. Knowles, No free lunch and free leftovers theorems for multiobjective optimisation problems, in: Proceedings of the 2Nd International Conference on Evolutionary Multi-criterion Optimization, EMO'03, Springer-Verlag, Berlin, Heidelberg, 2003, pp. 327–341. doi:10.1007/3-540-36970-8_23.

[52] D. E. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning, 1st Edition, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1989.

[53] R. Storn, K. Price, Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces, Journal of Global Optimization 11 (4) (1997) 41–359. doi:10.1023/A:1008202821328.

[54] J. Kennedy, R. Eberhart, Particle swarm optimization, in: Proceedings of IEEE International Conference on Neural Networks (1995), 1995, p. 1942–1948. doi:10.1109/ICNN.1995.488968.

[55] X.-S. Yang, A New Metaheuristic Bat-Inspired Algorithm, Springer Berlin Heidelberg, Berlin, Heidelberg, 2010, pp. 65–74. doi:10.1007/978-3-642-12538-6_6.

[56] X.-S. Yang, Firefly algorithms for multimodal optimization, in: O. Watanabe, T. Zeugmann (Eds.), Stochastic Algorithms: Foundations and Applications, Springer Berlin Heidelberg, Berlin, Heidelberg, 2009, pp. 169–178. doi:10.1007/978-3-642-04944-6_14.

[57] X. S. Yang, S. Deb, Cuckoo search via lévy flights, in: 2009 World Congress on Nature Biologically Inspired Computing (NaBIC), 2009, pp. 210–214. doi:10.1109/NABIC.2009.5393690.

[58] Z. W. Geem, J. H. Kim, G. Loganathan, A new heuristic optimization algorithm: Harmony search, SIMULATION 76 (2) (2001) 60–68. doi:10.1177/003754970107600201.

[59] C. M. Fonseca, P. J. Fleming, Genetic algorithms for multiobjective optimization: Formulationdiscussion and generalization, in: Proceedings of the 5th International Conference on Genetic Algorithms, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993, pp. 416–423.

[60] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Transactions on Evolutionary Computation 6 (2) (2002) 182–197. doi:10.1109/4235.996017.

[61] C. A. C. Coello, M. S. Lechuga, Mopso: a proposal for multiple objective particle swarm optimization, in: Evolutionary Computation, 2002. CEC '02. Proceedings of the 2002 Congress on, Vol. 2, 2002, pp. 1051–1056. doi:10.1109/CEC.2002.1004388.

[62] E. Mezura-Montes, M. Reyes-Sierra, C. A. C. Coello, Multi-objective Optimization Using Differential Evolution: A Survey of the State-of-the-Art, Springer Berlin Heidelberg, Berlin, Heidelberg, 2008, pp. 173–196. doi:10.1007/978-3-540-68830-3_7.

[63] Q. Zhang, H. Li, Moea/d: A multiobjective evolutionary algorithm based on decomposition, IEEE Transactions on Evolutionary Computation 11 (6) (2007) 712–731. doi:10.1109/TEVC.2007.892759.

[64] J. Bader, E. Zitzler, Hype: An algorithm for fast hypervolume-based many-objective optimization, Evolutionary Computation 19 (1) (2011) 45–76. doi:10.1162/EVCO_a_00009.

[65] K. Deb, H. 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) (2014) 577–601. doi:10.1109/TEVC.2013.2281535.

[66] K. Deb, S. Gupta, Understanding knee points in bicriteria problems and their implications as preferred solution principles, Engineering optimization 43 (11) (2011) 1175–1204. doi:10.1080/0305215X.2010.548863.

[67] E. E. Bischoff, A posteriori trade-off analysis in reference point approaches, in: Interactive Decision Analysis, Springer, 1984, pp. 139–145. doi:10.1007/978-3-662-00184-4_16.

[68] R. C. Purshouse, K. Deb, M. M. Mansor, S. Mostaghim, R. Wang, A review of hybrid evolutionary multiple criteria decision making methods, in: 2014 IEEE congress on evolutionary computation (CEC), IEEE, 2014, pp. 1147–1154. doi:10.1109/CEC.2014.6900368.

[69] X. Blasco, J. Herrero, J. Sanchis, M. Martínez, A new graphical visualization of n-dimensional pareto front for decision-making in multiobjective optimization, Information Sciences 178 (20) (2008) 3908 – 3924, special Issue on Industrial Applications of Neural Networks. doi:10.1016/j.ins.2008.06.010.

[70] T.-F. Liang, Distribution planning decisions using interactive fuzzy multi-objective linear programming, Fuzzy Sets and Systems 157 (10) (2006) 1303 – 1316. doi:10.1016/j.fss.2006.01.014.

[71] J. Chen, S. Lin, An interactive neural network-based approach for solving multiple criteria decision-making problems, Decision Support Systems 36 (2) (2003) 137 – 146. doi:10.1016/S0167-9236(02)00141-0.

[72] A. P. Wierzbicki, A mathematical basis for satisficing decision making, Mathematical modelling 3 (5) (1982) 391–405. doi:10.1016/0270-0255(82)90038-0.

[73] F. Hassanzadeh, H. Nemati, M. Sun, Robust optimization for interactive multiobjective programming with imprecise information applied to r&d project portfolio selection, European Journal of Operational Research 238 (1) (2014) 41–53. doi:10.1016/j.ejor.2014.03.023.

[74] A. Castelletti, A. Lotov, R. Soncini-Sessa, Visualization-based multiobjective improvement of environmental decision-making using linearization of response surfaces, Environmental Modelling & Software 25 (12) (2010) 1552 – 1564. doi:10.1016/j.envsoft.2010.05.011.

[75] G. Reynoso-Meza, J. Sanchis, X. Blasco, M. Martínez, Preference driven multi-objective optimization design procedure for industrial controller tuning, Information Sciences 339 (2016) 108 – 131. doi:10.1016/j.ins.2015.12.002.

[76] O. Lequin, M. Gevers, L. Triest, Optimizing the settling time with iterative feedback tuning, IFAC Proceedings Volumes 32 (2) (1999) 4659–4663. doi:10.1016/S1474-6670(17)56794-4.

[77] R. A. Freeman, M. Krstić, P. Kokotović, Robustness of adaptive nonlinear control to bounded uncertainties, Automatica 34 (10) (1998) 1227–1230. doi:10.1016/S0005-1098(98)00070-3.

[78] C. C. Hang, K. J. Åström, W. K. Ho, Refinements of the ziegler–nichols tuning formula, in: IEE Proceedings D (Control Theory and Applications), Vol. 138, IET, 1991, pp. 111–118. doi:10.1049/ip-d.1991.0015.

[79] K. Ogata, System dynamics, Vol. 3, Prentice Hall Upper Saddle River, NJ, 1998.

[80] L. Ljung (Ed.), System Identification (2nd Ed.): Theory for the User, Prentice Hall PTR, Upper Saddle River, NJ, USA, 1999.

[81] S. Bennett, Development of the pid controller, IEEE Control Systems Magazine 13 (6) (1993) 58–62. doi:10.1109/37.248006.

[82] Ying-Shieh Kung, Chang-Ming Liaw, A fuzzy controller improving a linear model following controller for motor drives, IEEE Transactions on Fuzzy Systems 2 (3) (1994) 194–202. doi:10.1109/91.298448.

[83] Y. Xue, D. Li, F. Gao, Multi-objective optimization and selection for the pi control of alstom gasifier problem, Control Engineering Practice 18 (1) (2010) 67 – 76. doi:10.1016/j.conengprac.2009.09.004.

[84] R. Kotteeswaran, L. Sivakumar, Performance evaluation of optimal pi controller for alstom gasifier during coal quality variations, Journal of Process Control 24 (1) (2014) 27 – 36. doi:10.1016/j.jprocont.2013.10.006.

[85] S. Zhao, M. W. Iruthayarajan, S. Baskar, P. Suganthan, Multi-objective robust pid controller tuning using two lbests multi-objective particle swarm optimization, Information Sciences 181 (16) (2011) 3323 – 3335. doi:10.1016/j.ins.2011.04.003.

[86] V. H. Alves Ribeiro, G. R. Meza, Multi-objective pid controller tuning for an industrial gasifier, in: 2018 IEEE Congress on Evolutionary Computation (CEC), 2018, pp. 1–6. doi:10.1109/CEC.2018.8477957.

[87] L. Kumar, P. kumar, Satyajeet, D. Narang, Tuning of fractional order pi$^{\lambda}$d$^{\mu}$ controllers using evolutionary optimization for pid tuned synchronous generator excitation system, IFAC-PapersOnLine 51 (4) (2018) 859 – 864, 3rd IFAC Conference on Advances in Proportional-Integral-Derivative Control PID 2018. doi:10.1016/j.ifacol.2018.06.121.

[88] G. Reynoso-Meza, H. S. Sánchez, V. H. A. Ribeiro, Control of refrigeration systems based on vapour compression using multi-objective optimization techniques, IFAC-PapersOnLine 51 (4) (2018) 722 – 727, 3rd IFAC Conference on Advances in Proportional-Integral-Derivative Control PID 2018. doi:10.1016/j.ifacol.2018.06.189.

[89] V. Mohan, H. Chhabra, A. Rani, V. Singh, An expert 2dof fractional order fuzzy pid controller for nonlinear systems, Neural Computing and Applications (2018) 1–18doi:10.1007/s00521-017-3330-z.

[90] N. K. Yegireddy, S. Panda, T. Papinaidu, K. Yadav, Multi-objective non dominated sorting genetic algorithm-ii optimized pid controller for automatic voltage regulator systems, Journal of Intelligent & Fuzzy Systems 35 (5) (2018) 4971–4975. doi:10.3233/JIFS-169781.

[91] V. Mohan, H. Chhabra, A. Rani, V. Singh, Robust self-tuning fractional order pid controller dedicated to non-linear dynamic system, Journal of Intelligent & Fuzzy Systems 34 (3) (2018) 1467–1478. doi:10.3233/JIFS-169442.

[92] S. Das, I. Pan, S. Das, Multi-objective lqr with optimum weight selection to design fopid controllers for delayed fractional order processes, ISA Transactions 58 (2015) 35 – 49. doi:10.1016/j.isatra.2015.06.002.

[93] A. Behroozsarand, S. Shafiei, Optimal control of distillation column using non-dominated sorting genetic algorithm-ii, Journal of Loss Prevention in the Process Industries 24 (1) (2011) 25 – 33. doi:10.1016/j.jlp.2010.08.009.

[94] I. Pan, S. Das, Frequency domain design of fractional order pid controller for avr system using chaotic multi-objective optimization, International Journal of Electrical Power & Energy Systems 51 (2013) 106 – 118. doi:10.1016/j.ijepes.2013.02.021.

[95] I. Pan, S. Das, Chaotic multi-objective optimization based design of fractional order piλdμ controller in avr system, International Journal of Electrical Power & Energy Systems 43 (1) (2012) 393 – 407. doi:10.1016/j.ijepes.2012.06.034.

[96] Z. Chen, X. Yuan, B. Ji, P. Wang, H. Tian, Design of a fractional order pid controller for hydraulic turbine regulating system using chaotic non-dominated sorting genetic algorithm ii, Energy Conversion and Management 84 (2014) 390 – 404. doi:10.1016/j.enconman.2014.04.052.

[97] M. Mahdavian, S. Sudeng, N. Wattanapongsakorn, Multi-objective optimization and decision making for greenhouse climate control system considering user preference and data clustering, Cluster Computing 20 (1) (2017) 835–853. doi:10.1007/s10586-017-0772-0.

[98] J. F. Whidborne, R. S. H. Istepanian, Genetic algorithm approach to designing finite-precision controller structures, IEE Proceedings - Control Theory and Applications 148 (5) (2001) 377–382. doi:10.1049/ip-cta:20010604.

[99] H. V. H. Ayala, L. dos Santos Coelho, Tuning of pid controller based on a multiobjective genetic algorithm applied to a robotic manipulator, Expert Systems with Applications 39 (10) (2012) 8968 – 8974. doi:10.1016/j.eswa.2012.02.027.

[100] K. S. Tang, K. F. Man, G. Chen, S. Kwong, An optimal fuzzy pid controller, IEEE Transactions on Industrial Electronics 48 (4) (2001) 757–765. doi:10.1109/41.937407.

[101] R. Lagunas-Jimenez, G. Fernandez-Anaya, J. C. Martinez-garcia, Tuning of two-degrees-of-freedom pid controllers via the multiobjective genetic algorithm nsga-ii, in: Electronics, Robotics and Automotive Mechanics Conference (CERMA'06), Vol. 2, 2006, pp. 145–150. doi:10.1109/CERMA.2006.94.

[102] S. Tavakoli, I. Griffin, P. J. Fleming, Multi-objective optimization approach to the pi tuning problem, in: 2007 IEEE Congress on Evolutionary Computation, 2007, pp. 3165–3171. doi:10.1109/CEC.2007.4424876.

[103] A. Hajiloo, W. F. Xie, Fuzzy fractional-order pid controller design using multi-objective optimization, in: 2013 Joint IFSA World Congress and NAFIPS Annual Meeting (IFSA/NAFIPS), 2013, pp. 1445–1450. doi:10.1109/IFSA-NAFIPS.2013.6608614.

[104] N. K. Yegireddy, S. Panda, Design and performance analysis of pid controller for an avr system using multi-objective non-dominated shorting genetic algorithm-ii, in: 2014 International Conference on Smart Electric Grid (ISEG), 2014, pp. 1–7. doi:10.1109/ISEG.2014.7005600.

[105] A. Noshadi, J. Shi, W. Lee, A. Kalam, Pid-type fuzzy logic controller for active magnetic bearing system, in: IECON 2014 - 40th Annual Conference of the IEEE Industrial Electronics Society, 2014, pp. 241–247. doi:10.1109/IECON.2014.7048506.

[106] P. Kumar, S. Narayan, Optimal design of robust fopid for the aircraft pitch control system using multi-objective ga, in: 2016 IEEE Students' Conference on Electrical, Electronics and Computer Science (SCEECS), 2016, pp. 1–6. doi:10.1109/SCEECS.2016.7509267.

[107] R. Gutiérrez-Urquídez, G. Valencia-Palomo, O. Rodríguez-Elias, L. Trujillo, Systematic selection of tuning parameters for efficient predictive controllers using a multiobjective evolutionary algorithm, Applied Soft Computing 31 (2015) 326 – 338. doi:10.1016/j.asoc.2015.02.033.

[108] M. Mahmoodabadi, H. Jahanshahi, Multi-objective optimized fuzzy-pid controllers for fourth order nonlinear systems, Engineering Science and Technology, an International Journal 19 (2) (2016) 1084 – 1098. doi:10.1016/j.jestch.2016.01.010.

[109] J. H. Crews, M. G. Mattson, G. D. Buckner, Multi-objective control optimization for semi-active vehicle suspensions, Journal of Sound and Vibration 330 (23) (2011) 5502 – 5516. doi:10.1016/j.jsv.2011.05.036.

[110] A. Herreros, E. Baeyens, J. R. Perán, Design of pid-type controllers using multiobjective genetic algorithms, ISA Transactions 41 (4) (2002) 457 – 472. doi:10.1016/S0019-0578(07)60102-5.

[111] A. Herreros, E. Baeyens, J. R. Perán, Mrcd: a genetic algorithm for multiobjective robust control design, Engineering Applications of Artificial Intelligence 15 (3) (2002) 285 – 301. doi:10.1016/S0952-1976(02)00036-2.

[112] A. Hajiloo, N. Nariman-zadeh, A. Moeini, Pareto optimal robust design of fractional-order pid controllers for systems with probabilistic uncertainties, Mechatronics 22 (6) (2012) 788 – 801, special Issue on Intelligent Mechatronics (LSMS2010 and ICSEE2010). doi:10.1016/j.mechatronics.2012.04.003.

[113] S. Panda, Multi-objective pid controller tuning for a facts-based damping stabilizer using non-dominated sorting genetic algorithm-ii, International Journal of Electrical Power and Energy Systems 33 (7) (2011) 1296 – 1308. doi:10.1016/j.ijepes.2011.06.002.

[114] P. Stewart, D. Stone, P. Fleming, Design of robust fuzzy-logic control systems by multi-objective evolutionary methods with hardware in the loop, Engineering Applications of Artificial Intelligence 17 (3) (2004) 275 – 284. doi:10.1016/j.engappai.2004.03.003.

[115] S. Algoul, M. S. Alam, M. A. Hossain, M. A. A. Majumder, Multi-objective optimal chemotherapy control model for cancer treatment, Medical & Biological Engineering & Computing 49 (1) (2011) 51–65. `doi:10.1007/s11517-010-0678-y`.

[116] Z. Chen, Y. Yuan, X. Yuan, Y. Huang, X. Li, W. Li, Application of multi-objective controller to optimal tuning of pid gains for a hydraulic turbine regulating system using adaptive grid particle swam optimization, ISA Transactions 56 (2015) 173 – 187. `doi:10.1016/j.isatra.2014.11.003`.

[117] M. Mahmoodabadi, A. Bagheri, S. A. Mostaghim, M. Bisheban, Simulation of stability using java application for pareto design of controllers based on a new multi-objective particle swarm optimization, Mathematical and Computer Modelling 54 (5) (2011) 1584 – 1607. `doi:10.1016/j.mcm.2011.04.032`.

[118] M. J. Mahmoodabadi, M. Taherkhorsandi, A. Bagheri, Pareto design of state feedback tracking control of a biped robot via multiobjective pso in comparison with sigma method and genetic algorithms: Modified nsgaii and matlab's toolbox, The Scientific World Journal`doi:10.1155/2014/303101`.

[119] H. Freire, P. B. Moura Oliveira, E. J. Solteiro Pires, From single to many-objective pid controller design using particle swarm optimization, International Journal of Control, Automation and Systems 15 (2) (2017) 918–932. `doi:10.1007/s12555-015-0271-0`.

[120] Y. Tian, Q. Wang, Y. Wang, Q. Jin, A novel design method of multi-objective robust pid controller for industrial process, in: 2014 9th IEEE Conference on Industrial Electronics and Applications, 2014, pp. 242–246. `doi:10.1109/ICIEA.2014.6931166`.

[121] H. P. Ren, X. Guo, Optimization controller design of caczvs three phase pfc converter using particle swarm optimization, in: IECON 2014 - 40th Annual Conference of the IEEE Industrial Electronics Society, 2014, pp. 1665–1671. `doi:10.1109/IECON.2014.7048726`.

[122] H. Fu, L. Pan, Y. Xue, L. Sun, D. Li, K. Lee, Z. Wu, T. He, S. Zheng, Cascaded pi controller tuning for power plant superheated steam temperature based on multi-objective optimization, IFAC-PapersOnLine 50 (1) (2017) 3227 – 3231, 20th IFAC World Congress. `doi:10.1016/j.ifacol.2017.08.447`.

[123] A. Farokhi, M. Mahmoodabadi, Optimal fuzzy inverse dynamics control of a parallelogram mechanism based on a new multi-objective pso, Cogent Engineering 5 (1) (2018) 1443675. `doi:10.1080/23311916.2018.1443675`.

[124] I. B. Tijani, R. Akmeliawati, A. G. A. Muthalif, A. Legowo, Optimization of pid controller for flexible link system using a pareto-based multi-objective differential (pmode) evolution, in: 2011 4th International Conference on Mechatronics (ICOM), 2011, pp. 1–6. doi:10.1109/ICOM.2011.5937190.

[125] G. Reynoso-Meza, J. Sanchis, X. Blasco, J. M. Herrero, Handling control engineer preferences: Getting the most of pi controllers, in: ETFA2011, 2011, pp. 1–8. doi:10.1109/ETFA.2011.6059074.

[126] J. V. Carrau, G. Reynoso-Meza, S. García-Nieto, X. 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 (2017) 52 – 66. doi:10.1016/j.engappai.2017.05.005.

[127] G. Reynoso-Meza, J. Sanchis, X. Blasco, R. Z. Freire, Evolutionary multi-objective optimisation with preferences for multivariable pi controller tuning, Expert Systems with Applications 51 (2016) 120 – 133. doi:10.1016/j.eswa.2015.11.028.

[128] G. Reynoso-Meza, J. Carrillo-Ahumada, Y. Boada, J. Picó, Pid controller tuning for unstable processes using a multi-objective optimisation design procedure, IFAC-PapersOnLine 49 (7) (2016) 284 – 289, 11th IFAC Symposium on Dynamics and Control of Process SystemsIncluding Biosystems DYCOPS-CAB 2016. doi:10.1016/j.ifacol.2016.07.287.

[129] M. Marinaki, Y. Marinakis, G. E. Stavroulakis, Fuzzy control optimized by a multi-objective differential evolution algorithm for vibration suppression of smart structures, Computers and Structures 147 (2015) 126 – 137, cIVIL-COMP. doi:10.1016/j.compstruc.2014.09.018.

[130] M. G. Villarreal-Cervantes, A. Rodríguez-Molina, C. V. García-Mendoza, O. Peñaloza-Mejía, G. Sepúlveda-Cervantes, Multi-objective on-line optimization approach for the dc motor controller tuning using differential evolution, IEEE Access 5 (2017) 20393–20407. doi:10.1109/ACCESS.2017.2757959.

[131] M. H. Hung, L. S. Shu, S. J. Ho, S. F. Hwang, S. Y. Ho, A novel intelligent multiobjective simulated annealing algorithm for designing robust pid controllers, IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans 38 (2) (2008) 319–330. doi:10.1109/TSMCA.2007.914793.

[132] H. Halsall-Whitney, J. Thibault, Multi-objective optimization for chemical processes and controller design: Approximating and classifying the pareto domain, Computers & Chemical Engineering 30 (6) (2006) 1155 – 1168. doi:10.1016/j.compchemeng.2006.02.010.

[133] G. Zeng, J. Chen, Y. Dai, L. Li, C. Zheng, M. Chen, Design of fractional order pid controller for automatic regulator voltage system based on multi-objective extremal optimization, Neurocomputing 160 (2015) 173 – 184. doi:10.1016/j.neucom.2015.02.051.

[134] X. Zhou, J. Zhou, C. Yang, W. Gui, Set-point tracking and multi-objective optimization-based pid control for the goethite process, IEEE Access 6 (2018) 36683–36698. doi:10.1109/ACCESS.2018.2847641.

[135] O. T. Altinoz, Optimal Controller Parameter Tuning from Multi/Many-objective Optimization Algorithms, Springer International Publishing, Cham, 2019, pp. 51–74. doi:10.1007/978-3-030-25446-9_3.

[136] H. Chhabra, V. Mohan, A. Rani, V. Singh, Multi-objective cuckoo search algorithm-based 2-dof fopd controller for robotic manipulator, in: B. S. Rawat, A. Trivedi, S. Manhas, V. Karwal (Eds.), Advances in Signal Processing and Communication, Springer Singapore, Singapore, 2019, pp. 345–352. doi:10.1007/978-981-13-2553-3_33.

[137] C. Zhang, T. Peng, C. Li, W. Fu, X. Xia, X. Xue, Multiobjective optimization of a fractional-order pid controller for pumped turbine governing system using an improved nsga-iii algorithm under multiworking conditions, Complexity 2019. doi:10.1155/2019/5826873.

[138] B. DeBoon, B. Kent, M. Lacki, S. Nokleby, C. Rossa, Multi-objective gain optimizer for an active disturbance rejection controller, in: 2019 IEEE Global Conference on Signal and Information Processing (GlobalSIP), 2019, pp. 1–5. doi:10.1109/GlobalSIP45357.2019.8969275.

[139] X. Zhou, X. Zhang, Multi-objective-optimization-based control parameters auto-tuning for aerial manipulators, International Journal of Advanced Robotic Systems 16 (1) (2019) 1729881419828071. doi:10.1177/1729881419828071.

[140] K. Sanprasit, P. Artrit, Optimal comparison using mowoa and mogwo for pid tuning of dc servo motor, Journal of Automation and Control Engineering 7 (1). doi:10.18178/joace.7.1.45-49.

[141] M. J. Mahmoodabadi, M. Taherkhorsandi, Intelligent control of biped robots: optimal fuzzy tracking control via multi-objective particle swarm optimization and genetic algorithms, AUT Journal of Mechanical Engineeringdoi:10.22060/AJME.2019.16171.5808.

[142] M. J. Mahmoodabadi, M. M. Shahangian, A new multi-objective artificial bee colony algorithm for optimal adaptive robust controller design, IETE Journal of Research (2019) 1–14doi:10.1080/03772063.2019.1644211.

[143] K. Quresh, S. Rahnamayan, Y. He, R. Liscano, Enhancing lqr controller using optimized real-time system by gde3 and nsga-ii algorithms and comparing with conventional method, in: 2019 IEEE Congress on Evolutionary Computation (CEC), 2019, pp. 2074–2081. doi:10.1109/CEC.2019.8789894.

[144] A. Rodríguez-Molina, M. G. Villarreal-Cervantes, E. Mezura-Montes, M. Aldape-Pérez, Adaptive controller tuning method based on online multiobjective optimization: A case study of the four-bar mechanism, IEEE Transactions on Cybernetics (2019) 1–14doi:10.1109/TCYB.2019.2903491.

[145] A. Susperregui, J. M. Herrero, M. I. Martinez, G. Tapia-Otaegui, X. Blasco, Multi-objective optimisation-based tuning of two second-order sliding-mode controller variants for dfigs connected to non-ideal grid voltage, Energies 12 (19) (2019) 3782. doi:10.3390/en12193782.

[146] Y. Sardahi, J.-Q. Sun, C. Hernández, O. Schütze, Many-objective optimal and robust design of proportional-integral-derivative controls with a state observer, Journal of Dynamic Systems, Measurement, and Control 139 (2). doi:10.1115/1.4034749.

[147] Y. Sardahi, A. Boker, Multi-objective optimal design of four-parameter PID controls, in: ASME 2018 Dynamic Systems and Control Conference, American Society of Mechanical Engineers Digital Collection, 2018. doi:10.1115/DSCC2018-8935.

[148] K. Ogata, Modern Control Engineering, 4th Edition, Prentice Hall PTR, Upper Saddle River, NJ, USA, 2001.

[149] C. Lin, P. Hajela, Genetic search strategies in large scale optimization, in: 34th Structures, Structural Dynamics and Materials Conference, 1993, p. 1585. doi:10.2514/6.1993-1585.

[150] J. J. Moré, On the performance of algorithms for large-scale bound constrained problems, Large-scale Numerical Optimization 46 (1990) 32.

[151] A. R. Rao, Process control engineering, CRC Press, 1993.

[152] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: Nsga-ii, IEEE Transactions on Evolutionary Computation 6 (2) (2002) 182–197. doi:10.1109/4235.996017.

[153] Y. Yusoff, M. S. Ngadiman, A. M. Zain, Overview of nsga-ii for optimizing machining process parameters, Procedia Engineering 15 (2011) 3978–3983. doi:10.1016/j.proeng.2011.08.745.

[154] M. Reyes-sierra, C. A. C. Coello, Multi-objective particle swarm optimizers: A survey of the state-of-the-art, International Journal of Computational Intelligence Research 2 (3) (2006) 287–308. doi:10.5019/j.ijcir.2006.68.

[155] E. Mezura-Montes, J. Velázquez-Reyes, C. A. Coello Coello, A comparative study of differential evolution variants for global optimization, in: Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation, GECCO '06, 2006. doi:10.1145/1143997.1144086.

[156] E. Zitzler, L. Thiele, An evolutionary algorithm for multiobjective optimization: The strength pareto approach, TIK-report 43. doi:10.1.1.40.7696.

[157] E. Zitzler, M. Laumanns, L. Thiele, Spea2: Improving the strength pareto evolutionary algorithm, TIK-report 103. doi:10.3929/ethz-a-004284029.

[158] R. Vlennet, C. Fonteix, I. Marc, Multicriteria optimization using a genetic algorithm for determining a pareto set, International Journal of Systems Science 27 (2) (1996) 255–260. doi:10.1080/00207729608929211.

[159] E. Rashedi, H. Nezamabadi-Pour, S. Saryazdi, Gsa: a gravitational search algorithm, Information sciences 179 (13) (2009) 2232–2248. doi:10.1016/j.ins.2009.03.004.

[160] M.-R. Chen, Y.-Z. Lu, A novel elitist multiobjective optimization algorithm: Multiobjective extremal optimization, European Journal of Operational Research 188 (3) (2008) 637–651. doi:10.1016/j.ejor.2007.05.008.

[161] M. Wagner, F. Neumann, A fast approximation-guided evolutionary multi-objective algorithm, in: Proceedings of the 15th annual conference on Genetic and evolutionary computation, 2013, pp. 687–694. doi:10.1145/2463372.2463448.

[162] M. Laumanns, L. Thiele, E. Zitzler, An efficient, adaptive parameter variation scheme for metaheuristics based on the epsilon-constraint method, European Journal of Operational Research 169 (3) (2006) 932–942. doi:10.1016/j.ejor.2004.08.029.

[163] K. Deb, An efficient constraint handling method for genetic algorithms, Computer Methods in Applied Mechanics and Engineering 186 (2) (2000) 311 – 338. doi:10.1016/S0045-7825(99)00389-8.

[164] E. Mezura-Montes, C. A. C. Coello, Constraint-handling in nature-inspired numerical optimization: past, present and future, Swarm and Evolutionary Computation 1 (4) (2011) 173–194. doi:10.1016/j.swevo.2011.10.001.

[165] E. Mezura-Montes, Constraint-Handling in Evolutionary Optimization, 1st Edition, Springer Publishing Company, Incorporated, 2009. doi:10.1007/978-3-642-00619-7.

[166] M. Elarbi, S. Bechikh, A. Gupta, L. Ben Said, Y. Ong, A new decomposition-based nsga-ii for many-objective optimization, IEEE Transactions on Systems, Man, and Cybernetics: Systems 48 (7) (2018) 1191–1210. doi:10.1109/TSMC.2017.2654301.

[167] C. C. Coello, M. S. Lechuga, Mopso: A proposal for multiple objective particle swarm optimization, in: Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No. 02TH8600), Vol. 2, IEEE, 2002, pp. 1051–1056. doi:10.1109/CEC.2002.1004388.

[168] Q. Zhang, H. Li, Moea/d: A multiobjective evolutionary algorithm based on decomposition, IEEE Transactions on evolutionary computation 11 (6) (2007) 712–731. doi:10.1109/TEVC.2007.892759.

[169] S. Kukkonen, J. Lampinen, Gde3: The third evolution step of generalized differential evolution, in: 2005 IEEE congress on evolutionary computation, Vol. 1, IEEE, 2005, pp. 443–450. doi:10.1109/CEC.2005.1554717.

[170] R. Cheng, Y. Jin, M. Olhofer, B. Sendhoff, A reference vector guided evolutionary algorithm for many-objective optimization, IEEE Transactions on Evolutionary Computation 20 (5) (2016) 773–791. doi:10.1109/TEVC.2016.2519378.

[171] A. H. Gandomi, X.-S. Yang, A. H. Alavi, Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems, Engineering with computers 29 (1) (2013) 17–35. doi:10.1007/s00366-012-0308-4.

[172] S. Mirjalili, A. Lewis, The whale optimization algorithm, Advances in engineering software 95 (2016) 51–67. doi:10.1016/j.advengsoft.2016.01.008.

[173] S. Mirjalili, S. M. Mirjalili, A. Lewis, Grey wolf optimizer, Advances in engineering software 69 (2014) 46–61. doi:10.1016/j.advengsoft.2013.12.007.

[174] D. Karaboga, B. Basturk, A powerful and efficient algorithm for numerical function optimization: artificial bee colony (abc) algorithm, Journal of global optimization 39 (3) (2007) 459–471. doi:10.1007/s10898-007-9149-x.

[175] E. Zitzler, K. Deb, L. Thiele, Comparison of multiobjective evolutionary algorithms: Empirical results, Evolutionary Computation 8 (2) (2000) 173–195. doi:10.1162/106365600568202.

[176] A. Czarn, C. MacNish, K. Vijayan, B. Turlach, R. Gupta, Statistical exploratory analysis of genetic algorithms, IEEE Transactions on Evolutionary Computation 8 (4) (2004) 405–421. doi:10.1109/TEVC.2004.831262.

[177] J. Derrac, S. García, D. Molina, F. Herrera, A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms, Swarm and Evolutionary Computation 1 (1) (2011) 3 – 18. doi:10.1016/j.swevo.2011.02.002.

[178] C. Zhu, L. Xu, E. D. Goodman, Generalization of pareto-optimality for many-objective evolutionary optimization, IEEE Transactions on Evolutionary Computation 20 (2) (2016) 299–315. doi:10.1109/TEVC.2015.2457245.

[179] B. Li, J. Li, K. Tang, X. Yao, Many-objective evolutionary algorithms: A survey, ACM Computing Surveys (CSUR) 48 (1) (2015) 13. doi:10.1145/2792984.

[180] F. Neri, C. Cotta, Memetic algorithms and memetic computing optimization: A literature review, Swarm and Evolutionary Computation 2 (2012) 1–14. doi:10.1016/j.swevo.2011.11.003.

[181] R. Wehrens, L. M. Buydens, Evolutionary optimisation: a tutorial, TrAC Trends in Analytical Chemistry 17 (4) (1998) 193–203. doi:10.1016/S0165-9936(98)00011-9.

[182] C. A. C. Coello, Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art, Computer methods in applied mechanics and engineering 191 (11-12) (2002) 1245–1287. doi:10.1016/S0045-7825(01)00323-1.

[183] S. Mahdavi, M. E. Shiri, S. Rahnamayan, Metaheuristics in large-scale global continues optimization: A survey, Information Sciences 295 (2015) 407–428. doi:10.1016/j.ins.2014.10.042.

[184] M. Birattari, J. Kacprzyk, Tuning metaheuristics: a machine learning perspective, Vol. 197, Springer, 2009. doi:10.1007/978-3-642-00483-4.

[185] P. M. Wensing, S. Kim, J. E. Slotine, Linear matrix inequalities for physically consistent inertial parameter identification: A statistical perspective on the mass distribution, IEEE Robotics and Automation Letters 3 (1) (2018) 60–67. doi:10.1109/LRA.2017.2729659.

[186] I. D. Landau, R. Lozano, M. M'Saad, et al., Adaptive control, Vol. 51, Springer New York, 1998. doi:10.1007/978-0-85729-664-1.

[187] M.-E. Miranda-Varela, E. Mezura-Montes, Constraint-handling techniques in surrogate-assisted evolutionary optimization. an empirical study, Applied Soft Computing 73 (2018) 215 – 229. doi:10.1016/j.asoc.2018.08.016.

[188] V.-D. Cung, S. L. Martins, C. C. Ribeiro, C. Roucairol, Strategies for the parallel implementation of metaheuristics, in: Essays and surveys in metaheuristics, Springer, 2002, pp. 263–308. doi:10.1007/978-1-4615-1507-4_13.

[189] S. Rostami, F. Neri, M. Epitropakis, Progressive preference articulation for decision making in multi-objective optimisation problems, Integrated Computer-Aided Engineering 24 (4) (2017) 315–335. doi:10.3233/ICA-170547.

[190] Nisha, A. Mittal, O. P. Sharma, N. Dhyani, V. Sharma, A. Kapoor, T. K. Saxena, Tuning of pid parameters using artificial neural network, in: 2009 International Multimedia, Signal Processing and Communication Technologies, 2009, pp. 309–312. doi:10.1109/MSPCT.2009.5164237.