Chapter 2
Background on Multiobjective Optimization for Controller Tuning

Abstract In this chapter a background on multiobjective optimization and a review on multiobjective optimization design procedures within the context of control systems and the controller tuning problem are provided. Focus is given on multiobjective problems where an analysis of the Pareto front is required, in order to select the most preferable design alternative for the control problem at hand.

2.1 Definitions

A MOP, without loss of generality,[1] can be stated as follows:

$$\min_{\boldsymbol{\theta}} \boldsymbol{J}(\boldsymbol{\theta}) = [J_1(\boldsymbol{\theta}), \ldots, J_m(\boldsymbol{\theta})] \tag{2.1}$$

subject to:

$$\boldsymbol{g}(\boldsymbol{\theta}) \leq \mathbf{0} \tag{2.2}$$

$$\boldsymbol{h}(\boldsymbol{\theta}) = \mathbf{0} \tag{2.3}$$

where $\boldsymbol{\theta} \in D \subseteq \Re^n$ is defined as the decision vector in the searching space D, $\boldsymbol{J}(\boldsymbol{\theta}) \in \Lambda \subseteq \Re^m$ as the objective vector and $\boldsymbol{g}(\boldsymbol{\theta}), \boldsymbol{h}(\boldsymbol{\theta})$ as the inequality and equality constraint vectors, respectively. As remarked previously, there is no single solution to this problem because in general there is no best solution for all objectives. However, a set of solutions, the Pareto Set $\boldsymbol{\Theta}_P$, is defined, where each decision $\boldsymbol{\theta} \in \boldsymbol{\Theta}_P$ defines an objective vector $\boldsymbol{J}(\boldsymbol{\theta})$ in the Pareto Front. All solutions in the Pareto Front are said to be a set of Pareto-optimal and non-dominated solutions:

[1] A maximization problem can be converted to a minimization one. For each of the objectives to maximize, the transformation: max $J_i(\boldsymbol{\theta}) = -min(-J_i(\boldsymbol{\theta}))$ should be applied.

G. Reynoso Meza et al., *Controller Tuning with Evolutionary Multiobjective Optimization*, Intelligent Systems, Control and Automation: Science and Engineering 85, DOI 10.1007/978-3-319-41301-3_2

Definition 2.1 (*Pareto Dominance*): A decision vector $\boldsymbol{\theta}^1$ dominates another vector $\boldsymbol{\theta}^2$, denoted as $\boldsymbol{\theta}^1 \preceq \boldsymbol{\theta}^2$, if $\boldsymbol{J}(\boldsymbol{\theta}^1)$ is not worse than $\boldsymbol{J}(\boldsymbol{\theta}^2)$ in all objectives and is better in at least one objective.

$$\forall i \in A := \{1, \ldots, m\}, \boldsymbol{J}_i(\boldsymbol{\theta}^1) \leq \boldsymbol{J}_i(\boldsymbol{\theta}^2) \wedge \exists i \in A : \boldsymbol{J}_i(\boldsymbol{\theta}^1) < \boldsymbol{J}_i(\boldsymbol{\theta}^2).$$

Definition 2.2 (*Strict Dominance* [91]): A decision vector $\boldsymbol{\theta}^1$ is strictly dominated by another vector $\boldsymbol{\theta}^2$ if $\boldsymbol{J}(\boldsymbol{\theta}^1)$ is worse than $\boldsymbol{J}(\boldsymbol{\theta}^2)$ in all objectives.

Definition 2.3 (*Weak Dominance* [91]): A decision vector $\boldsymbol{\theta}^1$ weakly dominates another vector $\boldsymbol{\theta}^2$ if $\boldsymbol{J}(\boldsymbol{\theta}^1)$ is not worse than $\boldsymbol{J}(\boldsymbol{\theta}^2)$ in all objectives.

Definition 2.4 (*Pareto optimal*): A solution vector $\boldsymbol{\theta}^*$ is Pareto optimal iff

$$\nexists \boldsymbol{\theta} \in D : \boldsymbol{\theta} \preceq \boldsymbol{\theta}^*.$$

Definition 2.5 (*Pareto Set*): In a MOP, the Pareto Set $\boldsymbol{\Theta}_P$ is the set including all the Pareto optimal solutions:

$$\boldsymbol{\Theta}_P := \{\boldsymbol{\theta} \in D | \nexists \boldsymbol{\theta}' \in D : \boldsymbol{\theta}' \preceq \boldsymbol{\theta}\}.$$

Definition 2.6 (*Pareto Front*): In a MOP, the Pareto Front $\boldsymbol{J}_P$ is the set including the objective vectors of all Pareto optimal solutions:

$$\boldsymbol{J}_P := \{\boldsymbol{J}(\boldsymbol{\theta}) : \boldsymbol{\theta} \in \boldsymbol{\Theta}_P\}.$$

For example, in Fig. 2.1, five different solutions $\boldsymbol{\theta}^1 \ldots \boldsymbol{\theta}^5$ and their corresponding objective vectors $\boldsymbol{J}(\boldsymbol{\theta}^1) \ldots \boldsymbol{J}(\boldsymbol{\theta}^5)$ are calculated to approximate the Pareto Set $\boldsymbol{\Theta}_P$

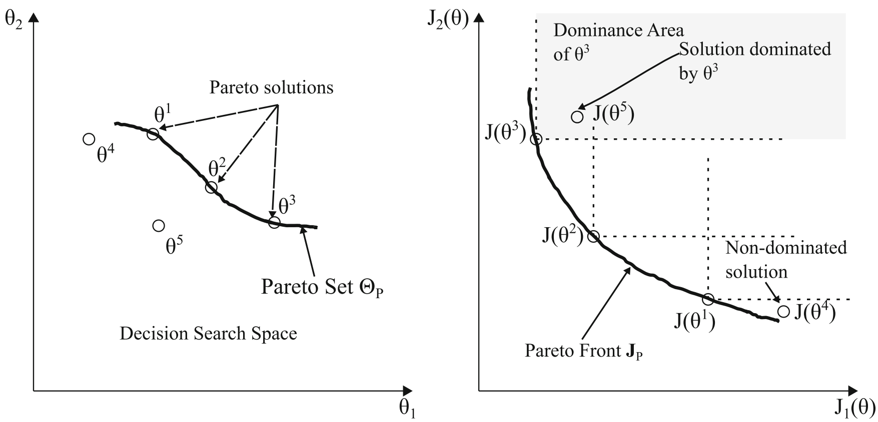

Fig. 2.1 Pareto optimality and dominance definitions. Pareto set and front in a bidimensional case ($m = 2$)

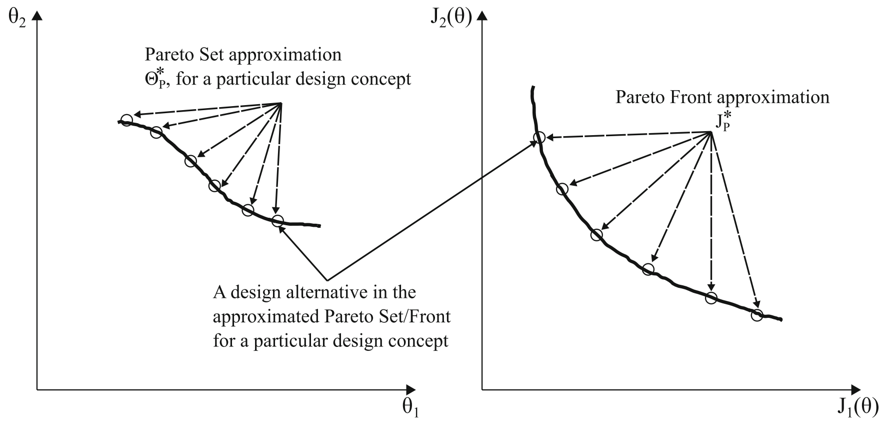

Fig. 2.2 Design concept and design alternative definitions

and Pareto Front $\boldsymbol{J}_P$ (bold lines). Solutions $\boldsymbol{\theta}^1 \ldots \boldsymbol{\theta}^4$ are non-dominated solutions, since there are no better solution vectors (in the calculated set) for all the objectives. Solution $\boldsymbol{\theta}^4$ is not Pareto optimal, since some solutions (not found in this case) dominate it. However, solutions $\boldsymbol{\theta}^1$, $\boldsymbol{\theta}^2$ and $\boldsymbol{\theta}^3$ are Pareto optimal, since they lie on the feasible Pareto front.

Obtaining $\boldsymbol{\Theta}_P$ is computationally infeasible, since most of the times the Pareto Front is unknown and likely it contains infinite solutions (notice that you shall only rely on approximations of the Pareto set $\boldsymbol{\Theta}_P^*$ and Front $\boldsymbol{J}_P^*$). In Fig. 2.1 the non-dominated solutions $\boldsymbol{\theta}^1 \ldots \boldsymbol{\theta}^4$ conform an approximated Pareto Set $\boldsymbol{\Theta}_P^*$ (although only $\boldsymbol{\theta}^1 \ldots \boldsymbol{\theta}^3$ are Pareto optimal) and their corresponding Pareto Front $\boldsymbol{J}_P^*$ approximation. Since $\boldsymbol{\Theta}_P$ contains all Pareto optimal solutions it will be desirable than $\boldsymbol{\Theta}_P^* \subset \boldsymbol{\Theta}_P$.

In [84], some refinement is incorporated into the Pareto Front notion to differentiate design concepts. A Pareto Front is defined given a design concept (or simply, a concept) that is, an idea about how to solve a given MOP. The design concept is built with a family of design choices (Pareto-optimal solutions) that are specific solutions in the design concept. In the leading example, the PI controller is the design concept, whereas a specific pair of proportional and integral gains is a design alternative. For example, in Fig. 2.2, the Pareto Set/Front (bold lines) for a particular design concept are approximated with a set of Pareto-optimal design alternatives (◯) (for example, a PI controller for a given MOP as a design concept). But sometimes, there are different concepts, all of which are viable for solving an MOP (for example, for a given control problem an LQR *or* a fuzzy controller can be used as alternative to PI concept). Therefore, the DM can calculate a Pareto Front approximation for each concept in order to make a comparison. Accordingly, in [84] the definition of Pareto Front and Pareto optimality were extended to a Pareto Front for a set of concepts (s-Pareto Front) where all solutions are s-Pareto optimal.

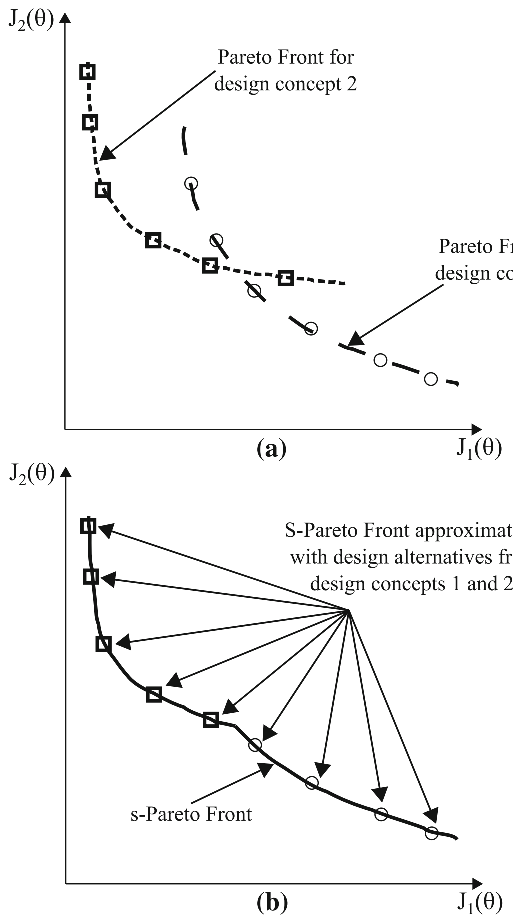

Fig. 2.3 s-pareto front definition

Definition 2.7 (*s-Pareto optimality*): Given an MOP and K design concepts, a solution vector $\boldsymbol{\theta}^1$ is s-Pareto optimal if there is no other solution $\boldsymbol{\theta}^2$ in the design concept k such that $J_i(\boldsymbol{\theta}^2) \leq J_i(\boldsymbol{\theta}^1)$ for all $i \in [1, 2, \ldots, m]$ and all concepts $k, k \in [1, \ldots, K]$; and $J_j(\boldsymbol{\theta}^2) < J_j(\boldsymbol{\theta}^1)$ for at least one $j, j \in [1, 2, \ldots, m]$ for any concept k.

Therefore, the s-Pareto Front is built joining the design alternatives of K design concepts. In Fig. 2.3 two different Pareto Front approximations for two different concepts (◯ and □ respectively) are shown (Fig. 2.3a). In Fig. 2.3b, an s-Pareto Front with both design concepts is built.

As remarked in [84], a comparison between design concepts is useful for the designer, because he/she will be able to identify the concepts strengths, weaknesses, limitations and drawbacks. It is also important to visualize such comparisons, and to have a quantitative measure to evaluate these strengths and weaknesses.

In the next section, it will be discussed how to incorporate such notions into a design procedure for multiobjective problems.

2.2 Multiobjective Optimization Design (MOOD) Procedure

It is important to perform an entire procedure [9] minding equally the decision making and optimization steps [14]. Therefore, a general framework is required to successfully incorporate this approach into any engineering design process. In Fig. 2.4 a general framework for any MOOD procedure is shown. It consists of (at least) three main steps [18, 19]: the MOP definition (measurement); the multiobjective optimization process (search); and the MCDM stage (decision making).

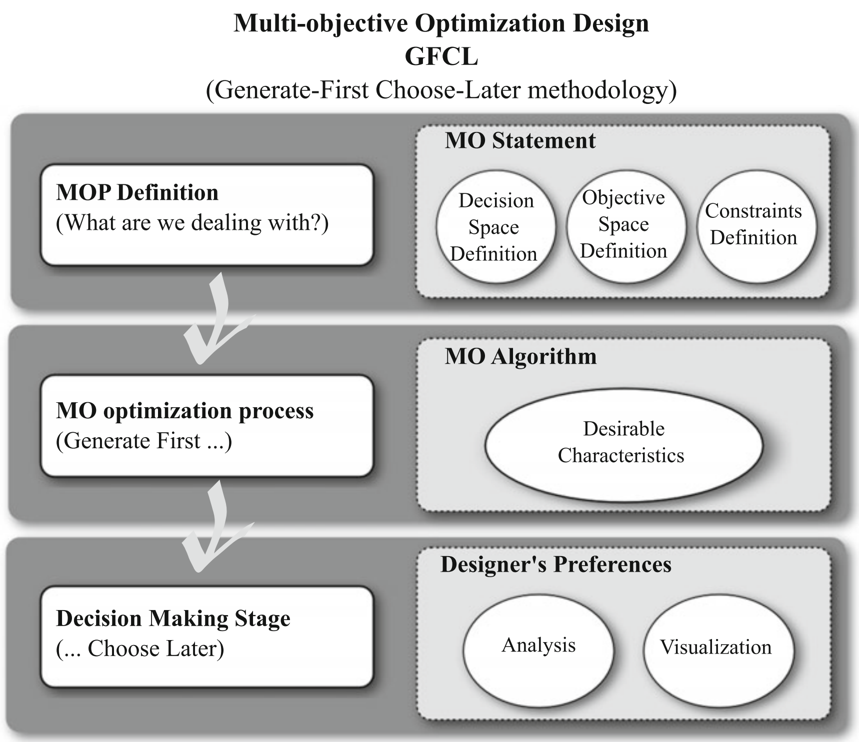

Fig. 2.4 A multiobjective optimization design (MOOD) procedure

2.2.1 Multiobjective Problem (MOP) Definition

In this stage, the design concept (how to tackle the problem at hand), the engineering requirements (which is important to optimize) and the constraints (which solutions are not practical/allowed) have to be defined. In [84] it is noted that the design concept implies the existence of a parametric model that defines the parameter values (the decision space) leading to a particular design alternative and performance (objective space). This is not a trivial task, since the problem formulation from the point of view of the designer is not that of the optimizer [45]. A lot of MOP definitions and their Pareto Front approximations have been proposed in several fields as described in [17]. Also, reviews on rule mining [123], supply chains [2, 79], energy systems [35, 38], flow shop scheduling [129], pattern recognition [21], hydrological modeling [34], water resources [107], machining [139], and portfolio management [88] can be consulted by interested readers.

The designer will search for a preferable solution at the end of the optimization process. As this book is dedicated to control system engineering, the discussed design concepts will be entirely related to this field. As a controller must satisfy a set of specifications and design objectives, a MOOD procedure could provide a deep insight into the controller's performance and capabilities. In counterpart, more time is required for optimization and decision making stages. Although several performance measurements are available, according to [3][2] the basic specifications will cover:

- Load disturbance rejection/attenuation.
- Measurement noise immunity/attenuation.
- Set point follow-up.
- Robustness to model uncertainties.

It is worthwhile noting how the selection of the optimization objectives for measuring the desired performance can be achieved. A convenient feature of using MOEAs is the flexibility to select interpretable objectives for the designer. That is, the objective selection can be close to the point of view of the designer. Sometimes, with classical optimization approaches, a cost function is built to satisfy a set of requirements such as convexity and/or continuity; that is, it is built from the point of view of the optimizer, in spite of a possible loss of interpretability for the designer. Therefore, the MOP statement is not a trivial task, since the problem formulation from the point of view of the designer is not that of the optimizer [45].

Given the MOP definition some characteristics for the MOEA could be required. That is, according to the expected design alternatives, the MOEA would need to include certain mechanisms or techniques to deal with the optimization statement. Some examples are related with robust, multi-modal, dynamic and/or computationally expensive optimization. Therefore, such instances could lead to certain desirable characteristics for the optimizer, which will be discussed in advance.

[2] Although specified in the context of PID control, they are applicable to all types of controllers.

2.2.2 Evolutionary Multiobjective Optimization (EMO)

Some of the classical strategies to approximate the Pareto Set/Front include: Normal constraint method [86, 116], Normal boundary intersection (NBI) method [24], Epsilon constraint techniques [91] and Physical programming [87]. In [55], a Matlab© toolbox kit for automatic control[3] is presented that includes some of the aforementioned utilities for multiobjective optimization. For the interested reader, in [81, 91] reviews of general optimization statements for MOP in engineering are given. However, as noticed earlier, this book focuses on the MOOD procedure by means of EMO so MOEAs will be discussed.

MOEAs have been used to approximate a Pareto set [144], due to their flexibility when evolving an entire population towards the Pareto front. A comprehensive review of the early stages of MOEAs is contained in [20]. There are several popular evolutionary and nature-inspired techniques used by MOEAs. The former, mainly based on the laws of natural selection where the fittest members (solutions) in a population (set of potential solutions) are more likely to survive as the population evolves. The latter is based on the natural behavior of organisms. Anyway in both cases a population is evolved towards the (unknown) Pareto Front. We will refer to them simply as evolutionary techniques.

The most popular techniques include Genetic Algorithms (GA) [69, 122], Particle Swarm Optimization (PSO) [15, 65], and Differential Evolution (DE) [27, 28, 90, 128]. Nevertheless, evolutionary techniques as Artificial Bee Colony (ABC) [64], Ant Colony Optimization (ACO) [33, 93] of Firefly algorithms [42] are becoming popular. No evolutionary technique is better than the others, since each has its drawbacks and advantages. These evolutionary/nature-inspired techniques require mechanisms to deal with EMO since they were originally used for single objective optimization. While the dominance criterion (Definition 2.1) could be used to evolve the population towards an approximated Pareto Front, it could be insufficient to achieve a minimum degree of satisfaction in other desirable characteristics for a MOEA (diversity, for instance). In Algorithm 2.1 a general structure for a MOEA is given. Its structure is very similar to most evolutionary techniques [43]: it builds and evaluates an initial population $P|_0$ (lines 1–2) and archives an initial Pareto Set approximation (line 3). Then, optimization (evolutionary) process begins with the lines 5–10. Inside this optimization process, the evolutionary operators (depending on the evolutionary technique) will build and evaluate a new population (line 7–8), and the solutions with better cost function will be selected for the next generation (line 10). The main difference is regarding line 9, where the Pareto Set approximation is updated; according to the requirements of the designer, such process will incorporate (or not) some desirable features.

Desirable characteristics for a MOEA could be related to the set of (useful) solutions required by the DM or the optimization design statement at hand (Fig. 2.5). Regarding a Pareto Set, some desirable characteristics include (in no particular order) convergence, diversity and pertinency. Regarding the optimization statement,

[3]Freely available at http://www.acadotoolkit.org/.

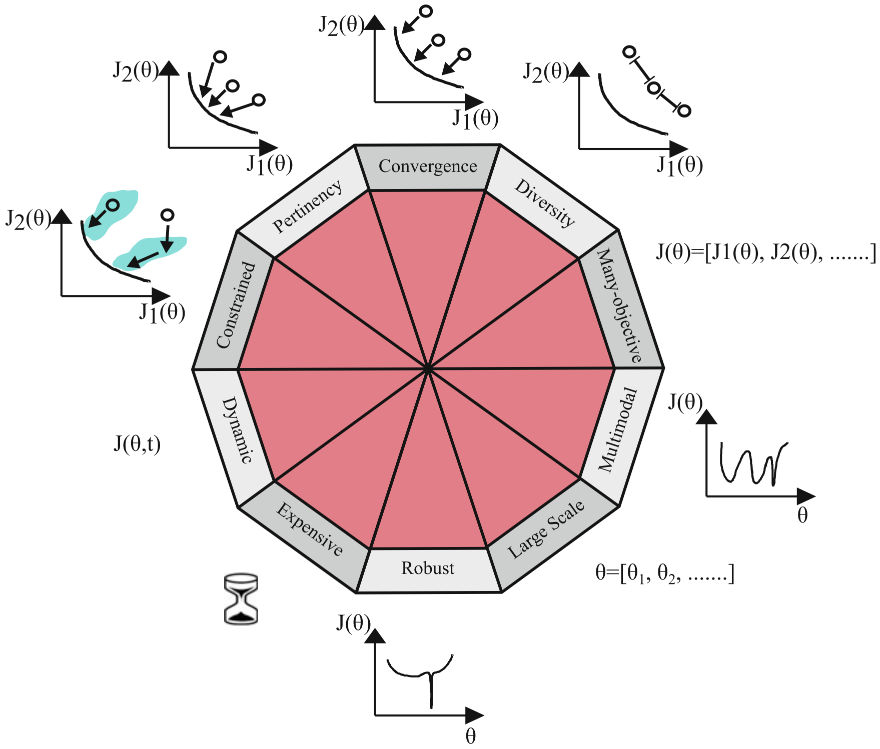

Fig. 2.5 Desired properties for MOEAs

Build initial population $P|_0$ with N_p individuals;
Evaluate $P|_0$;
Build initial Pareto set approximation $\boldsymbol{\Theta}^*_P|_0$ with $P|_0$;
Set generation counter $G = 0$;
while *convergence criteria not reached* **do**
  $G = G + 1$;
  Build population $P^*|_G$ using $P|_{G-1}$ with an evolutionary or bio-inspired technique;
  Evaluate new population $P^*|_G$;
  Build Pareto set approximation $\boldsymbol{\Theta}^*_P|_G$ with $\boldsymbol{\Theta}^*_P|_{G-1} \bigcup P^*|_G$;
  Update population $P|_G$ with $P^*|_G \bigcup P|_{G-1}$;
end
RETURN Pareto set approximation $\boldsymbol{\Theta}^*_P|_G$;

Algorithm 2.1: Basic MOEA.

some features could be related to deal with constrained, many-objectives, dynamic, multi-modal, robust, computationally expensive or large scale optimization instances. These desired characteristics are also a guide to appreciate current trends and on going research on EMO and MOEAs development [30, 144].

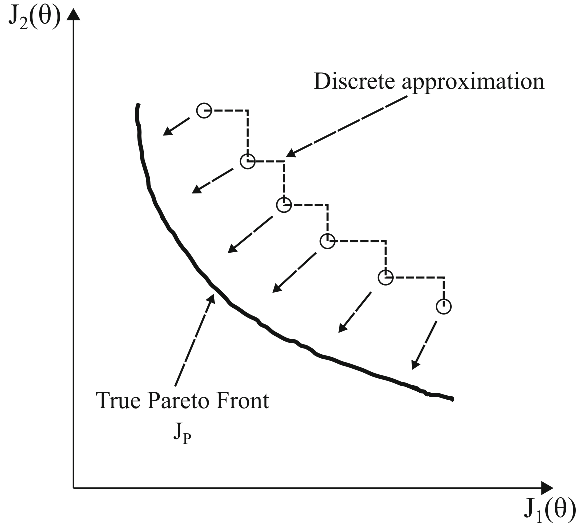

Fig. 2.6 Convergence towards the pareto front

Feature 1 *Convergence*

Convergence is the algorithm's capacity to reach the real (usually unknown) Pareto front (Fig. 2.6). Convergence properties usually depend on the evolutionary parameters of the MOEA used. Because of this, several adaptation mechanisms are available as well as several *ready to use* MOEAs with a default set of parameters. For example, the CEC (Congress on Evolutionary Computation) benchmarks on optimization [58, 142] provide a good set of these algorithms, comprising evolutionary techniques as GA, PSO and DE. Another idea to improve the convergence properties of a MOEA is by means of using local search routines through the evolutionary process. Such algorithms are know as memetic algorithms [95, 98].

Evaluating the convergence of a MOEA over another is not a trivial task, since you are comparing Pareto front approximations. For two objectives it could not be an issue, but in several dimensions is more difficult. Several metrics have been developed to evaluate the convergence properties (and other characteristics) for MOEAs [67, 148].

Convergence is a property common to all optimization algorithms; from the user's point of view it is an expected characteristic. Nevertheless, in the case of MOEAs it could be insufficient, and another desired (expected) feature, as diversity, is required.

Feature 2 *Diversity Mechanism*

Diversity is the algorithm's capacity to obtain a set of well-distributed solutions on the objective space; thus providing a useful description of objectives and decision variables trade-off (Fig. 2.7). Popular ideas include pruning mechanisms, spreading measures or performance indicators of the approximated front.

Regarding pruning mechanisms, probably the first technique was the ϵ-dominance method [70], which defines a threshold where a solution dominates other solutions

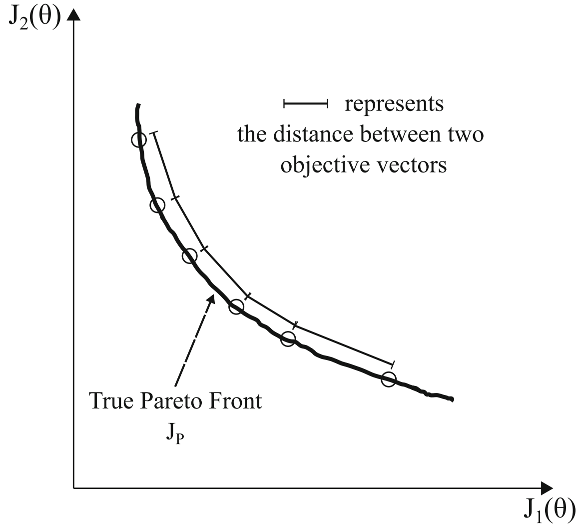

Fig. 2.7 Diversity notion in the pareto front

in their surroundings. That is, a solution dominates the solutions that are less fit for all the objectives, as well as the solutions inside a distance than a given parameter ϵ. Such dominance relaxation has been shown to generate Pareto Fronts with some desirable pertinency characteristics [82]. Algorithms based on such concept include ev-MOGA[4] [52], paϵ-MyDE [51], and paϵ-ODEMO [48]. Similar ideas have been developed using spherical coordinates (or similar statements) [5, 10, 113] in the objective or decision space.

In regard to spreading measures, the crowding distance [31] is used to instigate an algorithm to migrate its population to less crowded areas. This approach is used in algorithms such as NSGA-II[5] [31], which is a very popular MOEA. Other algorithms such as MOEA/D[6] [141] decompose the problem in several scalar optimization subproblems, which are solved simultaneously (as in NBI algorithm) and thereby assure diversity as a consequence of space segmentation when defining the scalar subproblems.

In the case of performance indicators, instead of comparing members of the population, at each generation solutions who best build a Pareto Front are selected based on some performance indicator. An example is the IBEA algorithm [147] which is an indicator based evolutionary algorithm. Most used performance indicators are the hypervolume and the epsilon-indicator [148].

However a good diversity across the Pareto Front must not be confused with solution pertinency (meaning, interesting and valuable solutions from the DM point of view). Several techniques trying to accomplish a good diversity on the Pareto Front

[4] Available for Matlab© at: http://www.mathworks.com/matlabcentral/fileexchange/31080.

[5] Source code available at: http://www.iitk.ac.in/kangal/codes.shtml; also, a variant of this algorithm is available in the global optimization toolbox of Matlab©.

[6] Matlab© code available at: http://cswww.essex.ac.uk/staff/zhang/IntrotoResearch/MOEAd.htm.

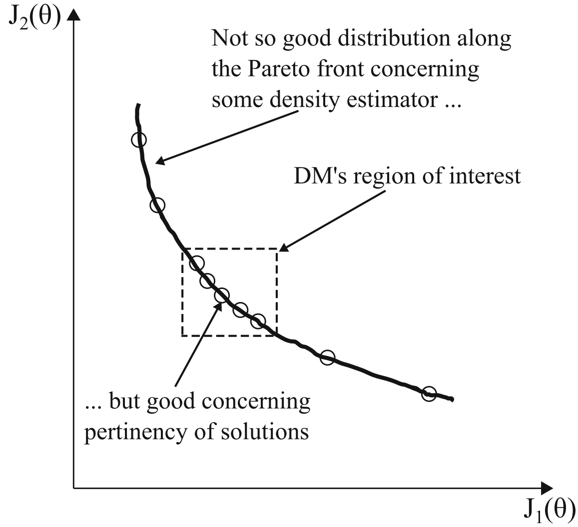

Fig. 2.8 Pertinency notion

seem to be based on (or compared with) uniform distributions. Nevertheless, a large set of solutions may not be of interest to the DM, owing to a strong degradation in one (or several) objectives [22]. Therefore, some mechanisms to incorporate designer preferences could be desirable to improve solutions pertinency.

Feature 3 *Pertinency*

Incorporating DM preferences into a MOEA has been suggested in order to improve the pertinency of solutions. That is, improving the capacity to obtain a set of interesting solutions from the DM point of view (Fig. 2.8). Several ways to include designer's preferences in the MOOD procedure comprise *a priori*, *progressive*, or *a posteriori* methods [96].

- A priori: the designer has some knowledge about his/her preferences in the objective space. In such cases, you could be interested in an algorithm that could incorporate such preferences in the optimization procedure.
- Progressive: the optimization algorithm embeds the designer into the optimization process adjusting his/her preferences *on the fly*. This could be a desirable characteristic for an algorithm when the designer has some knowledge of the objectives trade-off in complex problems.
- A posteriori: the designer analyzes the Pareto Front calculated by the algorithm and, according to the set of solutions, he/she defines his/her preferences in order to select the preferable solution.

Some popular techniques include ranking procedures, goal attainment, fuzzy relations, among others [14]. Improving pertinency in multiobjective algorithms could have a direct and positive impact in the MCDM stage, since the DM could be provided with a more compact set of potential and interesting solutions. It has been suggested that the size of the Pareto Front approximation must be kept to a manageable size for the DM. According to [87] it is usually impossible to retain information from more than 10 or 20 design alternatives.

A natural choice to improve solutions' pertinency is the inclusion of optimization constraints (besides bound constraints on decision variables). This topic will be exposed below.

Feature 4 *Constrained optimization*

Another desirable characteristic in MOEAs is constraint handling. Since most of the design optimization problems need to consider constraints, such mechanisms are always an interesting topic of research. Various techniques have been developed for evolutionary optimization [16, 44]. In [89], those techniques are classified as:

- Feasibility rules. An easy and basic manner to implement the approach is discussed in [29]. It consists in:
  - When comparing two feasible solutions, the one with the best objective function is selected.
  - When comparing a feasible and an infeasible solution, the feasible one is selected.
  - When comparing two infeasible solutions, the one with the lowest sum of constraint violation is selected.
- Stochastic ranking. This approach briefly consists in comparing two infeasible solutions by their fitness or by their constraint violations.
- ϵ-constrained method. This method uses a lexicographic ordering mechanism where the minimization of the constraint violation precedes the minimization of the objective function. This mechanism, with an adaptive parameter scheme,[7] won the CEC2010 competition in a special session on constrained real-parameter optimization [77].
- Novel penalty functions and novel special operators.
- Multiobjective concepts. In the case of MOO, it can be a straightforward approach where the constraint is treated as an additional objective to optimize towards a desired value (goal vector).
- Ensemble of constraint-handling techniques. This approach involves taking advantage of all the mechanisms for constraint handling and using them on a single optimization run (for example [78]).

Regarding controller tuning, constrained optimization instances may appear in complex processes where, for example, several constraints on settling time, overshoot and robustness must be fulfilled.

[7] Code available at http://www.ints.info.hiroshima-cu.ac.jp/~takahama/eng/index.html for single objective optimization.

Feature 5 *Many-Objectives optimization*

Algorithms with good diversity preservation mechanisms could have problems if solutions are dominance resistant in an m-dimensional objective space and so waste time and resources in non-optimal areas [104]. This is because of the self diverse nature and the large number of objectives (usually, $m \geq 5$). Furthermore, recent research has indicated that a random search approach can be competitive for generating a Pareto Front approximation for a many-objectives optimization [22]. Several approaches to deal with many-objectives optimization include [61]:

- Modification of Pareto dominance to improve the selection pressure towards the Pareto Front.
- Introduction of different ranks to define a metric based on the number of objectives for which a solution is better than the other.
- Use of indicator functions as performance indicators of the quality of the Pareto Front approximation.
- Use of scalarizing functions (weighting vectors, for example).
- Use of preference information (see above), that is, information on the region of interest for the DM.
- Reduction in the number of objectives.

Examples to deal with this last issue can be seen in [75] where an objective reduction is performed using principal component analysis (PCA), or in [120] where a heuristic approach is used for dimensional reduction. Besides, algorithms which incorporate preference information in the optimization approach could be used in many-objective instances [61].

In the specific case of controller tuning, a many-objective optimization instance would appear according with the complexity of a given control loop or process, and the number of requirements to fulfill.

Feature 6 *Dynamic optimization*

Sometimes the static approach is not enough to find a preferable solution and therefore, a dynamic optimization statement needs to be solved where the cost function is varying with time. The challenge, besides tracking the optimal solution, is to select the desired solution at each sampling time. In [23, 36] there are extensive reviews on this topic.

As it can be noticed, this kind of capabilities would be useful for problems related with Model Predictive Control (MPC) where a new control value is obtained at each sampling time taking into account new information of the process outputs.

Feature 7 *Multi-modal Optimization*

Multi-modal instances for controller tuning *per se* seem to be not usual; nevertheless they may appear in multi-disciplinary optimization [83] statements, where besides the tuning parameters, other design variables (as mechanical or geometrical) are involved.

In multi-modal optimization, different decision variable vectors could give the same objective vector. In some instances, it could be desirable to retain such solutions and perform, in the MCDM step, an analysis according with the decision space region where those solutions belongs. This could be important in instances where, for example, the decision variables have a physical meaning and it is convenient to analyze the impact of using one over another. In a EMO framework, this information could be added as additional objectives as noticed by [32]. For more details on multi-modal optimization, the interested reader could refer to [26].

Feature 8 *Robust Optimization*

In a general frame and according to [7], robust optimization could refer not only to the models used to measure the performance, but also with the sensitivity analysis on the calculated solutions. That is, how much could be degraded the objective vector under the presence of uncertainties. This sensibility analysis could be done by means of deterministic measures and/or with direct search (as Montecarlo methods). This kind of analysis could bring a different level of interpretability of the performance due to uncertainties in the model used in the optimization. This problem statement is related with reliability optimization, where a given performance must be assured for a certain solution along different scenarios.

An example is provided in [124] where an evaluation of the *American Control Conference Benchmark* [136] based on Montecarlo methods is done.

Feature 9 *Computationally Expensive optimization*

Computationally expensive optimization is related with line 8 of Algorithm 2.1. Sometimes cost function evaluation requires a huge amount of computational resources. Therefore stochastic approaches are a disadvantage, given the complexity to evaluate the fitness (performance) of an individual (design alternative). Recent solutions are mainly oriented to generate a surface *on-the-fly* of the objective space, with lower computational effort. One popular technique is the use of Neural Networks, trained through an evolutionary process, but any kind of model or surface approximation could be used. A review on the topic can be consulted in [117]. In the field of control systems engineering, such type of instances would appear when expensive calculations in complex simulations are needed to compute the objective vector.

In other instances, such computational effort could be relative; that is, there are limited computational resources to evaluate a cost function. To deal with this issue *compact evolutionary algorithms* has been proposed, but such instance has not reach yet the EMO approach. Some examples are exposed in [50] and [92]. Instances where this capabilities could be desirable include embedded solvers for optimization.

Feature 10 *Large scale optimization*

It refers to the capabilities of a given MOEA to deal with MOP with any number of decision variables with reasonable computational resources. Sometimes a MOEA can perform well for a relatively small number of decision variables, but it could be an impractical solution (according to the computational resources available) to solve a problem with a bigger number of decision variables. Whilst in expensive optimization instances (Feature 9) the complexity is due to the performance measurement (line 8 in Algorithm 2.1), in large scale may be related to the algorithm's mechanism to approximate a new set of design alternatives (lines 7 and 9). In the former the complexity is added by the problem, in the latter by the algorithm. A review on this topic can be consulted in [74].

The aforementioned features could be desirable characteristics for a given MOEA. Afer all, it would depend on the designer's preferences and the MOP statement at hand. Afterwards, a MCDM step must be carried, in order to select the most preferable solution. This step is commented below.

2.2.3 MultiCriteria Decision Making (MCDM)

Once the DM has been provided with a Pareto Front $\boldsymbol{J}_P^*$, she/he will need to analyze the trade-off between objectives and select the best solution according to her/his preferences. A comprehensive compendium on MCDM techniques (and software) for multi-dimensional data and decision analysis can be consulted in [41]. Assuming that all preferences have been handled as much as possible in the optimization stage, a final selection step must be taken with the approximated Pareto Front. Here we will emphasize the trade-off visualization.

It is widely accepted that visualization tools are valuable and provide the DM with a meaningful method to analyze the Pareto Front and make decisions [73]. Tools and/or methodologies are required for this final step to successfully embed the DM into the solution refinement and selection process. It is useful if the DM understands and appreciates the impact that a given trade-off in one sub-space could have on others [9]. Even if an EMO process has been applied to a reduced objective space, sometimes the DM needs to increase the space with additional metrics or measurements to have confidence in her/his own decision [9]. Usually, analysis on the Pareto Front may be related with design alternatives comparison and design concepts comparison.

For two-dimensional problems (and sometimes for three-dimensional ones) it is usually straightforward to make an accurate graphical analysis of the Pareto Front (see for example Fig. 2.9), but difficulty increases with the problem dimension. Tools such as VIDEO [68] incorporate a color coding in three-dimensional graphs to analyze trade-offs for 4-dimensional Pareto fronts. In [73], a review on visualization techniques includes techniques such as decision maps, star diagrams, value paths, GAIA, and heatmap graphs. Possibly the most common choices for Pareto Front visualization and analysis in control systems applications are: scatter diagrams, parallel coordinates [60], and level diagrams [8, 109].

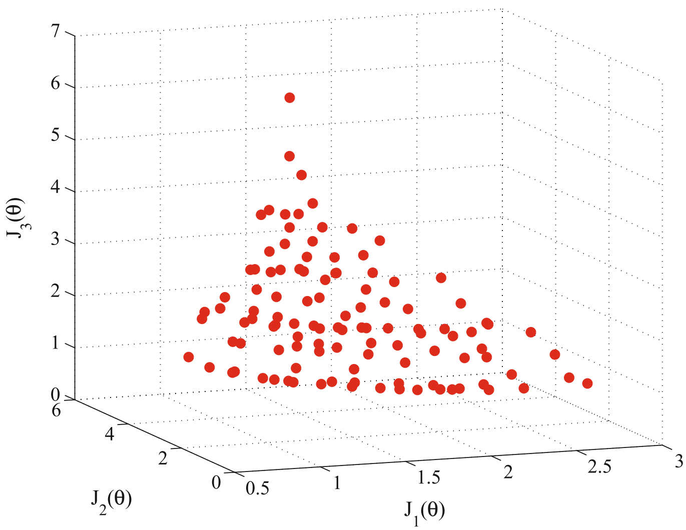

Fig. 2.9 3D Visualization of a 3-dimensional pareto front

Scatter diagram plots (SCp)[8] are straightforward visualizations. They generate an array of 2-D graphs to visualize each combination of a pair of objectives (see Fig. 2.10). This type of visualization is enough for two dimensional problems. To appreciate all the trade-offs of an m-dimensional Pareto Front, at least $\frac{m(m-1)}{2}$ combination plots are required. For example, the Pareto Front of Fig. 2.9 is visualized using SCp in Fig. 2.10. If the DM would like to see the trade-off for an objective and a decision variable from the n-dimensional decision space, she/he will need n times m additional plots.

Parallel coordinate (PAc) visualization strategy [60] plots an m-dimensional objective vector in a two-dimensional graph.[9] For each objective vector $\boldsymbol{J}(\boldsymbol{\theta}) = [J_1(\boldsymbol{\theta}), \ldots, J_m(\boldsymbol{\theta})]$ the ordered pairs $(i, J_i(\boldsymbol{\theta}))$, $i \in [1, \ldots, m]$ are plotted and linked with a line. This is a very compact way of presenting multidimensional information: just one 2-D plot is required. Nevertheless, to entirely represent the trade-off surface some axis relocation may be necessary. For example, in Fig. 2.11, it is possible to appreciate the PAc visualization of the Pareto Front depicted in Fig. 2.9. To appreciate tendencies with the decision space variable, an extended plot with $n+m$ vertical axes is required. An independent graph could be plotted, but some strategy (such as color coding) will be needed to link an objective vector with its corresponding decision vector in order to appreciate trade-off information from the objective space. This kind of feature is incorporated in visualization tools such as TULIP from INRIA,[10]

[8]Tool available in Matlab©.

[9]Tool available in the statistics toolbox of Matlab©.

[10]Available at http://tulip.labri.fr/TulipDrupal/. Includes applications for multidimensional analysis.

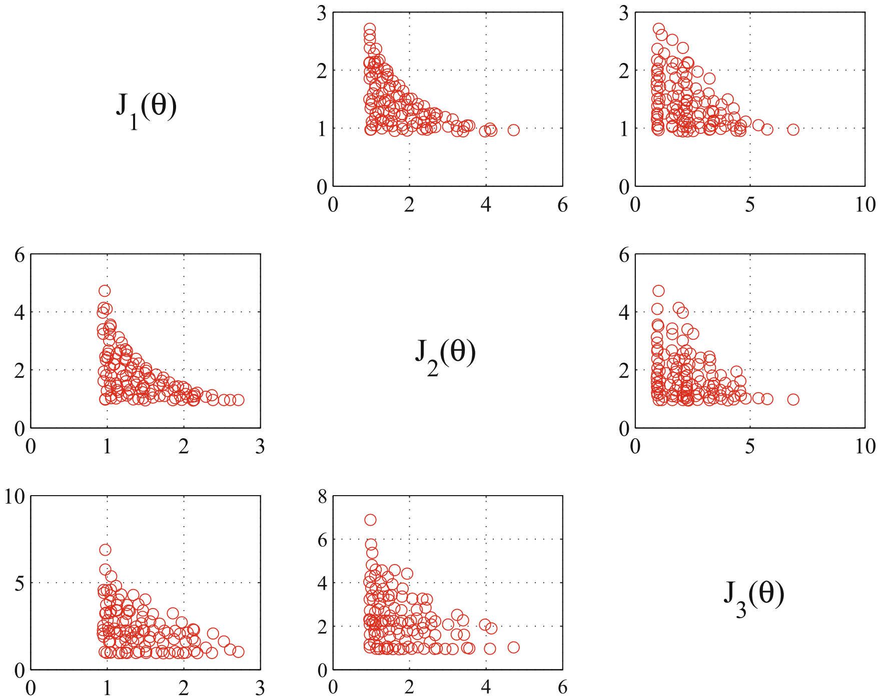

Fig. 2.10 Scatter plot (SCp) visualization for pareto front of Fig. 2.9

which are also helpful for analyzing multidimensional data. Finally, a normalization or y-axis re-scaling can be easily incorporated, if required, to facilitate the analysis.

The *Level Diagrams* (LD) visualization [8][11] is useful for analyzing m-objective Pareto Fronts [145, 146], as it is based on the classification of the approximation $\boldsymbol{J}_P^*$ obtained. Each objective $J_i(\boldsymbol{\theta})$ is normalized $\hat{J}_i(\boldsymbol{\theta})$ with respect to its minimum and maximum values. To each normalized objective vector $\hat{\boldsymbol{J}}(\boldsymbol{\theta})$ a p-norm is applied to evaluate the distance to an ideal[12] solution $\boldsymbol{J}^{ideal}$. The LD tool displays a two dimensional graph for each objective and decision variable. The ordered pairs $\left(J_i(\boldsymbol{\theta}), \|\hat{\boldsymbol{J}}(\boldsymbol{\theta})\|_p\right)$ in each objective sub-graph and $\left(\theta_l, \|\hat{\boldsymbol{J}}(\boldsymbol{\theta})\|_p\right)$ in each decision variable sub-graph are plotted (a total of $n + m$ plots). Therefore, a given solution will have the same y-value in all graphs (see Fig. 2.12). This correspondence will help to evaluate general tendencies along the Pareto Front and to compare solutions according to the selected norm. Also, with this correspondence, information from the objective space is directly embedded in the decision space, since a decision vector inherits its *y-value* from its corresponding objective vector.

---

[11] GUI for Matlab© is available at: http://www.mathworks.com/matlabcentral/fileexchange/24042.

[12] By default, minimum values for each objective in $\hat{\boldsymbol{J}}(\boldsymbol{\theta})$ could be used to build an ideal solution.

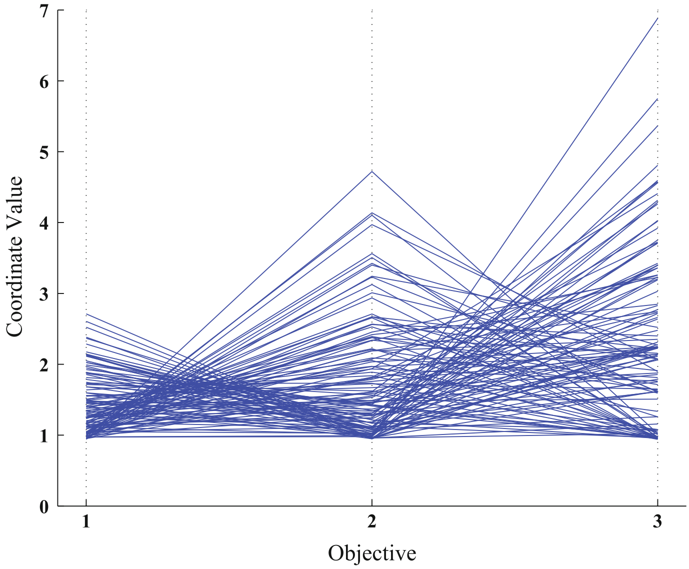

Fig. 2.11 Parallel coordinates plot (PAc) visualization for pareto front of Fig. 2.9

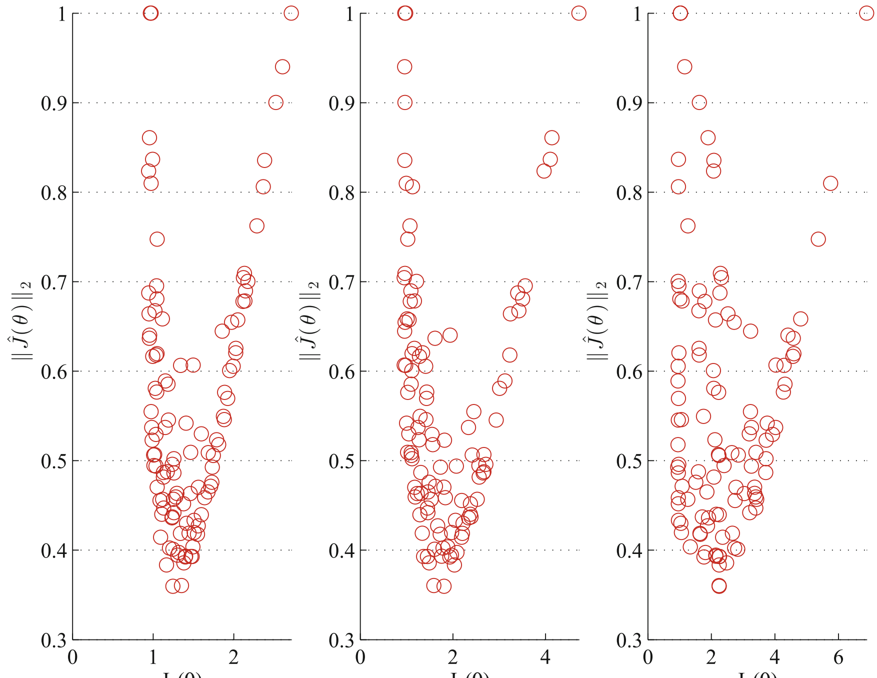

Fig. 2.12 Level diagram (LD) visualization for pareto front of Fig. 2.9

In any case, characteristics required for such a visualization were described in [73]: simplicity (must be understandable); persistence (information must be rememberable by the DM); and completeness (all relevant information must be depicted). Some degree of interactivity with the visualization tool is also desirable (during and/or before the optimization process) to successfully embed the DM into the selection process.

2.3 Related Work in Controller Tuning

As noticed in the previous chapter, multiobjective techniques might be useful for controller tuning applications. This section will provide a brief listing on related work over the last ten years (expanding and updating [115]), with a focus on four controller structures (design concepts): PID-like, State space representation, fuzzy control and model predictive control. While several works have dealt with MOP (using an AOF for example), those where dominance and Pareto Front concepts have been used actively for controller tuning purposes will be included.

Control engineers might select different design objectives in order to evaluate a given controller performance in the feedback loop. According to the basic control loop of Fig. 2.13, such design objectives are typically selected in order to have a measure of:

- Tracking performance of the set point (reference) $r(t)$.
- Rejection performance of load disturbance $d(t)$.
- Robustness to measurement noise $n(t)$.
- Robustness to model uncertainty.

Different measures are used for such purposes, typically in frequency and time domains.

2.3.1 Basic Design Objectives in Frequency Domain

- Maximum value of the complementary sensitivity function.

$$J_{M_p}(\boldsymbol{\theta}) = \left\| P(s)C(s)(I + P(s)C(s))^{-1} \right\|_{\infty} \tag{2.4}$$

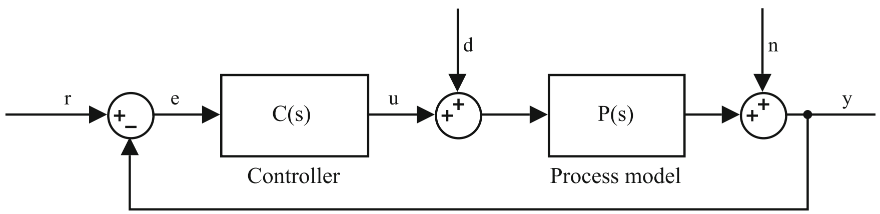

Fig. 2.13 Basic control loop

- Disturbance attenuation performance.

$$J_{W_1}(\boldsymbol{\theta}) = \left\| W(s) \cdot (I + P(s)C(s))^{-1} \right\|_\infty < 1 \tag{2.5}$$

- Maximum value of noise sensitivity function.

$$J_{M_u}(\boldsymbol{\theta}) = \left\| C(s)(I + P(s)C(s))^{-1} \right\|_\infty \tag{2.6}$$

- Maximum value of the sensitivity function.

$$J_{M_s}(\boldsymbol{\theta}) = \left\| (I + P(s)C(s))^{-1} \right\|_\infty \tag{2.7}$$

- Robust stability performance.

$$J_{W_2}(\boldsymbol{\theta}) = \left\| W(s) \cdot (P(s)C(s)(I + P(s)C(s))^{-1}) \right\|_\infty < 1 \tag{2.8}$$

where $W(s)$ are weighting transfer functions commonly used in mixed sensitivity techniques.

2.3.2 Basic Design Objectives in Time Domain

- Integral of the absolute error value.

$$J_{IAE}(\boldsymbol{\theta}) = \int_{t=t_0}^{t_f} |r(t) - y(t)| \, dt \tag{2.9}$$

- Integral of the time weighted absolute error value.

$$J_{ITAE}(\boldsymbol{\theta}) = \int_{t=t_0}^{t_f} t \, |r(t) - y(t)| \, dt \tag{2.10}$$

- Integral of the squared error value.

$$J_{ISE}(\boldsymbol{\theta}) = \int_{t=t_0}^{t_f} (r(t) - y(t))^2 \, dt \tag{2.11}$$

- Integral of the time weighted squared error value.

$$J_{ITSE}(\boldsymbol{\theta}) = \int_{t=t_0}^{t_f} t \, (r(t) - y(t))^2 \, dt \tag{2.12}$$

- Settling time: time elapsed from a step change input to the time at which $y(t)$ is within a specified error band of $\Delta\%$.

$$J_{t(100-\Delta)\%}(\boldsymbol{\theta}) \tag{2.13}$$

- Overshoot (for a positive input change).

$$J_{over}(\boldsymbol{\theta}) = \max\left[\max\left(\frac{y(t)-r(t)}{r(t)}\right), 0\right], t \in [t_0, t_f] \tag{2.14}$$

- Maximum deviation (for a load disturbance).

$$J_{overd}(\boldsymbol{\theta}) = \max\left(\left|\frac{y(t)-r(t)}{r(t)}\right|\right), t \in [t_0, t_f] \tag{2.15}$$

- Integral of the squared control action value.

$$J_{ISU}(\boldsymbol{\theta}) = \int_{t=t_0}^{t_f} u(t)^2 dt \tag{2.16}$$

- Integral of the absolute control action value.

$$J_{IAU}(\boldsymbol{\theta}) = \int_{t=t_0}^{t_f} |u(t)| dt \tag{2.17}$$

- Total variation of control action.

$$J_{TV}(\boldsymbol{\theta}) = \int_{t=t_0}^{t_f} \left|\frac{du}{dt}\right| dt \tag{2.18}$$

- Maximum value of control action.

$$J_{maxU}(\boldsymbol{\theta}) = \max(u(t)), t \in [t_0, t_f] \tag{2.19}$$

where $r(t)$, $y(t)$, $u(t)$ are the set-point, controlled variable and manipulated variable respectively in time t. Such objectives, for the sake of simplicity, have been stated in a general sense.

2.3.3 PI-PID Controller Design Concept

PID controllers are reliable control solutions thanks to their simplicity and efficacy [3, 4]. They represent a common solution for industrial applications and therefore, there is still ongoing research on new techniques for robust PID controller tuning [135]. Any improvement in PID tuning is worthwhile, owing to the minimum number of changes required for their incorporation into already operational control loops [125, 130]. As expected, several works have focused on the PID performance improvement.

Given a process model $P(s)$, the following general description for a PID controller of two-degree-of-freedom is used (see Fig. 2.14):

$$\begin{aligned} C(s) &= K_c\left(b + \frac{1}{T_i s^{\lambda}} + c\frac{T_d \cdot s^{\mu}}{\frac{T_d}{N}s^{\mu} + 1}\right)R(s) \\ &\quad - K_c\left(1 + \frac{1}{T_i s^{\lambda}} + \frac{T_d \cdot s^{\mu}}{\frac{T_d}{N}s^{\mu} + 1}\right)Y(s) \end{aligned} \tag{2.20}$$

where K_c is the proportional gain, T_i the integral time, T_d the derivative time, N the derivative filter, a, b the set-point weighting for proportional and derivative actions; λ and μ are used to represent a PID controller with fractional order [103]. Therefore, the following design concepts (controllers) with their decision variables can be stated:

PI: $\boldsymbol{\theta}_{PI} = [K_c, T_i]$. $b = 1, Td = 0, \lambda = 1$.
PD: $\boldsymbol{\theta}_{PD} = [K_c, T_d]$. $b = c = 1, \frac{1}{N} = 0, \frac{1}{T_i} = 0, \mu = 1$.
PID: $\boldsymbol{\theta}_{PID} = [K_c, T_i, T_d]$. $b = c = 1, \frac{1}{N} = 0, \lambda = 1, \mu = 1$.
PID/N: $\boldsymbol{\theta}_{PID/N} = [K_c, T_i, T_d, N]$. $b = c = \lambda = \mu = 1$.
PI^1: $\boldsymbol{\theta}_{PI^1} = [K_c, T_i, b]$. $T_d = 0, \lambda = 1$.
PID^2: $\boldsymbol{\theta}_{PID^2} = [K_c, T_i, T_d, b, c]$. $\frac{1}{N} = 0, \lambda = \mu = 1$.
PID^2/N: $\boldsymbol{\theta}_{PID^2/N} = [K_c, T_i, T_d, N, b, c]$. , $\lambda = \mu = 1$.
PI$^\lambda$D$^\mu$: $\boldsymbol{\theta}_{FOPID} = [K_c, T_i, T_d, \lambda, \mu]$. $b = c = 1, \frac{1}{N} = 0$.

In Table 2.1 a summary of contributions using these design concepts is provided. Brief remarks on MOP, EMO and MCDM for each work are given. Regarding the MOP, it is important to notice that there are more works focusing on controller tuning for SISO loops; besides, there is also an equilibrium with MOP problems dealing

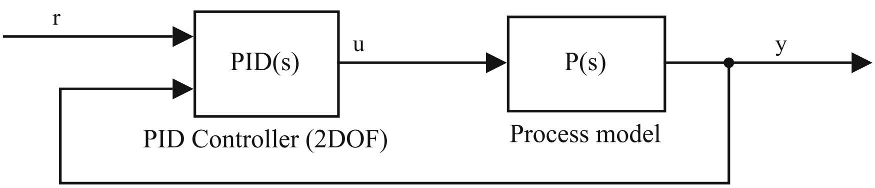

Fig. 2.14 Control loop with a two-degree-of-freedom PID (2DOF-PID) controller

Table 2.1 Summary of MOOD procedures for PID design concept. MOP refers to the number of design objectives; EMO to the algorithm implemented (or used as basis for a new one) in the optimization process. MCDM to the visualization and selection process used

Concept(s)	Process(es)	References	MOP	EMO	MCDM
PID^2/N, PI^1	SISO, MIMO	[53]	4	GA	3D, SCp Concepts comparison
PI^1	FOPDT	[131]	4	GA	3D, 2D Tuning rule methodology
PID	Electromagnetic valve actuator	[126]	7	GA	PAc Iterative controllability analysis for a given design
PID/N	SISO	[57]	3	Ad hoc	SCp Incorporate analysis of time domain objectives
PID	Aeronautical longitudinal control system for an aircraft	[59]	3	SA	3D analysis with other tuning techniques
PD	Mechatronical design (mechanical and control)	[106]	5	GA	SCp design alternatives comparison
PID^2/N	SISO	[108]	15	GA	LD selection according to preferences
PI	Alstom Gasifier MIMO process	[138]	6	NSGA-II	SCp, new indicator included for selection
PID	Flexible AC transmission system	[119]	2	NSGA-II	Fuzzy based selection
PID, I-PD	Chemotherapy control	[1]	2	GA	SCp concepts comparison; intended to support specific treatment
PID	Methanol-Ethanol distillation column, F18/HARV aircraft	[143]	2	PSO	SCp, AFO selection
PI	Wood and Berry MIMO system	[112]	7	DE	LD design alternatives analysis
$PI^{\lambda}D^{\mu}$	SISO	[49]	5	GA	LD design alternatives comparison
$PI^{\lambda}D^{\mu}$	SISO,Hydraulic turbine regulating system	[11]	2	NSGA-II	2D design concepts comparison with a PID
PI, PID/N	Two-area non-reheat thermal system	[102]	3	NSGA-II	Fuzzy-based membership value assignment approach

(continued)

Table 2.1 (continued)

Concept(s)	Process(es)	References	MOP	EMO	MCDM
PI	Speed control reluctance motor	[63]	2	NSGA-II	Selection with an AOF
PID	Twin Rotor MIMO system	[110]	5	DE	LD design alternatives comparison
$PI^{\lambda}D^{\mu}$	Load frequency control	[101]	2	NSGA-II	2D design concepts comparison with PID
$PI^{\lambda}D^{\mu}$	Automatic Voltage Regulator	[100]	2	NSGA-II	2D design concepts comparison
PI	MIMO boiler process	[114]	5	DE	LD design alternatives comparison
PI	MIMO Wood and Berry MIMO system	[111]	7	DE	LD design alternatives comparison
$PI^{\lambda}D^{\mu}$	Automatic Voltage Regulator	[140]	3	Ad hoc	3D design alternative analysis. Design concepts comparison with PID
PI, PID	Two-area non-reheat thermal system; Three-area hydro-thermal power system	[99]	3	GA	SCp, fuzzy-based membership value assignment approach

with 2–3 objectives versus many-objectives optimization. Regarding the optimizer, MOEAs based on GA seem to be more popular for such design concept. In the MCDM stage while a design alternatives comparison is in general performed, the design concepts comparison seems to be more popular when dealing with fractional PID controllers. This is done in order to justify increasing the complexity of the controller. Finally, in the MCDM, classical approaches for visualization based on SCp and 3D representation are the most used, despite the number of objectives managed.

2.3.4 *Fuzzy Controller Design Concept*

Fuzzy systems have been widely and successfully used in control system applications as referenced in [40]. Similar to the use of PID as design concept, the MOOD is useful for analyzing the trade-off between conflicting objectives. In this case, the fuzzy controller is more complex to tune, given its nonlinearity and the major number of variables involved in fuzzyfication, inference and defuzzification steps (see Fig. 2.15).

A comprehensive compendium on the synergy between fuzzy tools and MOEAs is given in [39]. This book will focus on controller implementations. In general, decision variables consider $\boldsymbol{\theta} = [\Lambda, \Upsilon, \overline{\overline{\Lambda}}, \overline{\overline{\Upsilon}}, \mu]$, where:

Λ: is the membership function shape.
$\overline{\overline{\Lambda}}$: is the number of membership functions.
Υ: is the fuzzy rule structure.
$\overline{\overline{\Upsilon}}$: is the number of fuzzy rules.
μ: are the weights of the fuzzy inference system.

In Table 2.2 a summary on these applications is provided. The difference in the quantity of the works dedicated to fuzzy controllers and PID controllers is noticeable.

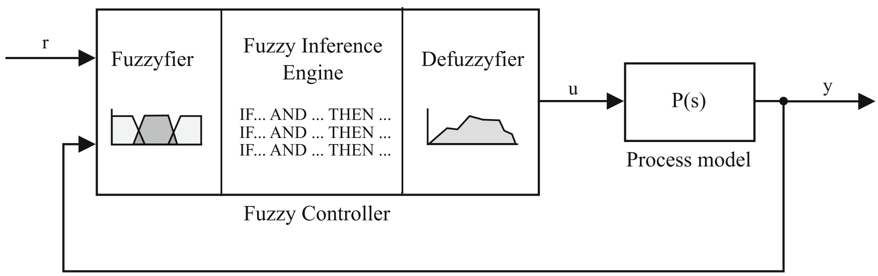

Fig. 2.15 Control loop with a fuzzy controller

Table 2.2 Summary of MOOD procedures for Fuzzy design concept. MOP refers to the number of design objectives; EMO to the algorithm implemented (or used as basis for a new one) in the optimization process. MCDM to the visualization and selection process used

Process(es)	References	MOP	EMO	MCDM
Aeronautical	[12]	9	GA	PAc Constraint violation analysis; fine tuning
DC motor (HiL)	[127]	4	GA	None According performance
Geological	[66]	4	NSGA-II	SCp Design alternatives comparison
Bio-medical	[37]	2	SPEA based	2D Design alternatives/concepts comparison with other controllers. Selection by norm-2 criteria
Mechanical	[80]	3	PSO	3D Design alternatives comparison
HVAC system	[46]	2	SPEA based	2D Design alternatives comparison at two levels: different controllers and different MOEAs
Wall following robot	[56]	4	SPEA based	2D with an AFO

With regard to MOP definition, it seems that EMO has been popular to simultaneously optimize objectives related with performance and interpretation of the fuzzy inference system. Nevertheless, as noticed in [39] scalability issue is a problem worthwhile to address such a design concept. Finally, in the MCDM step, SCp tools have been sufficient for Pareto Front visualization and analysis, due to the low number of objectives stated in the MOP.

2.3.5 State Space Feedback Controller Design Concept

The state space representation has shown to be a remarkable tool for controller design. Several advanced control techniques use this representation to calculate a controller (in the same representation) with a desired performance. In this case, the decision variables are the gains of the matrix K (see Fig. 2.16). Classical optimization approaches in a MOOD framework have been used in [85] with good results. In several instances, it seems that the MOOD procedure has been used to compare classical approaches with the EMO approach, as presented below.

In Table 2.3 a summary on these applications is provided. There are still few works focusing on this design concept.

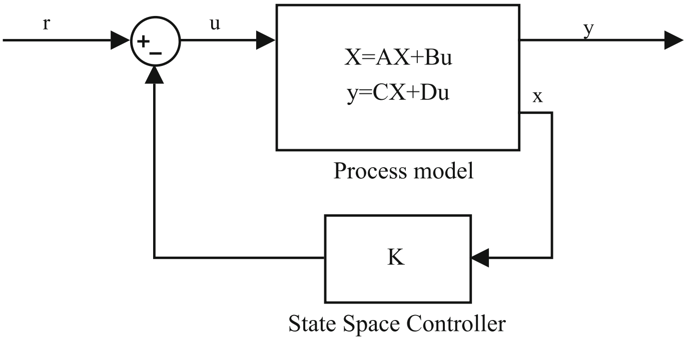

Fig. 2.16 Control loop with a state space controller

Table 2.3 Summary of MOOD procedures for state space representation design concept. MOP refers to the number of design objectives; EMO to the algorithm implemented (or used as basis for a new one) in the optimization process. MCDM to the visualization and selection process used

Process(es)	References	MOP	EMO	MCDM
SISO, MIMO	[54]	3	GA	SCp Concepts comparison with LMI design
SISO	[94]	3	GA	2D Concepts comparison with LMI
Mechanical	[62]	4	GA	SCp Design alternatives comparison
Networked Predictive control, various examples	[25]	2	NSGA-II with LMIs	2D Design alternatives analysis on examples
Biped robot	[76]	2	MOPSO and NSGA-II	2D Design alternatives analysis on examples
Twin Rotor MIMO system	[110]	18	DE	LD Design concepts comparison with a PID controller; design alternatives comparison

2.3.6 *Predictive Control Design Concept*

On-line applications for MOOD are not straightforward, since the MCDM stage must be carried out, in some instances, automatically. As a result, analysis that relies on the DM must be codified to become an automatic process. Approaches using EMO in the MOOD procedure are presented below; where decision variables $\boldsymbol{\theta}$ is conformed by the control action $\boldsymbol{u}$ through the control horizon, see Fig. 2.17.

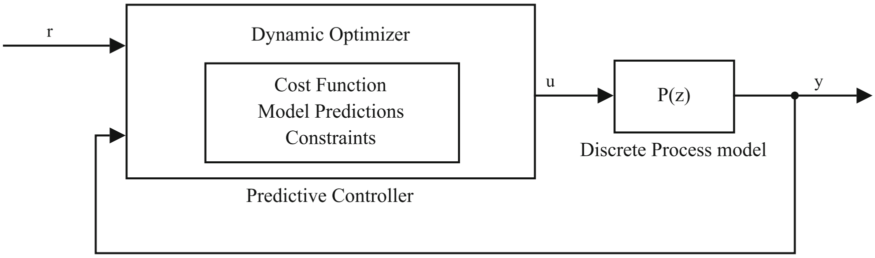

Fig. 2.17 Control loop with a predictive controller

Table 2.4 Summary of MOOD procedures for predictive control design concept. MOP refers to the number of design objectives; EMO to the algorithm implemented (or used as basis for a new one) in the optimization process. MCDM to the visualization and selection process used

Process(es)	References	MOP	EMO	MCDM
Mechanical	[47]	2	GA	Fuzzy inference system is used
Chemical	[13]	8	NSGA-II	Successive ordering according to feasibility
Subway ventilation system	[72]	2	NSGA-II	Decision rule
Smart energy efficient buildings	[118]	2	GA	Decision rule

In Table 2.4 a summary on these applications is provided. Predictive control seems to be an opportunity to apply the MOOD approach, due to the few works dedicated to this control design alternative. Nevertheless, it can also be seen that the problem relies on tracking the Pareto Front each sampling time.

2.4 Conclusions on This Chapter

In this chapter, fundamental concepts regarding multiobjective optimization were introduced. Besides, some notions and remarks on the fundamental steps of a holistic MOOD procedure were commented: the MOP definition, the EMO process and the MCDM stage.

Furthermore, related work on controller tuning applications using such MOOD techniques was revisited including works were dominance and Pareto Front concepts were used actively for controller tuning purposes. Design concepts (controller structures) listed were PID-like controllers, fuzzy structures, state space representation and model predictive control. Even focusing on contributions using EMO, there are also examples solving MOPs with other (deterministic) techniques, for example:

- PID-like: [71, 121].
- State space representation: [137].
- Predictive control: [6, 97, 105, 133].
- Optimal control: [132, 134].

As commented in the previous chapter, MOOD procedures might be a useful tool for controller tuning purposes. With such techniques, it is possible appreciating trade-off between conflictive control objectives (performance and robustness for instance). Which is important to remember is the fundamental question for such techniques:

- *What kind of problems are worth to address with MOOD?*

That question leads to others:

- *Is it difficult to find a controller with a reasonable trade-off among design objectives?*
- *Is it worthwhile analysing the trade-off among controllers (design alternatives)?*

If the answer is yes to both questions, then the MOOD procedure could be an appropriate tool for the problem at hand. Otherwise, other tuning techniques or AOF approaches could be enough.

During the remaining chapters a set of tools and algorithms for EMO and MCDM stage will be presented, in order to provide to the readers an introductory toolbox for MOOD procedures.

References

1. Algoul S, Alam M, Hossain M, Majumder M (2011) Multi-objective optimal chemotherapy control model for cancer treatment. Med Biol Eng Comput 49:51–65. doi:10.1007/s11517-010-0678-y
2. Aslam T, Ng A (2010) Multi-objective optimization for supply chain management: a literature review and new development. In: 2010 8th international conference on supply chain management and information systems (SCMIS) (Oct 2010), pp 1 –8
3. Åström K, Hägglund T (2001) The future of PID control. Control Eng Pract 9(11):1163–1175
4. Åström KJ, Hägglund T (2005) Advanced PID Control. ISA Instrum Syst Autom Soc Res Triangle Park, NC 27709
5. Batista L, Campelo F, Guimar£es F, Ramrez J (2011) Pareto cone ϵ-dominance: improving convergence and diversity in multiobjective evolutionary algorithms. In: Takahashi R, Deb K, Wanner E, Greco S (eds) Evolutionary multi-criterion optimization vol. 6576 of Lecture notes in computer science. Springer, Heidelberg, pp 76–90. doi:10.1007/978-3-642-19893-9-6
6. Bemporad A, noz de la Peña DM (2009) Multiobjective model predictive control. Automatica 45(12):2823–2830
7. Beyer H-G, Sendhoff B (2007) Robust optimization - a comprehensive survey. Comput Meth Appl Mech Eng 196(33–34):3190–3218

8. Blasco X, Herrero J, Sanchis J, Martínez M (2008) A new graphical visualization of n-dimensional pareto front for decision-making in multiobjective optimization. Inf Sci 178(20):3908–3924
9. Bonissone P, Subbu R, Lizzi J (2009) Multicriteria decision making (MCDM): a framework for research and applications. IEEE Comput Intell Mag 4(3):48 –61 (2009)
10. Branke J, Schmeck H, Deb K, Reddy SM (2004) Parallelizing multi-objective evolutionary algorithms: cone separation. In: Congress on evolutionary computation, 2004. CEC2004 (June 2004), vol 2, pp 1952–1957
11. Chen Z, Yuan X, Ji B, Wang P, Tian H (2014) Design of a fractional order PID controller for hydraulic turbine regulating system using chaotic non-dominated sorting genetic algorithm II. Energy Convers Manag 84:390–404
12. Chipperfield A, Bica B, Fleming P (2002) Fuzzy scheduling control of a gas turbine aero-engine: a multiobjective approach. IEEE Trans Indus Electron 49(3):536–548
13. Chuk OD, Kuchen BR (2011) Supervisory control of flotation columns using multi-objective optimization. Miner Eng 24(14):1545–1555
14. Coello C (2000) Handling preferences in evolutionary multiobjective optimization: a survey. In: Proceedings of the 2000 congress on evolutionary computation, vol 1, pp 30–37
15. Coello C (2011) An introduction to multi-objective particle swarm optimizers. In: Gaspar-Cunha A, Takahashi R, Schaefer G, Costa L (eds) Soft computing in industrial applications, vol 96 of Advances in intelligent and soft computing. Springer, Heidelberg, pp 3–12. doi:10.1007/978-3-642-20505-7_1
16. Coello CAC (2002) Theorical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art. Comput Meth Appl Mech Eng 191:1245–1287
17. Coello CAC, Lamont GB (2004) Applications of multi-objective evolutionary algorithms. In: Advances in natural computation, vol 1. World Scientific Publishing
18. Coello CAC, Lamont GB, Veldhuizen DAV (2007) Multi-criteria decision making. In: Evolutionary algorithms for solving multi-objective problems. Genetic and evolutionary computation series. Springer US, pp 515–545
19. Coello CAC., Veldhuizen DV, Lamont G (2002) Evolutionary algorithms for solving multi-objective problems. Kluwer Academic Press
20. Coello Coello C (2006) Evolutionary multi-objective optimization: a historical view of the field. IEEE Comput Intellig Magaz 1(1):28–36
21. Coello Coello C (2011) Evolutionary multi-objective optimization: basic concepts and some applications in pattern recognition. In: Martnez-Trinidad J, Carrasco-Ochoa J, Ben-Youssef Brants C, Hancock E (eds.) Pattern recognition, vol 6718 of Lecture notes in computer science. Springer, Heidelberg, pp 22–33. doi:10.1007/978-3-642-21587-2_3
22. Corne DW, Knowles JD (2007) Techniques for highly multiobjective optimisation: some nondominated points are better than others. In: Proceedings of the 9th annual conference on genetic and evolutionary computation (New York, NY, USA, 2007), GECCO '07, ACM, pp 773–780
23. Cruz C, González JR, Pelta DA (2011) Optimization in dynamic environments: a survey on problems, methods and measures. Soft Comput 15:1427–1448
24. Das I, Dennis J (1998) Normal-boundary intersection: a new method for generating the pareto surface in non-linear multicriteria optimization problems. SIAM J Optim 8:631–657
25. Das S, Das S, Pan I (2013) Multi-objective optimization framework for networked predictive controller design. ISA Trans 52(1):56–77
26. Das S, Maity S, Qu B-Y, Suganthan P (2011) Real-parameter evolutionary multimodal optimization - a survey of the state-of-the-art. Swarm Evol Comput 1(2):71–88
27. Das S, Mullick SS, Suganthan P (2016) Recent advances in differential evolution an updated survey. Swarm Evol Comput 27:1–30
28. Das S, Suganthan PN (2010) Differential evolution: a survey of the state-of-the-art. IEEE Trans Evol Comput 99:1–28

29. Deb K (2000) An efficient constraint handling method for genetic algorithms. Comput Meth Appl Mech Eng 186(2–4):311–338
30. Deb K (2012) Advances in evolutionary multi-objective optimization. In: Fraser G, Teixeira de Souza J (eds) Search based software engineering, vol 7515 of Lecture notes in computer science. Springer, Berlin, Heidelberg, pp 1–26
31. Deb K, Pratap A, Agarwal S, Meyarivan T (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans Evol Comput 6(2):124–141
32. Deb K, Saha A (2002) Multimodal optimization using a bi-objective evolutionary algorithm. Evol Comput 27–62
33. Dorigo M, Sttzle T (2010) Ant colony optimization: overview and recent advances. In: Gendreau M, Potvin J-Y (eds) Handbook of metaheuristics, vol 146 of International series in operations research & management science. Springer US, pp 227–263
34. Efstratiadis A, Koutsoyiannis D (2010) One decade of multi-objective calibration approaches in hydrological modelling: a review. Hydrol Sci J 55(1):58–78
35. Fadaee M, Radzi M (2012) Multi-objective optimization of a stand-alone hybrid renewable energy system by using evolutionary algorithms: a review. Renew Sustain Energy Rev 16(5):3364–3369
36. Farina M, Deb K, Amato P (2004) Dynamic multiobjective optimization problems: test cases, approximations, and applications. IEEE Trans Evol Comput 8(5):425–442
37. Fazendeiro P, de Oliveira J, Pedrycz W (2007) A multiobjective design of a patient and anaesthetist-friendly neuromuscular blockade controller. IEEE Trans Biomed Eng 54(9):1667–1678
38. Fazlollahi S, Mandel P, Becker G, Maréchal F (2012) Methods for multi-objective investment and operating optimization of complex energy systems. Energy 45(1):12–22
39. Fazzolar M, Alcalá R, Nojima Y, Ishibuchi H, Herrera F (2013) A review of the application of multi-objective evolutionary fuzzy systems: current status and further directions. IEEE Trans Fuzzy Syst 21(1):45–65
40. Feng G (2006) A survey on analysis and design of model-based fuzzy control systems. IEEE Trans Fuzzy Syst 14(5):676–697
41. Figueira J, Greco S, Ehrgott M (2005) Multiple criteria decision analysis: state of the art surveys. Springer International Series
42. Fister I Jr, Yang, X-S, Brest J (2013) A comprehensive review of firefly algorithms. Swarm Evol Comput 13:34–46
43. Fleming P, Purshouse R (2002) Evolutionary algorithms in control systems engineering: a survey. Control Eng Pract 10:1223–1241
44. Fonseca C, Fleming P (1998) Multiobjective optimization and multiple constraint handling with evolutionary algorithms-I: a unified formulation. IEEE Trans Systems, Man Cybern Part A: Syst Humans 28(1):26–37
45. Fonseca C, Fleming P (1998) Multiobjective optimization and multiple constraint handling with evolutionary algorithms-II: application example. IEEE Trans Systems, Man Cybern Part A: Syst Humans 28(1):38–47
46. Gacto M, Alcalá R, Herrera F (2012) A multi-objective evolutionary algorithm for an effective tuning of fuzzy logic controllers in heating, ventilating and air conditioning systems. Appl Intell 36:330–347. doi:10.1007/s10489-010-0264-x
47. Garca JJV, Garay VG, Gordo EI, Fano FA, Sukia ML (2012) Intelligent multi-objective nonlinear model predictive control (iMO-NMPC): towards the on-line optimization of highly complex control problems. Expert Syst Appl 39(7):6527–6540
48. Gong W, Cai Z, Zhu L (2009) An efficient multiobjective. Differential Evolution algorithm for engineering design. Struct Multidisciplinary Optim 38:137–157. doi:10.1007/s00158-008-0269-9
49. Hajiloo A, Nariman-zadeh N, Moeini A (2012) Pareto optimal robust design of fractional-order PID controllers for systems with probabilistic uncertainties. Mechatronics 22(6):788–801

50. Harik G, Lobo F, Goldberg D (1999) The compact genetic algorithm. IEEE Trans Evol Comput 3(4):287–297
51. Hernández-Daz AG, Santana-Quintero LV, Coello CAC, Molina J (2007) Pareto-adaptive ϵ-dominance. Evol Comput 4:493–517
52. Herrero J, Martínez M, Sanchis J, Blasco X (2007) Well-distributed Pareto front by using the ϵ-MOGA evolutionary algorithm. In: Computational and ambient intelligence, vol LNCS 4507. Springer-Verlag, pp 292–299
53. Herreros A, Baeyens E, Perán JR (2002) Design of PID-type controllers using multiobjective genetic algorithms. ISA Trans 41(4):457–472
54. Herreros A, Baeyens E, Perán JR (2002) MRCD: a genetic algorithm for multiobjective robust control design. Eng Appl Artif Intell 15:285–301
55. Houska B, Ferreau HJ, Diehl M (2011) ACADO toolkit an open source framework for automatic control and dynamic optimization. Optim Control Appl Meth 32(3):298–312
56. Hsu C-H, Juang C-F (2013) Multi-objective continuous-ant-colony-optimized fc for robot wall-following control. Comput Intell Mag IEEE 8(3):28–40
57. Huang L, Wang N, Zhao J-H (2008) Multiobjective optimization for controller design. Acta Automatica Sinica 34(4):472–477
58. Huang V, Qin A, Deb K, Zitzler E, Suganthan P, Liang J, Preuss M, Huband S (2007) Problem definitions for performance assessment on multi-objective optimization algorithms. Nanyang Technological University, Singapore, Tech. rep
59. Hung M-H, Shu L-S, Ho S-J, Hwang S-F, Ho S-Y (2008) A novel intelligent multiobjective simulated annealing algorithm for designing robust PID controllers. IEEE Trans Syst Man Cybern Part A: Syst Humans 38(2):319–330
60. Inselberg A (1985) The plane with parallel coordinates. Visual Comput 1:69–91
61. Ishibuchi H, Tsukamoto N, Nojima Y (2008) Evolutionary many-objective optimization: a short review. In: CEC 2008. (IEEE World Congress on Computational Intelligence). IEEE Congress on Evolutionary Computation, 2008 (June 2008), pp 2419–2426
62. Jamali A, Hajiloo A, Nariman-zadeh N (2010) Reliability-based robust pareto design of linear state feedback controllers using a multi-objective uniform-diversity genetic algorithm (MUGA). Expert Syst Appl 37(1):401–413
63. Kalaivani L, Subburaj P, Iruthayarajan MW (2013) Speed control of switched reluctance motor with torque ripple reduction using non-dominated sorting genetic algorithm (nsga-ii). Int J Electr Power Energy Syst 53:69–77
64. Karaboga D, Gorkemli B, Ozturk C, Karaboga N (2012) A comprehensive survey: artificial bee colony (ABC) algorithm and applications. Artif Intell Rev 1–37
65. Kennedy J, Eberhart R (1995) Particle swarm optimization. In: Proceedings IEEE International Conference on Neural Networks, vol 4, pp 1942–1948
66. Kim H-S, Roschke PN (2006) Fuzzy control of base-isolation system using multi-objective genetic algorithm. Comput-Aided Civil Infrastruct Eng 21(6):436–449
67. Knowles J, Thiele L, Zitzler E (2014) A tutorial on the performance assessment of stochastic multiobjective optimizers. Tech. Rep. TIK report No. 214, Computer Engineering and networks laboratory. ETH Zurich, 2006
68. Kollat JB, Reed P (2007) A framework for visually interactive decision-making and design using evolutionary multi-objective optimization (VIDEO). Environ Modell Softw 22(12):1691–1704
69. Konak A, Coit DW, Smith AE (2006) Multi-objective optimization using genetic algorithms: a tutorial. Reliab Eng Syst Safety 91(9):992–1007. Special Issue - Genetic Algorithms and Reliability
70. Laumanns M, Thiele L, Deb K, Zitzler E (2002) Combining convergence and diversity in evolutionary multiobjective optimization. Evol Comput 3:263–282
71. Leiva MC, Rojas JD (2015) New tuning method for pi controllers based on pareto-optimal criterion with robustness constraint. IEEE Latin America Trans 13(2):434–440
72. Liu H, Lee S, Kim M, Shi H, Kim JT, Wasewar KL, Yoo C (2013) Multi-objective optimization of indoor air quality control and energy consumption minimization in a subway ventilation system. Energy Build 66:553–561

73. Lotov A, Miettinen K (2008) Visualizing the pareto frontier. In: Branke J, Deb K, Miettinen K, Slowinski R (eds) Multiobjective optimization, vol 5252 of Lecture notes in computer science. Springer, Heidelberg, pp 213–243
74. Lozano M, Molina D, Herrera F (2011) Soft computing: special issue on scalability of evolutionary algorithms and other metaheuristics for large-scale continuous optimization problems, vol 15. Springer-Verlag
75. Lygoe R, Cary M, Fleming P (2010) A many-objective optimisation decision-making process applied to automotive diesel engine calibration. In: Deb K, Bhattacharya A, Chakraborti N, Chakroborty P, Das S, Dutta J, Gupta S, Jain A, Aggarwal V, Branke J, Louis S, Tan K (eds) Simulated evolution and learning, vol 6457 of Lecture notes in computer science. Springer, Heidelberg, pp 638–646. doi:10.1007/978-3-642-17298-4_72
76. Mahmoodabadi M, Taherkhorsandi M, Bagheri A (2014) 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 matlabs toolbox. Scientific World J
77. Mallipeddi R, Suganthan P (2009) Problem definitions and evaluation criteria for the CEC 2010 competition on constrained real-parameter optimization. Nanyang Technological University, Singapore, Tech. rep
78. Mallipeddi R, Suganthan P (2010) Ensemble of constraint handling techniques. IEEE Trans Evol Comput 14(4):561–579
79. Mansouri SA, Gallear D, Askariazad MH (2012) Decision support for build-to-order supply chain management through multiobjective optimization. Int J Prod Econ 135(1):24–36
80. Marinaki M, Marinakis Y, Stavroulakis G (2011) Fuzzy control optimized by a multi-objective particle swarm optimization algorithm for vibration suppression of smart structures. Struct Multidisciplinary Optim 43:29–42. doi:10.1007/s00158-010-0552-4
81. Marler R, Arora J (2004) Survey of multi-objective optimization methods for engineering. Struct Multidisciplinary Optim 26:369–395
82. Martínez M, Herrero J, Sanchis J, Blasco X, García-Nieto S (2009) Applied Pareto multi-objective optimization by stochastic solvers. Eng Appl Artif Intell 22:455–465
83. Martins JRRA, Lambe AB (2013) Multidisciplinary design optimization: a survey of architectures. AIAA J 51(9):2049–2075
84. Mattson CA, Messac A (2005) Pareto frontier based concept selection under uncertainty, with visualization. Optim Eng 6:85–115
85. Meeuse F, Tousain RL (2002) Closed-loop controllability analysis of process designs: application to distillation column design. Comput Chem Eng 26(4–5):641–647
86. Messac A, Ismail-Yahaya A, Mattson C (2003) The normalized normal constraint method for generating the pareto frontier. Struct Multidisciplinary Optim 25:86–98
87. Messac A, Mattson C (2002) Generating well-distributed sets of pareto points for engineering design using physical programming. Optim Eng 3:431–450. doi:10.1023/A:1021179727569
88. Metaxiotis K, Liagkouras K (2012) Multiobjective evolutionary algorithms for portfolio management: a comprehensive literature review. Expert Syst Appl 39(14):11685–11698
89. Mezura-Montes E, Coello CAC (2011) Constraint-handling in nature-inspired numerical optimization: past, present and future. Swarm Evol Comput 1(4):173–194
90. Mezura-Montes E, Reyes-Sierra M, Coello C (2008) Multi-objective optimization using differential evolution: a survey of the state-of-the-art. Adv Differ Evol SCI 143:173–196
91. Miettinen KM (1998) Nonlinear multiobjective optimization. Kluwer Academic Publishers
92. Mininno E, Neri F, Cupertino F, Naso D (2011) Compact differential evolution. IEEE Trans Evol Comput 15(1):32–54
93. Mohan BC, Baskaran R (2012) A survey: ant colony optimization based recent research and implementation on several engineering domain. Expert Syst Appl 39(4):4618–4627
94. Molina-Cristóbal A, Griffin I, Fleming P, Owens D (2006) Linear matrix inequialities and evolutionary optimization in multiobjective control. Int J Syst Sci 37(8):513–522
95. Moscato P, Cotta C (2010) A modern introduction to memetic algorithms. In: Gendreau M, Potvin J-Y (eds) Handbook of metaheuristics, vol 146 International series in operations research & management science. Springer US, pp 141–183

96. Munro M, Aouni B (2012) Group decision makers' preferences modelling within the goal programming model: an overview and a typology. J Multi-Criteria Dec Anal 19(3–4):169–184
97. MZavala V, Flores-Tlacuahuac A (2012) Stability of multiobjective predictive control: an utopia-tracking approach. Automatica 48(10):2627–2632
98. Neri F, Cotta C (2012) Memetic algorithms and memetic computing optimization: a literature review. Swarm Evol Comput 2:1–14
99. Nikmanesh E, Hariri O, Shams H, Fasihozaman M (2016) Pareto design of load frequency control for interconnected power systems based on multi-objective uniform diversity genetic algorithm (muga). Int J Electric Power Energy Syst 80:333–346
100. Pan I, Das S (2013) Frequency domain design of fractional order PID controller for AVR system using chaotic multi-objective optimization. Int J Electric Power Energy Syst 51:106–118
101. Pan I, Das S (2015) Fractional-order load-frequency control of interconnected power systems using chaotic multi-objective optimization. Appl Soft Comput 29:328–344
102. Panda S, Yegireddy NK (2013) Automatic generation control of multi-area power system using multi-objective non-dominated sorting genetic algorithm-ii. Int J Electric Power Energy Syst 53:54–63
103. Podlubny I (1999) Fractional-order systems and pi/sup /spl lambda//d/sup /spl mu//-controllers. IEEE Trans Autom Control 44(1):208–214
104. Purshouse R, Fleming P (2007) On the evolutionary optimization of many conflicting objectives. IEEE Trans Evol Comput 11(6):770–784
105. Ramrez-Arias A, Rodrguez F, Guzmán J, Berenguel M (2012) Multiobjective hierarchical control architecture for greenhouse crop growth. Automatica 48(3):490–498
106. Rao JS, Tiwari R (2009) Design optimization of double-acting hybrid magnetic thrust bearings with control integration using multi-objective evolutionary algorithms. Mechatronics 19(6):945–964
107. Reed P, Hadka D, Herman J, Kasprzyk J, Kollat J (2013) Evolutionary multiobjective optimization in water resources: the past, present, and future. Adv Water Res 51(1):438–456
108. Reynoso-Meza G, Blasco X, Sanchis J (2009) Multi-objective design of PID controllers for the control benchmark 2008–2009 (in spanish). Revista Iberoamericana de Automática e Informática Industrial 6(4):93–103
109. Reynoso-Meza G, Blasco X, Sanchis J, Herrero JM (2013) Comparison of design concepts in multi-criteria decision-making using level diagrams. Inf Sci 221:124–141
110. Reynoso-Meza G, García-Nieto S, Sanchis J, Blasco X (2013) Controller tuning using multiobjective optimization algorithms: a global tuning framework. IEEE Trans Control Syst Technol 21(2):445–458
111. Reynoso-Meza G, Sanchis J, Blasco X, Freire RZ (2016) Evolutionary multi-objective optimisation with preferences for multivariable PI controller tuning. Expert Syst Appl 51:120–133
112. Reynoso-Meza G, Sanchis J, Blasco X, Herrero JM (2012) Multiobjective evolutionary algortihms for multivariable PI controller tuning. Expert Syst Appl 39:7895–7907
113. Reynoso-Meza G, Sanchis J, Blasco X, Martínez M (2010) Multiobjective design of continuous controllers using differential evolution and spherical pruning. In: Chio CD, Cagnoni S, Cotta C, Eber M, Ekárt A, Esparcia-Alcaráz AI, Goh CK, Merelo J, Neri F, Preuss M, Togelius J, Yannakakis GN (eds) Applications of evolutionary computation, Part I (2010) vol LNCS 6024, Springer-Verlag, pp 532–541
114. Reynoso-Meza G, Sanchis J, Blasco X, Martínez M (2016) Preference driven multi-objective optimization design procedure for industrial controller tuning. Inf Sci 339:108–131
115. Reynoso-Meza G, Sanchis J, Blasco X, Martínez M Controller tuning using evolutionary multi-objective optimisation: current trends and applications. Control Eng Pract (20XX) (Under revision)
116. Sanchis J, Martínez M, Blasco X, Salcedo JV (2008) A new perspective on multiobjective optimization by enhanced normalized normal constraint method. Struct Multidisciplinary Optim 36:537–546

117. Santana-Quintero L, Montaño A, Coello C (2010) A review of techniques for handling expensive functions in evolutionary multi-objective optimization. In: Tenne Y, Goh C-K (eds) Computational intelligence in expensive optimization problems, vol 2 of Adaptation learning and optimization. Springer, Heidelberg, pp 29–59
118. Shaikh PH, Nor NBM, Nallagownden P, Elamvazuthi I, Ibrahim T (2016) Intelligent multi-objective control and management for smart energy efficient buildings. Int J Electric Power Energy Syst 74:403–409
119. Sidhartha Panda (2011) Multi-objective PID controller tuning for a facts-based damping stabilizer using non-dominated sorting genetic algorithm-II. Int J Electr Power Energy Syst 33(7):1296–1308
120. Singh H, Isaacs A, Ray T (2011) A Pareto corner search evolutionary algorithm and dimensionality reduction in many-objective optimization problems. IEEE Trans Evol Comput 15(4):539–556
121. Snchez HS, Vilanova R (2013) Multiobjective tuning of pi controller using the nnc method: simplified problem definition and guidelines for decision making. In: 2013 IEEE 18th conference on emerging technologies factory automation (ETFA) (Sept 2013), pp 1–8
122. Srinivas M, Patnaik L (1994) Genetic algorithms: a survey. Computer 27(6):17–26
123. Srinivasan S, Ramakrishnan S (2011) Evolutionary multi objective optimization for rule mining: a review. Artif Intell Rev 36:205–248. doi:10.1007/s10462-011-9212-3
124. Stengel RF, Marrison CI (1992) Robustness of solutions to a benchmark control problem. J Guid Control Dyn 15:1060–1067
125. Stewart G, Samad T (2011) Cross-application perspectives: application and market requirements. In: Samad T, Annaswamy A (eds) The impact of control technology. IEEE Control Systems Society, pp 95–100
126. Stewart P, Gladwin D, Fleming P (2007) Multiobjective analysis for the design and control of an electromagnetic valve actuator. Proc Inst Mech Eng Part D: J Autom Eng 221:567–577
127. Stewart P, Stone D, Fleming P (2004) Design of robust fuzzy-logic control systems by multi-objective evolutionary methods with hardware in the loop. Eng Appl Artif Intell 17(3): 275–284
128. Storn R, Price K (1997) Differential evolution: a simple and efficient heuristic for global optimization over continuous spaces. J Global Optim 11:341–359
129. Sun Y, Zhang C, Gao L, Wang X (2011) Multi-objective optimization algorithms for flow shop scheduling problem: a review and prospects. Int J Adv Manuf Technol 55:723–739. doi:10.1007/s00170-010-3094-4
130. Tan W, Liu J, Fang F, Chen Y (2004) Tuning of PID controllers for boiler-turbine units. ISA Trans 43(4):571–583
131. Tavakoli S, Griffin I, Fleming P (2007) Multi-objective optimization approach to the PI tuning problem. In: Proceedings of the IEEE congress on evolutionary computation (CEC2007), pp 3165–3171
132. Vallerio M, Hufkens J, Impe JV, Logist F (2015) An interactive decision-support system for multi-objective optimization of nonlinear dynamic processes with uncertainty. Expert Syst Appl 42(21):7710–7731
133. Vallerio M, Impe JV, Logist F (2014) Tuning of NMPC controllers via multi-objective optimisation. Comput Chem Eng 61:38–50
134. Vallerio M, Vercammen D, Impe JV, Logist F (2015) Interactive NBI and (e)nnc methods for the progressive exploration of the criteria space in multi-objective optimization and optimal control. Comput Chem Eng 82:186–201
135. Vilanova R, Alfaro VM (2011) Robust PID control: an overview (in spanish). Revista Iberoamericana de Automática e Informática Industrial 8(3):141–158
136. Wie B, Bernstein DS (1992) Benchmark problems for robust control design. J Guidance Control Dyn 15:1057–1059
137. Xiong F-R, Qin Z-C, Xue Y, Schtze O, Ding Q, Sun J-Q (2014) Multi-objective optimal design of feedback controls for dynamical systems with hybrid simple cell mapping algorithm. Commun Nonlinear Sci Numer Simul 19(5):1465–1473

138. Xue Y, Li D, Gao F (2010) Multi-objective optimization and selection for the PI control of ALSTOM gasifier problem. Control Eng Pract 18(1):67–76
139. Yusup N, Zain AM, Hashim SZM (2012) Evolutionary techniques in optimizing machining parameters: review and recent applications (2007–2011). Expert Syst Appl 39(10):9909–9927
140. Zeng G-Q, Chen J, Dai Y-X, Li L-M, Zheng C-W, Chen M-R (2015) Design of fractional order PID controller for automatic regulator voltage system based on multi-objective extremal optimization. Neurocomputing 160:173–184
141. Zhang Q, Li H (2007) MOEA/D: a multiobjective evolutionary algorithm based on decomposition. IEEE Trans Evol Comput 11(6):712–731
142. Zhang Q, Zhou A, Zhao S, Suganthan P, Liu W, Tiwari S (2008) Multiobjective optimization test instances for the cec 2009 special session and competition. Tech. Rep. CES-887, University of Essex and Nanyang Technological University
143. Zhao S-Z, Iruthayarajan MW, Baskar S, Suganthan P (2011) Multi-objective robust PID controller tuning using two lbests multi-objective particle swarm optimization. Inf Sci 181(16):3323–3335
144. Zhou A, Qu B-Y, Li H, Zhao S-Z, Suganthan PN, Zhang Q (2011) Multiobjective evolutionary algorithms: a survey of the state of the art. Swarm Evol Comput 1(1):32–49
145. Zio E, Bazzo R (2011) Level diagrams analysis of pareto front for multiobjective system redundancy allocation. Reliab Eng Syst Safety 96(5):569–580
146. Zio E, Razzo R (2010) Multiobjective optimization of the inspection intervals of a nuclear safety system: a clustering-based framework for reducing the pareto front. Ann Nuclear Energy 37:798–812
147. Zitzler E, Knzli S (2004) Indicator-based selection in multiobjective search. In Yao X, Burke E, Lozano J, Smith J, Merelo-Guervós J, Bullinaria J, Rowe J, Tino P, Kabán A, Schwefel H-P (eds) Parallel problem solving from nature - PPSN VIII, vol 3242 of Lecture notes in computer science. Springer, Heidelberg, pp 832–842. doi:10.1007/978-3-540-30217-9_84
148. Zitzler E, Thiele L, Laumanns M, Fonseca C, da Fonseca V (2003) Performance assessment of multiobjective optimizers: an analysis and review. IEEE Trans Evol Comput 7(2):117–132

http://www.springer.com/978-3-319-41299-3

Controller Tuning with Evolutionary Multiobjective Optimization
A Holistic Multiobjective Optimization Design Procedure
Reynoso Meza, G.; Blasco Ferragud, X.; Sanchis Saez, J.; Herrero Durá, J.M.
2017, XIII, 227 p. 124 illus., 83 illus. in color., Hardcover
ISBN: 978-3-319-41299-3