

Review

Robust Optimization over Time Problems—Characterization and Literature Review

Pavel Novoa-Hernández ^{1,2,*} , Amilkar Puris ³  and David A. Pelta ¹ 

¹ Departamento de Ciencias de la Computación e Inteligencia Artificial, Universidad de Granada, 18071 Granada, Spain; dpelta@ugr.es

² Escuela de Ciencias Empresariales, Universidad Católica del Norte, Coquimbo 1800000, Chile

³ Facultad de Ciencias de la Ingeniería, Universidad Técnica Estatal de Quevedo, Quevedo 120550, Ecuador

* Correspondence: pavelnovoa@ugr.es

Abstract: Robust optimization over time (ROOT) is a relatively recent topic in the field of dynamic evolutionary optimization (EDO). The goal of ROOT problems is to find the optimal solution for several environments at the same time. Although significant contributions to ROOT have been published in the past, it is not clear to what extent progress has been made in terms of the type of problem addressed. In particular, we believe that there is confusion regarding what it actually means to solve a ROOT problem. To overcome these limitations, the objective of this paper is twofold. On the one hand, to provide a characterization framework of ROOT problems in terms of their most relevant features, and on the other hand, to organize existing contributions according to it. As a result, from an initial set of 186 studies, the characterization framework was applied to 35 of them, allowing to identification of some important gaps and proposing new research opportunities. We have also experimentally addressed the effect of available information on ROOT problems, concluding that there is indeed a significant impact on the performance of the algorithm and that the proposed classification is appropriate to characterize the complexity of ROOT problems. To help identify further research opportunities, we have implemented an interactive dashboard with the results of the review conducted, which is available online.

Keywords: robust optimization over time; literature review; dynamic optimization problems



Citation: Novoa-Hernández, P.; Puris, A.; Pelta, D.A. Robust Optimization over Time Problems—Characterization and Literature Review. *Electronics* **2023**, *12*, 4609. <https://doi.org/10.3390/electronics12224609>

Academic Editors: Juan M. Corchado, Carlos A. Iglesias, Byung-Gyu Kim, Rashid Mehmood, Fuji Ren and In Lee

Received: 11 October 2023

Revised: 26 October 2023

Accepted: 7 November 2023

Published: 11 November 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Several real-world optimization problems are dynamic. Some examples are the landing of a spaceship, the traffic light system of a city, and the optimal distribution of goods and services to customers that appear or disappear over time, among others. The mathematical goal in these problems is to find optimal solutions at every time step (or at every time period where no changes happen). In some cases, the complexity present in the problem makes this task difficult for traditional optimization methods. So, an effective alternative is to employ methods based on computational intelligence [1] (e.g., bio-inspired meta-heuristics), which are able to find high-quality solutions with an acceptable computational cost [2].

During the last three decades, using population-based meta-heuristics to solve dynamic optimization (DOP) problems [3,4] became an active research area, giving rise to a field known as dynamic evolutionary optimization (EDO) [5]. While some DOP variants have been studied extensively in the past, for instance tracking moving optimum (TMO), others remain somewhat unexplored. This is the case of what is known as robust optimization over time (ROOT) [6]. Unlike traditional DOPs, where the goal is to find the best solution in the current time (environment), in ROOT problems, the goal is more complex. It involves finding a solution that can be “useful” for several environments at the same time. For example, for the current and for the next n future environments. An environment is

defined as a time period where the problem does not change, and therefore, it is possible to perform optimization.

Given the wide range of possible variants of ROOT problems that can be derived from the previous description, it would be important to know the progress made in solving these problems. This would avoid “re-inventing the wheel” while some interesting research opportunities can be easily identified. To our knowledge, just a few works have focused on providing answers to this issue. In [7], the authors briefly review the literature on ROOT published up to 2019, while [8,9] surveyed the literature on evolutionary dynamic optimization up to 2021. More recently, the review of [10] organized the literature around a classification for ROOT problems based on two criteria: the requirements for changing or keeping deployed solutions and the number of deployed solutions. While these studies are useful to obtain an overview of what has and has not been carried out, we believe there is still room to make this overview more complete and comprehensive. In particular, there are aspects of ROOT problems, not addressed by these previous reviews, that can help us characterize current progress in this field.

In this context, this work has two objectives: on the one hand, to characterize ROOT problems through their most relevant features, and on the other hand, to organize the existing literature according to this characterization. Specifically, a classification framework is proposed that includes seven characteristics of ROOT problems organized in five dimensions. This framework was applied to classify the existing literature contributions, i.e., with the purpose of identifying trends and knowledge gaps that may lead to the development of future research. In addition, we have experimentally addressed the effect of one of the categories of major interest in our proposed classification: the availability of information.

The rest of the work is organized as follows. Section 2 defines the ROOT problems and explains in detail the task involved when solving a ROOT problem. Section 3 describes the proposed framework. The application of the framework in organizing current contributions on ROOT is presented in Section 4, where the experimentation carried out is also described, as well as the gaps and further research opportunities that were identified. Finally, the main conclusions of the investigation, including our future work, are outlined in Section 5.

2. Robust Optimization over Time

In general terms, a robust optimization over time problem is defined as:

$$\max_{x \in \Omega} \mathcal{R}(x, t) \quad (1)$$

where Ω is the search space and $x \in \Omega$ is a candidate solution. $t \in \mathbb{N}$ is an environment of the problem. So, function $R : \Omega \times \mathbb{N} \rightarrow \mathbb{R}$ defines the robustness of a given solution x at environment t . The task is to find the set of solutions $X^* = \{x_t^* : t = 1, 2, \dots, N\}$ such that x_t^* maximizes \mathcal{R} .

Several definitions for Equation (1) have been proposed in the past, which largely depends on what the decision maker assumes as a robust solution. A typical definition involves the aggregation of several values from specific objective functions characterizing the solution’s quality in T subsequent environments. T is commonly referred as the Time window. This approach has been used by authors like [11], in which two variants for defining \mathcal{R} were proposed. The first one, known as Average Fitness, is the average of the solution’s quality from environments in the range $\{t, \dots, t + T - 1\}$. Formally, this function is defined as [11]:

$$\mathcal{R}^a(x, t) = \frac{1}{T} \sum_{i=0}^{T-1} f_{t+i}(x) \quad (2)$$

where f_{t+i} is the objective function for the environment $t + i$.

The second variant called Survival Time, defines robustness as the number of environments in a solution that remains with quality over a certain threshold V . More formally, the Survival Time is defined as [11]:

$$\mathcal{R}^s(x, t) = \begin{cases} 0 & \text{if } f_i(x) < V, \\ 1 + \max\{l \mid \forall i \in \{t, \dots, t+l\} : f_i(x) \geq V\} & \text{otherwise.} \end{cases} \quad (3)$$

Aside from the fact that f_t can be defined in multiple forms, as we will see in the next section, \mathcal{R} can admit other forms different from the previous two.

3. Proposed Framework

A proper characterization of an optimization problem allows for a better design of a solving strategy [12]. Of course, this statement also applies to ROOT problems, which so far have not been explicitly characterized.

In this section, we propose a framework to characterize ROOT problems across five dimensions, namely, the (assumed) available information, the definition of robustness, the number of objectives, search space features (i.e., in terms of its dimension, decision variables, and feasible region) and the problem source.

This framework allows to mapping of current contributions on ROOT to highlight which types of problems have been addressed and which ones not, thus allowing researchers not only to define specific solution strategies but also to tackle problems not studied before.

3.1. Dimension 1: Available Information

The availability of information represents, from our viewpoint, the most important factor for classifying the types of problems in ROOT. It refers to the information that an algorithm is assumed to know about the environments of the problem, i.e., the environments' fitness function. Note that three types of environments are relevant: past environments, present (current) environments, and future environments, each of them with a specific fitness function. Since robustness in a given environment depends on the fitness of a subset of environments, knowing or not knowing these fitness functions is crucial to computing the robustness accurately.

Putting this issue in a real context, consider, for example, a stock market, where a decision is required at present to maximize not only the current profits but also future ones. Since, in this scenario, it is not possible to precisely know the future profits, finding the robust optimal solution to this problem is an error-prone task. Forecasting the future environments is perhaps the only effective strategy here, but that depends largely on how well the past environments have been stored or modeled [13]. Even in a case where the past environments can be exactly stored or modeled (e.g., knowing or accessing the exact definition of an environment's fitness function) and an optimized forecasting model is employed, estimating the robustness is error-prone. Finally, there is an extreme case (although somewhat unrealistic) where the fitness function from future environments can be evaluated exactly. In this case, past environments are not necessary, but this does not necessarily mean that the optimization is free of uncertainty. Since fitness functions are black box models, the optimization process is an approximation task, i.e., without any guarantee that an optimal solution can be found.

As such, uncertainty arises in ROOT problems from three major sources: the future environments, the past environments, and the black-box nature of the environments' fitness functions. In this context, we defined three ROOT scenarios according to the available information:

- Scenario 1 (S1): This is the case of ROOT problems where fitness functions from future environments can be evaluated exactly. Hence, it comprises a family of ROOT problems where the robustness of a solution can be quantified exactly. As mentioned earlier, the only source of uncertainty here is the black-box nature of the environment's fitness function. So, in this type of ROOT problem, typical TMO algorithms like [4], ref. [14] could be applied without any major adaptation. In other words, problems belonging to this scenario are slightly different from traditional dynamic optimization problems, where tracking the optimum is the main goal. To better visualize this, consider the Averaged Fitness as the robustness definition and as the fitness function guiding the optimizer during the search. Since it is an aggregation of the present and future values of objective functions that can be evaluated exactly, the problem is reduced to a TMO problem with a single, dynamic objective function, i.e., the Averaged Fitness. In the case of robustness defined in terms of Survival Time, it is not trivial to transform problems from this scenario into TMO problems. The difficulty arises because, in theory, the robustness value could be infinite. This is the case of a problem with an endless number of changes (environments) in which it is possible to find a solution that both in the current and future environments maintains a fitness above the threshold V (Equation (3)). If the number of changes has a finite, computationally tractable limit, then it would be possible to construct an objective function that captures this definition in each environment t . Note that here, instead of using an average to aggregate the information from several environments, the values of the objective function are integers. For these reasons, this scenario is suitable for studying how the optimizer part of the algorithm copes with the uncertainty of the black box fitness function.
- Scenario 2 (S2): In this scenario, we consider ROOT problems with unknown future environments, but with past environments that can be evaluated exactly. Hence, the algorithm requires a forecasting model to estimate the future fitness values. Uncertainty is present here due to the future environments and the black box definition of fitness functions. From an experimental perspective, problems in this category are suitable to study forecasting models which help to evaluate future environments.
- Scenario 3 (S3): In this type of ROOT problem, neither the future environments nor the past environments can be evaluated precisely. This implies that algorithms must include mechanisms to not only forecast the future but also to store or approximate the past effectively. It is easy to infer that, of the three scenarios, this one encloses ROOT problems with the highest level of uncertainty. In addition to the performance of the optimizer and the forecasting model, problems in this category are suitable for studying function approximation models that properly represent past environments.

Figure 1 shows the above scenarios, illustrating how they can be represented taking into account what can be evaluated exactly (past and/or future environments). See that uncertainty increases from S1 to S3, which indicates that, under similar conditions, scenario S1 is less complex than S2, while S3 is the most complex one. It is worth noting that the proposed classification is in line with that existing in other contexts such as Game Theory [15], in which environments can be of perfect information, imperfect information (asymmetric), or incomplete information. Of course, these categories would correspond to Scenarios 1, 2, and 3 of our classification, respectively. We will return to this category later, when we will analyze experimentally what effects the available information has on the performance of an algorithm.

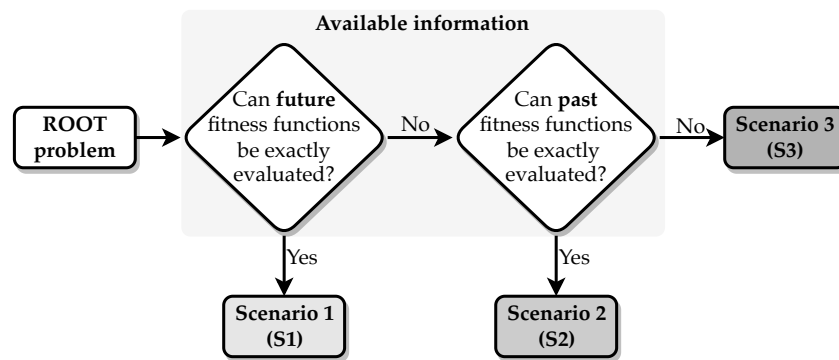


Figure 1. Possible scenarios for classifying ROOT problems according to the information available (e.g., the possibility of exactly evaluating (or not) both past and future fitness functions).

3.2. Dimension 2: Robustness Definition

Like in other optimization/decision problems, in ROOT, the decision makers play a crucial role in defining the desired kind of solution, i.e., what they consider to be a robust solution. As far as we know, this aspect has not been taken into account by existing studies on ROOT, which have been focused on finding solutions that are robust in a given period T (time window). In this context, robustness has been studied mainly in the form of Averaged Fitness and Survival Time. Real-world problems may sometimes involve different perspectives about robustness. Since robustness relies on several environments, it is not hard to consider its definition as a multi-objective optimization problem. Consequently, robustness can be defined as a weighted sum of fitness functions from T environments, or as the task of finding Pareto optimal solutions using such fitness functions. Note that, from this perspective, the Averaged Fitness definition can be viewed as a special case of the weighted version, where all environments' fitness values have the same importance. However, no research exists that explores other cases where weights can be different in each environment.

Each of these cases assumes a mathematical representation that approximates reality as closely as possible. Starting to study these types of robustness can therefore expand the application area of ROOT to real scenarios.

3.3. Dimension 3: Number of Objectives

One aspect that distinguishes optimization problems regardless of whether they are ROOT or not, is the number of objective functions to be optimized. When one seeks to optimize two or more of these objectives, then the optimal solutions that are individually optimal for each objective will not necessarily be optimal for all objectives at the same time. Consequently, the algorithm must implement some approach to deal with this issue. In this sense, the field of dynamic evolutionary multi-objective optimization [16] is a well-established field involving a large number of approaches and algorithms. In the context of ROOT, one of the main challenges is how to define the robustness of a solution from a multi-objective perspective. Concepts such as “robust Pareto-optimal solution over time” have been proposed by some authors [17]. In summary, in this dimension, we will consider two levels to classify problems: single and many to refer to single-objective and multi-objective problems, respectively.

3.4. Dimension 4: Search Space Features

Although search space features are not exclusive to ROOT problems, they are important factors that affect the complexity of the problem. Hence, they influence our selection of certain problem-solving strategies. Several search space features can be considered, for instance, ref. [12], which provides a classification of optimization problems according to eight characteristics that include: the existence of constraints, the nature of the decision variables, the physical structure of the problem, among others. However, we focused here on the following three: dimension, nature or type of decision variables, and feasible region.

Regarding the dimension of the search space, which states the number of decision variables of the problem, we consider three levels: low ($D \leq 30$), moderate ($30 < D \leq 100$) and high ($D > 100$). The reader should be aware that other levels could have been considered for this characteristic, i.e., by defining other value ranges or labels. However, for our research, we consider the levels defined above to be sufficient and in line with well-accepted problem definitions in the community [18].

In this case, the reported literature on ROOT has been only focused on problems with low dimensions. Most of the works studied problems with dimension $D = 2$. However, real-world problems may involve moderate and even higher dimensions. So, research on these problems could be of practical interest.

It should be noted that as the dimension increases, several challenges arise when solving a ROOT problem. For instance, in problems belonging to S3 not only the optimization task will be affected, but also storing or modeling the past environments.

Regarding the type of decision variables, we considered three levels: continuous, discrete, and mixed-variable. Often, discrete optimization problems are harder to solve than continuous ones. However, so far, ROOT research has focused mainly on problems with continuous decision variables, regardless of several real-world problems involving discrete [19] and mixed-variable search spaces [20].

The shape of the search space depends on the constraints present in the problem. Solutions that satisfy the problem constraints conform to the feasible region. In this regard, we considered three cases: (1) unbounded, (2) bounded, unconstrained, and (3) constrained.

Most of the works on ROOT studied problems with bounded and unconstrained feasible regions. More complex problems, such as those with functional constraints (typical of real problems), have been less addressed.

3.5. Dimension 5: Problem Source

Another important factor to consider when organizing ROOT's contributions around the problem addressed is the source of the problem. We refer to three categories: (1) real-world, (2) real-world inspired, and (3) artificial. In the first case, we include problems coming from real-world decision scenarios (e.g., based on real data). The second category groups those problems that, although based on real-life decision scenarios, use artificial data from hypothetical situations. Finally, artificial problems are those that are abstract and bear no relation to real scenarios. A typical example within this last category is the existing benchmarks for comparing algorithms.

4. Literature Review, Analysis, and Research Opportunities

This section describes the application of the proposed framework for organizing current progress and identifying potential research opportunities. To gather the current published studies on ROOT, we conducted searches on six databases, namely Scopus (<https://www.scopus.com>, accessed on 1 May 2023), ACM Digital Library (<http://portal.acm.org>, accessed on 1 May 2023), IEEE Xplore (<http://ieeexplore.ieee.org>, accessed on 1 May 2023), ScienceDirect (<http://www.sciencedirect.com>, accessed on 1 May 2023), Springer Link (<http://link.springer.com>, accessed on 1 May 2023) and ISI Web of Science (<https://www.sciencedirect.com>, accessed on 1 May 2023). Specifically, we employed the following query:

("robust optimization over time" OR "robust optimisation over time")

The search and selection methodology is illustrated in Figure 2. As can be seen, 186 studies were found in the database searches. After excluding 66 duplicate studies, the title and abstracts of the remaining 120 were screened. At this point, 81 studies were excluded for several reasons. Note that in Figure 2, the two reasons for rejection were studies that did not address ROOT problems (60) and studies whose scope was that of a literature review or survey. After this step, the remaining 39 were subjected to an in-depth review of their full text. As a result, four studies were discarded, of which three were not

related to the topic of ROOT and one was not accessible. Finally, 35 were selected and classified using the proposed framework.

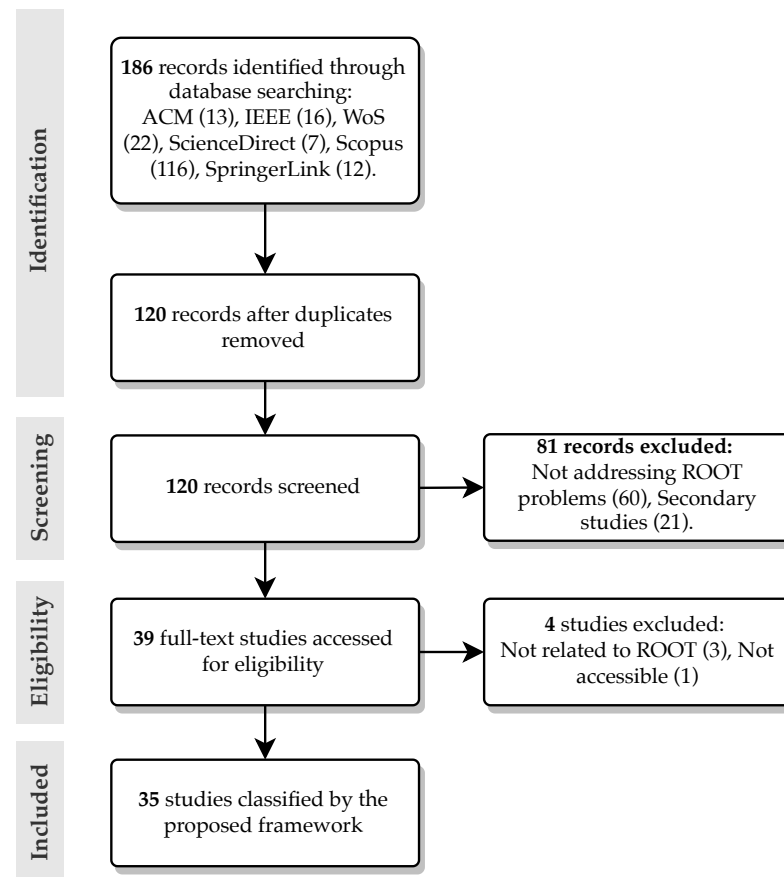


Figure 2. Literature search and selection methodology.

In what follows, we present the results obtained and the possible research opportunities derived from them.

4.1. Results

Table A1 (Appendix A) contains the 35 studies we found. Note that, in addition to the dimensions/categories proposed in our framework, we include the year of publication and the paper source (e.g., Conference or Journal). To supplement this analysis, we have also summarized these data in the graphs grouped in Figures 3 and 4. Note that these graphs describe how the number of studies is distributed according to the combination of the Available Information (y-axis) vs. the rest of the characteristics. We decided to keep Available Information as a reference in all plots because it is, in our opinion, the most relevant aspect to characterize ROOT problems. Thus, the trends and gaps that will be discussed below will be of most interest to the reader who wishes to contribute to this field. However, the reader interested in exploring combinations of different variables is free to do so through the dashboard hosted on https://modougr.shinyapps.io/root_review/, accessed on 1 November 2023.

Particularly, from Figure 3a, we can see that since 2010, interest in ROOT moderately increased over the years, with a notable tendency to address problems S1 and S3. In addition, Figure 3b shows that the number of journal articles is very similar to the number of conference papers, with the three papers exploring S3 problems being published as conference papers.

Regarding the features defined in our framework, the graph in Figure 4a shows that Survival Time is the most commonly used definition of robustness so far ($n = 19$, 54% of

studies). In contrast, other studies are based exclusively on the Average Fitness ($n = 5, 14\%$) or very specific definitions ($n = 7, 20\%$), with different formulas than Equations (2) and (3).

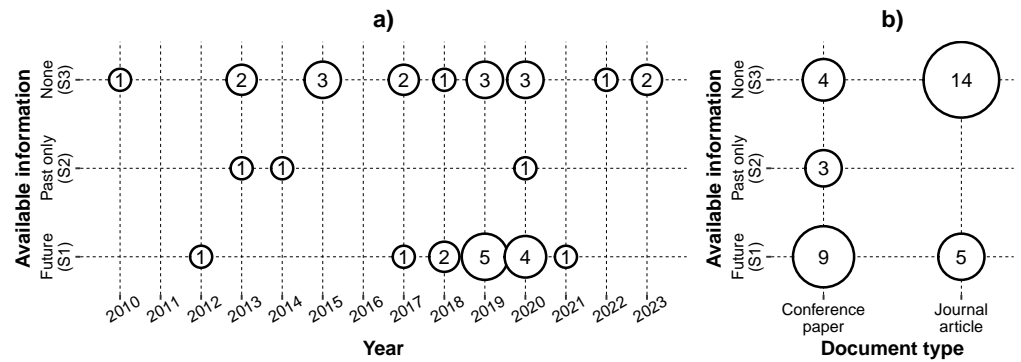


Figure 3. Distribution of the number of studies over the years (plot a) and according to the type of document (plot b).

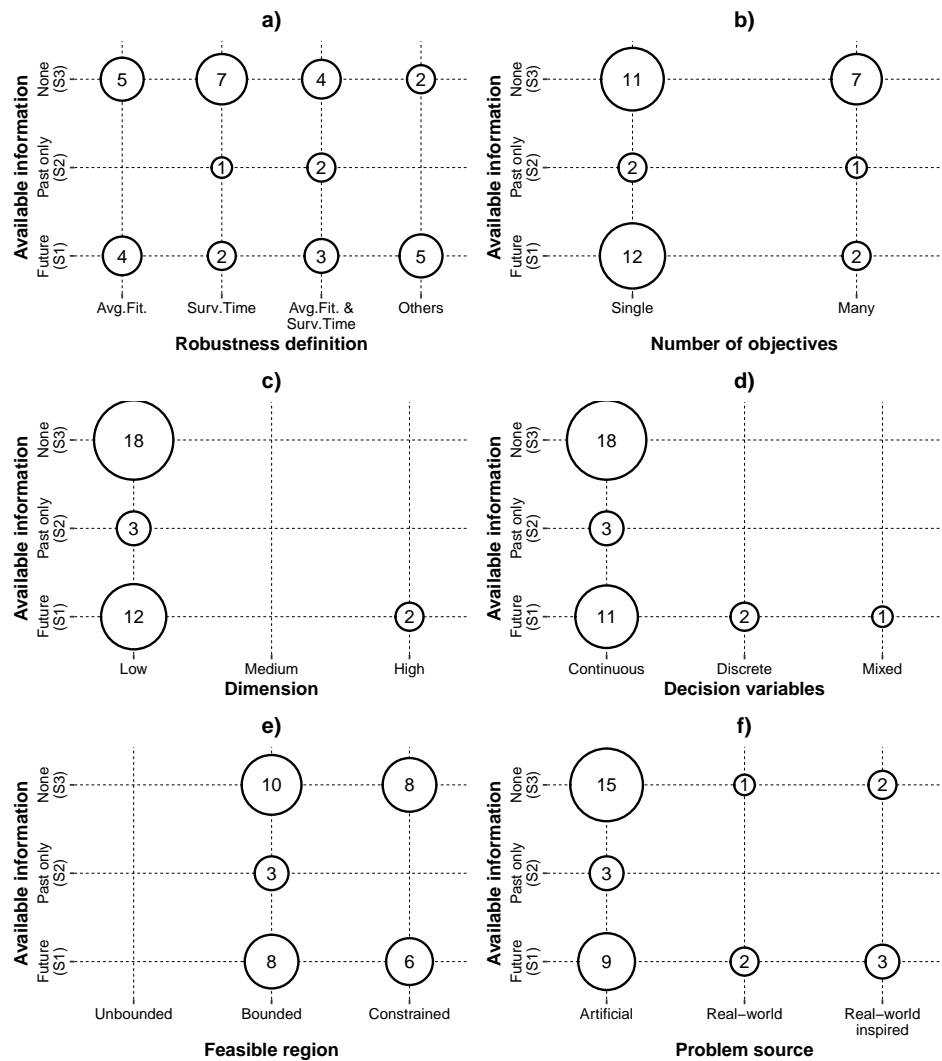


Figure 4. Distribution of the studies according to the features of the proposed framework: (a) robustness definition, (b) number of objectives, (c) search space dimension, (d) decision variables, (e) feasible region, (f) problem source.

From Figure 4b, it can be seen that most of the contributions addressed problems with only one objective function ($n = 25, 71\%$), while the rest ($n = 10, 28\%$) multi-objective problems. Within the latter group, the study by [21] stands out, in which robustness is defined in terms of two objectives: Average Fitness and Survival Time.

If we turn our attention to the dimension Search Space Features, represented by plots c, d and e in Figure 4, we can notice that the problems most frequently addressed are those of low dimension, with continuous decision variables and bounded search space. Although not shown in these graphs, 21 (60%) studies addressed problems with these characteristics.

Finally, Figure 4f most of the researchers ($n = 27, 77\%$) rely on artificial ROOT problems. This pattern is consistent with the predominance of robustness definitions based on fitness or survival time, which originated in artificial problem experimentation environments [11].

4.2. On the Effect of the Available Information

To illustrate to what extent the performance of an algorithm is affected by the information available, we have designed a small computational experiment involving the three scenarios defined in Section 3.1. To this end, we considered one of the artificial problems proposed in [11], which has the characteristics listed in Table 1. This kind of problem is known as RMPB-I [11]. Note that this is a single-objective problem, with a two-dimensional, bounded search space and continuous decision variables. In addition, the landscape of the objective function is multi-modal (with several local optima) as shown in Figure 5. This aspect is controlled by the presence of peaks for each dimension of the problem, in this case, five peaks.

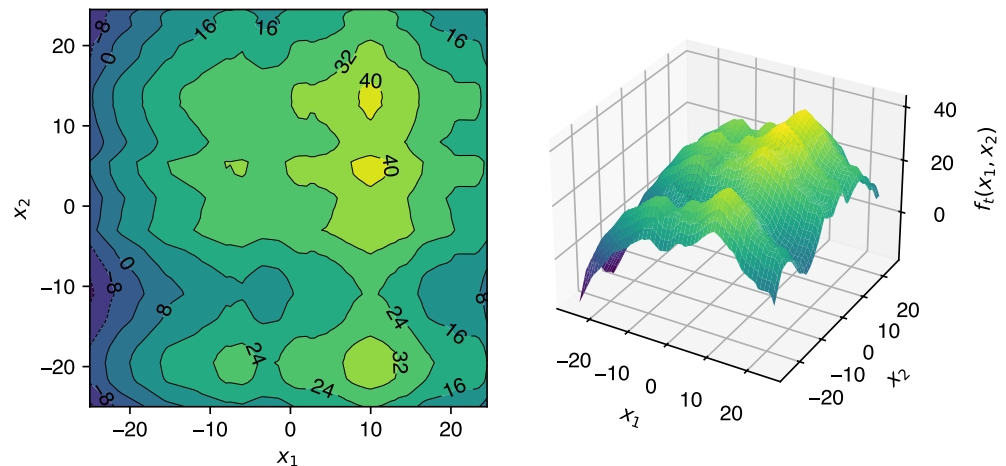


Figure 5. Landscape of the fitness function in one of the problem environments. The graph on the left shows its representation in 2D, while the one on the right is in 3D. The regions in lighter shades correspond to high values of the function, while the darker ones correspond to lower values..

Structurally, the problem can be viewed as a composition of several environments marked by time $t \in \mathbb{N}_0$. Each environment t is a stationary optimization problem defined by a specific objective function f_t , and with a common search space. As previously noted, if this problem is approached as a Tracking Moving Optima problem, the objective of the algorithm is to optimize f_t in each environment t . On the other hand, if approached as a ROOT problem, the algorithm must consider several consecutive environments at the same time to optimize robustness. In our experiment, we considered the Average Fitness (Equation (2)) as the robustness definition.

The objective function f_t in each environment t is as follows:

$$f_t(x) = \frac{1}{d} \sum_{j=1}^D \max_{i=1, \dots, m} \left\{ h_{ij}^{(t)} - w_{ij}^{(t)} \left| c_{ij}^{(t)} - x_j \right| \right\} \tag{4}$$

where i stands for the i -th peak, while j denotes the j -th dimension of the search space. Parameters c_{ij} , h_{ij} and w_{ij} are the position, height and width of peak i in dimension j , respectively.

Table 1. Parameter settings of the test problem.

Parameter	Values
Number of peaks (m)	5
Dimension (D)	2
Search space	$[-25.0, 25.0] \times [-25.0, 25.0]$
Number of changes (environments) (N)	30
Change frequency (Δe)	1000
Change type	Small step (Equation (5))
Peaks' height range (h)	$[30.0, 70.0]$
Height severity ($h_{severity}$)	5.0
Peaks' width range (w)	$[1.0, 13.0]$
Width severity ($w_{severity}$)	0.5
Rotation angle range (θ)	$[-\pi, \pi]$
Angle severity ($\theta_{severity}$)	1.0
Time window (T)	3
History size (period P)	12

In each environment t , the algorithm has a budget (change frequency) of 1000 evaluations of the objective function f_t . After consuming all of them, the problem switches to environment $t + 1$. This new environment shares the same search space as the previous one, but with a new objective function that is obtained by perturbing the parameters defining the objective function of the previous environment. Formally, the parameters w and h are perturbed as follows:

$$\phi_{t+1} = \phi_t + 0.04(\phi_{max} - \phi_{min})rand(-1, 1)\phi_{severity} \quad (5)$$

where $\phi \in \{w, h\}$ and $rand(-1, 1)$ is a rand number uniformly generated in the range $[-1, 1]$. In the case of the peak coordinates (c), they change through a rotation operator. This operator is implemented by a rotation matrix and an angle θ that is previously perturbed according to Equation (5).

Given the structure of the expression f_t , in [11], it was shown that the optimal robust solution in terms of Average Fitness can be calculated exactly. Briefly, for a given environment t the optimal solution in terms of average fitness is the solution $c_t^* = (c_1^*, c_2^*)$, where c_1^* and c_2^* are two peak centers in dimension 1 and 2, respectively. Specifically, these centers are the ones guarantying that $\mathcal{R}_t^a(c_1^*, c_2^*) \geq \mathcal{R}_t^a(c_{1i}^{(k)}, c_{2i}^{(l)})$, $\forall i, j \in \{1, \dots, 5\}$ and $\forall k, l \in \{t, \dots, T - 1\}$. In simple terms, c_t^* is one of the possible combinations of two centers of peaks from environment t and the next $T - 1$ future environments. In our problem, since we have five peaks per dimension and the time window is $T = 3$, then c_t^* is one of the 225 combinations of these centers. Consequently, we can evaluate the performance of the algorithm in terms of its error. Among the available alternatives [3,5,22], we considered here the error of the best solution before the change. Formally, this measure is defined as:

$$BEBC(t) = \left| \mathcal{R}_t^a(c_t^*) - \mathcal{R}_t^a(x_t^*) \right| \quad (6)$$

where \mathcal{R}_t^a is the robustness calculated according to Equation (2), c_t^* is the optimal solution of the problem in environment t , and x_t^* is the best solution found by the algorithm in the environment t .

We utilized Differential Evolution (DE) [23] as the solution method, which is a traditional meta-heuristic for continuous optimization problems [24]. Our instance of DE employed a population size of 50 individuals (candidate solutions) and a mutation rate

that varied randomly between 0.5 and 1.0 during the run. It also included a recombination rate of 0.7 and the Rand2Exp mutation strategy.

We assume that the algorithm can detect changes in the environment and react to the loss of diversity by restarting the search at the beginning of each environment. We have opted for this simplification to mitigate the influence that the factors involved may have on the performance of the algorithm. As a benefit, we can isolate the effect of the available information, which is the factor of interest in this experiment.

To deal with the problem of Scenario 1, where the future is known with certainty, the algorithm as described above should be sufficient. As we highlighted in Section 3.1, it would be performing a Tracking Moving Optima task. In contrast, in Scenario 2, where only the past (and present) is known accurately, the algorithm should exploit that advantage to predict the future. As suggested in [11,13], an alternative here is to apply time series models. Following [11] we opt for an auto-regressive (AR) model of order 4. So the algorithm will optimize a variant of Average Fitness in which the values of the future correspond to those predicted by the AR model based on the history of each candidate solution. More formally, let x be a candidate solution of the algorithm in environment t , and $\{f_{t-P}(x), f_{t-P+1}(x), \dots, f_{t-1}(x), f_t(x)\}$ the history of x in the P environments prior to t ; then, the approximate robustness of solution x at time t is given by:

$$\tilde{\mathcal{R}}^a(x, t) = \frac{1}{T} \left(f_t(x) + \tilde{f}_{t+1} + \dots + \tilde{f}_{t+T-1} \right) \quad (7)$$

where \tilde{f}_i are the values predicted by the AR model based on the history of x . Note that the sequence of objective function values is dependent on the parameter P , which is commonly known as the learning period.

Finally, Scenario 3 optimizes Equation (7) with a crucial variation: each solution's history incorporates surrogate models approximating the objective functions of past environments rather than their exact values. Specifically, each past environment was approximated by a Radial Basis Function (RBF) interpolation model with a Thin Plate Spline kernel. For model fitting, we randomly generated 50 search space points using a Latin hypercube sampling.

Note that in both Scenario 2 and Scenario 3, the algorithm cannot properly evaluate robustness with the expression Equation (7) until it is in an environment with a time stamp greater than or equal to P (learning period). Consequently, this creates difficulties when comparing the results with Scenario 1, where this is not required. Therefore, we opted to evaluate the algorithm's performance beginning from environment $t = 12$ for all three scenarios.

We executed 30 independent runs of the DE algorithm for each scenario. In every run, we gathered 30 values of the BEBC measurement. The summarized outcomes of these runs are displayed in the plots showcased in Figure 6. Specifically, Figure 6a depicts the evolution of BEBC performance across 30 successive environments starting from $t = 12$. Each line represents the average BEBC result of 30 runs, while the gray ribbons denote the range of values (minimum and maximum). Note that the logarithmic scale base 10 has been employed on the y-axis to aid the visual comparison of outcomes. As expected, the algorithm performs better in Scenario 1 than in the other scenarios. The mean BEBC values are mostly below 0.1. Conversely, in the other scenarios, performance substantially deteriorates, with values significantly surpassing those achieved in Scenario 1. To supplement this analysis, the distribution of BEBC values for each environment and run is summarized in a box plot shown in Figure 6b. As observed, the best performance corresponds to Scenario 1. Furthermore, it corroborates the order of difficulty in the scenarios, as described in Section 3.1, with S3 being more complex than S2, and S2 more complex than S1.

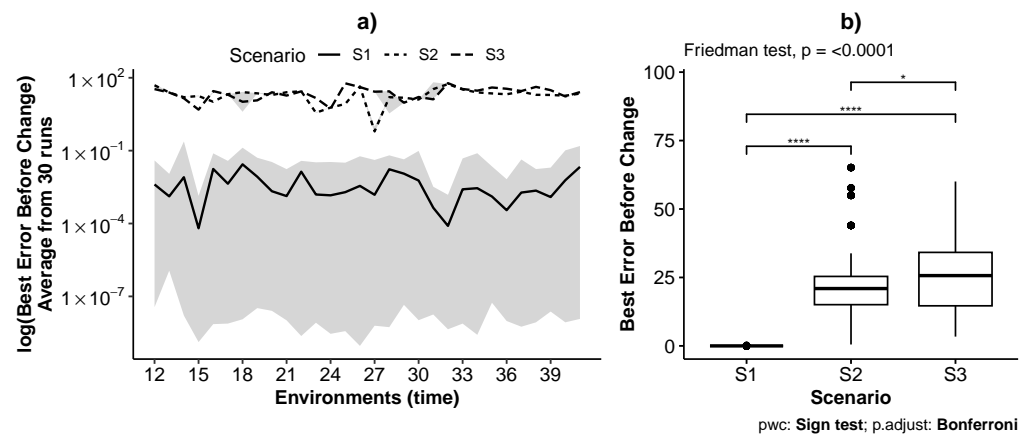


Figure 6. Performance of the algorithm in three ROOT problem scenarios. Plot (a) shows how performance changed over time, while plot (b) displays the performance distribution for each scenario and the results of the conducted statistical analyses (* p -value < 0.05, **** p -value < 0.0001).

To formally confirm these relationships, we conducted a statistical analysis to compare the algorithm's performance in each scenario. Since we controlled for both random number generation and the environment, we conducted an analysis of variance for paired samples. However, the assumptions of normality and equal variances were not met (e.g., Shapiro–Wilk and Levene tests produced p -values of less than the significance level of 0.05). Therefore, we relied on non-parametric tests. We conducted a Friedman test which yielded a p -value significantly below 0.05, indicating that there are overall differences among scenario performances. To identify the specific pairs of scenarios that have such differences, we proceeded with a post hoc analysis. Specifically, we applied a Sign test with corrections by the Bonferroni method. As a result, we found that there are significant differences among all scenarios. The box plot displayed in Figure 6b summarizes these results, with brackets indicating the adjusted p -values. See that in all cases, they were under 0.05.

Although a wide range of problems or strategies related to the issue of available information has not been addressed, some conclusions can be drawn from the performed experiment. Firstly, the algorithm's performance is significantly impacted by the available information used to compute the robustness of the solutions. Even in a low-dimensional problem where robustness is a function of only two future environments, it is challenging to achieve performance comparable to accurately evaluated future scenarios. We acknowledge that this is heavily reliant on the forecasting model's ability to capture trends and make predictions with minimal errors. In our case, the AR model may be inadequately calibrated or not suitable to the problem. One challenge in utilizing time series models is the lack of time for visually interpreting the behavior of a specific solution's sequence of values during the optimization process, which is an essential methodological recommendation for appropriately configuring model parameters. One option is to employ statistical tests to guide the selection process or to optimize the model parameters. In any case, computational complexity is added to the optimization process, since for each evaluation in the objective function that defines robustness, it is necessary to perform these operations. Similarly, when the algorithm attempts to approximate the history of the solutions (Scenario 3), it is crucial to do so accurately. This is because the error from this task may make the forecasting task more prone to errors. In other words, the more precise the approximation models are, the more reliable our predictions for future environments will be.

4.3. Research Opportunities

From the previous results, it is possible to identify some important research opportunities. Most of them are somewhat obvious since they are based on the gaps detected by

our classification. However, we have decided not to exclude them as we believe they are crucial to help advance this field of research. Specifically, we suggest:

1. Defining and/or solving problems with discrete variables, with a moderate or high number of variables, and considering feasible regions unbounded or constrained;
2. Address more problems of type S2 or S3, rather than those of type S1. Although all three categories of problems have been addressed in the past, some recent contributions focus on the simpler S1 scenario (e.g., where future fitness functions can be accurately evaluated). As was stated earlier, this type of problem is in essence the same as the Tracking Moving Optima if we defined robustness in terms of Average Fitness. In our opinion, the true ROOT problems are those derived from the assumptions defining scenarios S2 and S3;
3. Exploring alternative robustness definitions. Although alternatives to the typical ones (e.g., Average Fitness and Survival Time) have appeared in recent years, these definitions are very specific to the problem being solved. Thus, there is room to propose alternatives that can efficiently guide the algorithm in the search for robust solutions.
4. Address more real-world (inspired) problems. Except for just a few works like [25–30], the rest were focused on artificial problems. Note that addressing a larger number of real-world problems is a way to increase the number of contributions that exploit the opportunities indicated above. This is because a large number of these problems involve discrete variables, various constraints, uncertainty about the available information, and very specific definitions of robustness.

5. Conclusions and Future Works

Solving robust optimization over time problems is a recent research topic within the field of evolutionary dynamic optimization. Although some important contributions have been made in the last 10 years, there is still room for progress in this field. From our viewpoint, this topic is in its childhood if it is compared to similar topics like Tracking Moving Optima [3,5,8,9,22].

According to the proposed framework, which aimed to organize the current advances around the problems addressed so far, it can be concluded that the most important contributions have been produced in artificial problems with low-dimensional continuous search spaces. In addition, most of these contributions have considered at most two types of robustness definitions (e.g., Averaged Fitness and Survival Time). Multi-objective approaches rely on a combination of one or two of such definitions with another specific fitness function.

Problems with the above characteristics represent a very small part of real-world optimization problems. Therefore, research in scenarios with discrete and constrained search spaces, or with different definitions of robustness (e.g., weighted) would be welcome to advance the field.

One aspect that affects the performance of the algorithm in ROOT problems is the information available about future and past environments. Particularly, in scenarios where the past or future fitness functions can not be evaluated exactly, erroneous estimation of the robustness guides the algorithm towards solutions far away from the true optimal solutions. The literature reflects just a few works devoted to understanding such impact and what causes it [31,32]. We identified three possible scenarios that will contribute to answering these and further research questions.

Our future work will be aimed at exploiting the opportunities identified above and increasing the attention of the evolutionary computing community on this type of problem. In addition, we intend to extend this work by including other important factors in ROOT, such as performance measures and algorithms.

Author Contributions: Conceptualization, P.N.-H. and A.P.; methodology, D.A.P.; software, P.N.-H.; validation, D.A.P., A.P. and P.N.-H.; formal analysis, P.N.-H.; investigation, A.P.; resources, P.N.-H.; data curation, A.P.; writing—original draft preparation, P.N.-H. and A.P.; writing—review and editing, D.A.P.; visualization, P.N.-H.; supervision, D.A.P.; project administration, P.N.-H.; funding acquisition, P.N.-H. All authors have read and agreed to the published version of the manuscript.

Funding: The authors acknowledge support from project PID2020-112754GB-I00, MCIN/AEI 347/10.13039/501100011033, Ministry of Science and Innovation, Spain.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available in this article.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

Table A1. Studies classified according to the proposed framework.

Ref.	Source	Year	Scenario	Robust. Def.	N. Object.	Dimen.	Dec. Var.	Feas. Region	Prob. Source
[6]	Conference	2010	S3	A.F. ¹	Single	Low	Contin.	Bounded	Artificial
[33]	Conference	2012	S1	Others	Single	Low	Contin.	Constrain.	Artificial
[34]	Conference	2013	S2	S.T. ²	Single	Low	Contin.	Bounded	Artificial
[25]	Conference	2013	S3	Others	Single	Low	Contin.	Constrain.	Real-world inspired
[13]	Journal	2013	S3	A.F.	Single	Low	Contin.	Bounded	Artificial
[21]	Conference	2014	S2	A.F., S.T.	Many	Low	Contin.	Bounded	Artificial
[17]	Journal	2015	S3	S.T.	Many	Low	Contin.	Constrain.	Artificial
[11]	Journal	2015	S3	A.F., S.T.	Single	Low	Contin.	Bounded	Artificial
[35]	Conference	2015	S3	A.F., S.T.	Single	Low	Contin.	Bounded	Artificial
[36]	Journal	2017	S3	S.T.	Many	Low	Contin.	Constrain.	Artificial
[37]	Journal	2017	S3	S.T.	Many	Low	Contin.	Bounded	Artificial
[38]	Conference	2017	S1	A.F., S.T.	Single	Low	Contin.	Bounded	Artificial
[26]	Journal	2018	S1	Others	Many	Low	Mixed	Constrain.	Real-world inspired
[31]	Conference	2018	S3	A.F.	Single	Low	Contin.	Bounded	Artificial
[39]	Conference	2018	S1	Others	Single	Low	Contin.	Bounded	Artificial
[40]	Conference	2019	S1	A.F., S.T.	Single	Low	Contin.	Bounded	Artificial
[41]	Conference	2019	S1	A.F.	Single	High	Discrete	Constrain.	Real-world inspired
[42]	Journal	2019	S3	A.F.	Single	Low	Contin.	Bounded	Artificial
[43]	Conference	2019	S1	A.F.	Single	Low	Contin.	Bounded	Artificial
[44]	Journal	2019	S3	A.F., S.T.	Many	Low	Contin.	Constrain.	Artificial
[45]	Conference	2019	S1	A.F.	Single	Low	Contin.	Bounded	Artificial
[27]	Conference	2019	S1	Others	Single	Low	Contin.	Constrain.	Real-world
[46]	Journal	2019	S3	S.T.	Single	Low	Contin.	Bounded	Artificial
[28]	Conference	2020	S1	A.F.	Single	High	Discrete	Constrain.	Real-world inspired
[47]	Journal	2020	S1	A.F., S.T.	Single	Low	Contin.	Bounded	Artificial
[32]	Conference	2020	S2	A.F., S.T.	Single	Low	Contin.	Bounded	Artificial
[48]	Journal	2020	S1	S.T.	Single	Low	Contin.	Bounded	Artificial
[49]	Journal	2020	S3	A.F., S.T.	Many	Low	Contin.	Constrain.	Artificial
[29]	Journal	2020	S1	S.T.	Many	Low	Contin.	Bounded	Artificial
[50]	Journal	2020	S3	A.F.	Many	Low	Contin.	Constrain.	Real-world
[51]	Journal	2020	S3	Others	Single	Low	Contin.	Constrain.	Real-world inspired
[30]	Journal	2021	S1	Others	Single	Low	Contin.	Constrain.	Real-world
[52]	Journal	2022	S3	S.T.	Single	Low	Contin.	Bounded	Artificial
[53]	Journal	2023	S3	S.T.	Single	Low	Contin.	Bounded	Artificial
[54]	Journal	2023	S3	S.T.	Many	Low	Contin.	Constrain.	Artificial

¹Average Fitness, ²Survival Time.

References

1. Siddique, N.; Adeli, H. *Computational Intelligence: Synergies of Fuzzy Logic, Neural Networks and Evolutionary Computing*, 1st ed.; John Wiley & Sons, Ltd.: Chichester, UK, 2013.
2. Boussaïd, I.; Lepagnot, J.; Siarry, P. A Survey on Optimization Metaheuristics. *Inf. Sci.* **2013**, *237*, 82–117. [[CrossRef](#)]
3. Cruz, C.; González, J.R.; Pelta, D.A. Optimization in Dynamic Environments: A Survey on Problems, Methods and Measures. *Soft Comput.* **2011**, *15*, 1427–1448. [[CrossRef](#)]
4. Novoa-Hernández, P.; Corona, C.C.; Pelta, D.A. Self-adaptation in dynamic environments—A survey and open issues. *Int. J. Bio-Inspired Comput.* **2016**, *8*, 1–13. [[CrossRef](#)]
5. Nguyen, T.T.; Yang, S.; Branke, J. Evolutionary dynamic optimization: A survey of the state of the art. *Swarm Evol. Comput.* **2012**, *6*, 1–24. [[CrossRef](#)]
6. Yu, X.; Jin, Y.; Tang, K.; Yao, X. Robust optimization over time—A new perspective on dynamic optimization problems. In Proceedings of the IEEE Congress on Evolutionary Computation, Barcelona, Spain, 18–23 July 2010; pp. 1–6. [[CrossRef](#)]
7. Novoa-Hernández, P.; Puris, A. Robust optimization over time: A review of most relevant contributions. *RISTI Rev. Iber. Sist. Technol. Inf.* **2019**, *2019*, 156–164.
8. Yazdani, D.; Cheng, R.; Yazdani, D.; Branke, J.; Jin, Y.; Yao, X. A Survey of Evolutionary Continuous Dynamic Optimization Over Two Decades—Part A. *IEEE Trans. Evol. Comput.* **2021**, *25*, 609–629. [[CrossRef](#)]
9. Yazdani, D.; Cheng, R.; Yazdani, D.; Branke, J.; Jin, Y.; Yao, X. A Survey of Evolutionary Continuous Dynamic Optimization Over Two Decades—Part B. *IEEE Trans. Evol. Comput.* **2021**, *25*, 630–650. [[CrossRef](#)]
10. Yazdani, D.; Omidvar, M.N.; Yazdani, D.; Branke, J.; Nguyen, T.T.; Gandomi, A.H.; Jin, Y.; Yao, X. Robust Optimization Over Time: A Critical Review. *IEEE Trans. Evol. Comput.* **2023**, 1–21. [[CrossRef](#)]
11. Fu, H.; Sendhoff, B.; Tang, K.; Yao, X. Robust optimization over time: Problem difficulties and benchmark problems. *IEEE Trans. Evol. Comput.* **2015**, *19*, 731–745. [[CrossRef](#)]
12. Rao, S. *Engineering Optimization: Theory and Practice*, 5th ed.; John Wiley & Sons, Inc.: Hoboken, NJ, USA, 2019.
13. Jin, Y.; Tang, K.; Yu, X.; Sendhoff, B.; Yao, X. A framework for finding robust optimal solutions over time. *Memetic Comput.* **2013**, *5*, 3–18. [[CrossRef](#)]
14. Kordestani, J.K.; Ranginkaman, A.E.; Meybodi, M.R.; Novoa-Hernández, P. A novel framework for improving multi-population algorithms for dynamic optimization problems: A scheduling approach. *Swarm Evol. Comput.* **2019**, *44*, 788–805. [[CrossRef](#)]
15. Samuelson, L. Game Theory in Economics and Beyond. *J. Econ. Perspect.* **2016**, *30*, 107–130. [[CrossRef](#)]
16. Jiang, S.; Zou, J.; Yang, S.; Yao, X. Evolutionary Dynamic Multi-objective Optimisation: A Survey. *ACM Comput. Surv.* **2022**, *55*, 1–47. [[CrossRef](#)]
17. Chen, M.; Guo, Y.; Liu, H.; Wang, C. The evolutionary algorithm to find robust pareto-optimal solutions over time. *Math. Probl. Eng.* **2015**, *2015*, 1–18. [[CrossRef](#)]
18. Li, X.; Tang, K.; Omidvar, M.; Yang, Z.; Qin, K. *Benchmark Functions for the CEC'2013 Special Session and Competition on Large Scale Global Optimization*; Technical Report; Evolutionary Computation and Machine Learning Group, RMIT University: Melbourne, Australia, 2013.
19. Raggl, S.; Beham, A.; Hauder, V.; Wagner, S.; Affenzeller, M. Discrete Real-world Problems in a Black-box Optimization Benchmark. In Proceedings of the Genetic and Evolutionary Computation Conference Companion, Kyoto, Japan, 15–19 July 2018; Association for Computing Machinery: New York, NY, USA, 2018; pp. 1745–1752. [[CrossRef](#)]
20. Pelamatti, J.; Brevault, L.; Balesdent, M.; Talbi, E.G.; Guerin, Y. How to Deal with Mixed-Variable Optimization Problems: An Overview of Algorithms and Formulations. In *Advances in Structural and Multidisciplinary Optimization*; Schumacher, A., Vietor, T., Fiebig, S., Bletzinger, K.U., Maute, K., Eds.; Springer: Cham, Switzerland, 2018; pp. 64–82. [[CrossRef](#)]
21. Guo, Y.; Chen, M.; Fu, H.; Liu, Y. Find robust solutions over time by two-layer multi-objective optimization method. In Proceedings of the 2014 IEEE Congress on Evolutionary Computation (CEC), Beijing, China, 6–11 July 2014; pp. 1528–1535. [[CrossRef](#)]
22. Novoa-Hernández, P.; Corona, C.C.; Pelta, D.A. A Software Tool for Assisting Experimentation in Dynamic Environments. *Appl. Comput. Intell. Soft Comput.* **2015**, *2015*, 302172. [[CrossRef](#)]
23. Opara, K.R.; Arabas, J. Differential Evolution: A survey of theoretical analyses. *Swarm Evol. Comput.* **2019**, *44*, 546–558. [[CrossRef](#)]
24. Novoa-Hernández, P.; Corona, C.C.; Pelta, D.A. Self-adaptive, multipopulation differential evolution in dynamic environments. *Soft Comput.* **2013**, *17*, 1861–1881. [[CrossRef](#)]
25. Hutterer, S.; Affenzeller, M. Policy function approximation for optimal power flow control issues. In Proceedings of the 1st International Workshop on Simulation for Energy, Sustainable Development and Environment, SESDE 2013, Held at the International Multidisciplinary Modeling and Simulation Multiconference, I3M, Athens, Greece, 25–27 September 2013; pp. 66–70.
26. Guo, Y.N.; Cheng, J.; Luo, S.; Gong, D.; Xue, Y. Robust Dynamic Multi-Objective Vehicle Routing Optimization Method. *IEEE/ACM Trans. Comput. Biol. Bioinform.* **2018**, *15*, 1891–1903. [[CrossRef](#)]
27. Liu, Y.; Liang, H. A ROOT Approach for Stochastic Energy Management in Electric Bus Transit Center with PV and ESS. In Proceedings of the 2019 IEEE Global Communications Conference (GLOBECOM), Waikoloa, HI, USA, 9–13 December 2019; pp. 1–6. [[CrossRef](#)]
28. Ankrah, R.; Lacroix, B.; McCall, J.; Hardwick, A.; Conway, A.; Owusu, G. Racing Strategy for the Dynamic-Customer Location-Allocation Problem. In Proceedings of the 2020 IEEE Congress on Evolutionary Computation (CEC), Glasgow, UK, 19–24 July 2020; pp. 1–8. [[CrossRef](#)]

29. Huang, Y.J.; Jin, Y.C.; Hao, K.R. Robust optimization over time for constrained optimization based on swarm intelligence. *Kongzhi yu Juece/Control Decision* **2020**, *35*, 740–748. [[CrossRef](#)]
30. Liu, Y.; Liang, H. A Three-Layer Stochastic Energy Management Approach for Electric Bus Transit Centers With PV and Energy Storage Systems. *IEEE Trans. Smart Grid* **2021**, *12*, 1346–1357. [[CrossRef](#)]
31. Novoa-Hernández, P.; Pelta, D.A.; Corona, C.C. Approximation Models in Robust Optimization Over Time—An Experimental Study. In Proceedings of the 2018 IEEE Congress on Evolutionary Computation (CEC), Rio de Janeiro, Brazil, 8–13 July 2018; pp. 1–6. [[CrossRef](#)]
32. Fox, M.; Yang, S.; Caraffini, F. An Experimental Study of Prediction Methods in Robust optimization Over Time. In Proceedings of the 2020 IEEE Congress on Evolutionary Computation (CEC), Glasgow, UK, 19–24 July 2020; pp. 1–7. [[CrossRef](#)]
33. Fu, H.; Sendhoff, B.; Tang, K.; Yao, X. Characterizing environmental changes in Robust Optimization Over Time. In Proceedings of the 2012 IEEE Congress on Evolutionary Computation, CEC 2012, Brisbane, QLD, Australia, 10–15 June 2012. [[CrossRef](#)]
34. Fu, H.; Sendhoff, B.; Tang, K.; Yao, X. Finding Robust Solutions to Dynamic Optimization Problems. In *Proceedings of the Applications of Evolutionary Computation*; Lecture Notes in Computer Science; Esparcia-Alcázar, A.I., Ed.; Springer: Berlin/Heidelberg, Germany, 2013; Volume 7835, pp. 616–625. [[CrossRef](#)]
35. Huang, Y.; Jin, Y.; Ding, Y. New performance indicators for robust optimization over time. In Proceedings of the 2015 IEEE Congress on Evolutionary Computation (CEC), Sendai, Japan, 25–28 May 2015; pp. 1380–1387. [[CrossRef](#)]
36. Chen, M.R.; Guo, Y.N.; Gong, D.W.; Yang, Z. A novel dynamic multi-objective robust evolutionary optimization method. *Zidonghua Xuebao/Acta Autom. Sin.* **2017**, *43*, 2014–2032. [[CrossRef](#)]
37. Huang, Y.; Ding, Y.; Hao, K.; Jin, Y. A multi-objective approach to robust optimization over time considering switching cost. *Inf. Sci.* **2017**, *394–395*, 183–197. [[CrossRef](#)]
38. Yazdani, D.; Nguyen, T.T.; Branke, J.; Wang, J. A New Multi-swarm Particle Swarm Optimization for Robust Optimization Over Time. In *Proceedings of the Applications of Evolutionary Computation*; Squillero, G., Sim, K., Eds.; Springer: Cham, Switzerland, 2017; pp. 99–109.
39. Yazdani, D.; Branke, J.; Omidvar, M.N.; Nguyen, T.T.; Yao, X. Changing or Keeping Solutions in Dynamic Optimization Problems with Switching Costs. In Proceedings of the Genetic and Evolutionary Computation Conference, Kyoto, Japan, 15–19 July 2018; Association for Computing Machinery: New York, NY, USA, 2018; pp. 1095–1102. [[CrossRef](#)]
40. Adam, L.; Yao, X. A Simple Yet Effective Approach to Robust Optimization Over Time. In Proceedings of the 2019 IEEE Symposium Series on Computational Intelligence (SSCI), Xiamen, China, 6–9 December 2019; pp. 680–688. [[CrossRef](#)]
41. Ankrah, R.; Lacroix, B.; McCall, J.; Hardwick, A.; Conway, A. Introducing the Dynamic Customer Location-Allocation Problem. In Proceedings of the 2019 IEEE Congress on Evolutionary Computation (CEC), Wellington, New Zealand, 10–13 June 2019. [[CrossRef](#)]
42. Fei, X.; Branke, J.; Gulpinar, N. New Sampling Strategies When Searching for Robust Solutions. *IEEE Trans. Evol. Comput.* **2019**, *23*, 273–287. [[CrossRef](#)]
43. Guzman-Gaspar, J.Y.; Mezura-Montes, E. Differential Evolution Variants in Robust Optimization Over Time. In Proceedings of the 2019 International Conference on Electronics, Communications and Computers (CONIELECOMP), Cholula, Mexico, 27 February–1 March 2019. [[CrossRef](#)]
44. Guo, Y.; Yang, H.; Chen, M.; Cheng, J.; Gong, D. Ensemble prediction-based dynamic robust multi-objective optimization methods. *Swarm Evol. Comput.* **2019**, *48*, 156–171. [[CrossRef](#)]
45. Guzmán-Gaspar, J.Y.; Mezura-Montes, E. Robust Optimization Over Time with Differential Evolution using an Average Time Approach. In Proceedings of the 2019 IEEE Congress on Evolutionary Computation (CEC), Wellington, New Zealand, 10–13 June 2019; pp. 1548–1555. [[CrossRef](#)]
46. Yazdani, D.; Nguyen, T.; Branke, J. Robust Optimization over Time by Learning Problem Space Characteristics. *IEEE Trans. Evol. Comput.* **2019**, *23*, 143–155. [[CrossRef](#)]
47. Essiet, I.; Sun, Y. Tracking variable fitness landscape in dynamic multi-objective optimization using adaptive mutation and crossover operators. *IEEE Access* **2020**, *8*, 188927–188937. [[CrossRef](#)]
48. Guzmán-Gaspar, J.Y.; Mezura-Montes, E.; Domínguez-Isidro, S. Differential Evolution in Robust Optimization Over Time Using a Survival Time Approach. *Math. Comput. Appl.* **2020**, *25*, 72. [[CrossRef](#)]
49. Guo, Y.; Yang, H.; Chen, M.; Gong, D.; Cheng, S. Grid-based dynamic robust multi-objective brain storm optimization algorithm. *Soft Comput.* **2020**, *24*, 7395–7415. [[CrossRef](#)]
50. Huang, Y.; Jin, Y.; Hao, K. Decision-making and multi-objectivization for cost sensitive robust optimization over time. *Knowl.-Based Syst.* **2020**, *199*, 105857. [[CrossRef](#)]
51. Hui, Q.; Teng, Y.; Zuo, H.; Chen, Z. Reactive power multi-objective optimization for multi-terminal AC/DC interconnected power systems under wind power fluctuation. *CSEE J. Power Energy Syst.* **2020**, *6*, 630–637. [[CrossRef](#)]
52. Fox, M.; Yang, S.; Caraffini, F. A new moving peaks benchmark with attractors for dynamic evolutionary algorithms. *Swarm Evol. Comput.* **2022**, *74*, 101125. [[CrossRef](#)]

53. Yazdani, D.; Yazdani, D.; Branke, J.; Omidvar, M.N.; Gandomi, A.H.; Yao, X. Robust Optimization Over Time by Estimating Robustness of Promising Regions. *IEEE Trans. Evol. Comput.* **2023**, *27*, 657–670. [[CrossRef](#)]
54. Chen, M.; Guo, Y.; Jin, Y.; Yang, S.; Gong, D.; Yu, Z. An environment-driven hybrid evolutionary algorithm for dynamic multi-objective optimization problems. *Complex Intell. Syst.* **2023**, *9*, 659–675. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.