Les Cahiers du GERAD ISSN: 0711–2440

Micro-PRIAD: A power utility simulator for blackbox optimization benchmarking

F. Gentile, X. Lebeuf, S. Le Digabel, C. Audet, M. Diago

G–2026–45

August 2026

La publication de ces rapports de recherche est rendue possible grâce au soutien de HEC Montréal, Polytechnique Montréal, Université McGill, Université du Québec à Montréal, ainsi que du Fonds de recherche du Québec – Nature et technologies.

Dépôt légal – Bibliothèque et Archives nationales du Québec, 2026
– Bibliothèque et Archives Canada, 2026

The publication of these research reports is made possible thanks to the support of HEC Montréal, Polytechnique Montréal, McGill University, Université du Québec à Montréal, as well as the Fonds de recherche du Québec – Nature et technologies.

Legal deposit – Bibliothèque et Archives nationales du Québec, 2026
– Library and Archives Canada, 2026

GERAD HEC Montréal
3000, chemin de la Côte-Sainte-Catherine
Montréal (Québec) Canada H3T 2A7

Tél. : 514 340-6053
Téléc. : 514 340-5665
info@gerad.ca
www.gerad.ca

Micro-PRIAD: A power utility simulator for blackbox optimization benchmarking

Frédérick Gentile [a, b]

Xavier Lebeuf [a, b]

Sébastien Le Digabel [a, b]

Charles Audet [a, b]

Miguel Diago [c]

[a] *GERAD, Montréal (Qc), Canada, H3T 1J4*

[b] *Polytechnique Montréal, Montréal (Qc), Canada, H3T 0A3*

[c] *Centre de recherche d'Hydro-Québec, Varennes (Qc), Canada, J3X 1S1*

frederick.gentile@etud.polymtl.ca
xavier.lebeuf@etud.polymtl.ca
sebastien.le-digabel@polymtl.ca
charles.audet@gerad.ca
diagomartinez.miguel@hydroquebec.com

August 2026
Les Cahiers du GERAD
G–2026–45

Abstract : The field of Blackbox Optimization (BBO) lacks publicly available realistic engineering benchmark problems. This work proposes Micro-PRIAD, a collection of ten stochastic blackbox benchmark instances, of which nine are realistic industrial constrained problems. All ten are Monte Carlo (MC) simulation-based. Micro-PRIAD offers a simplified version of a complex and costly-to-evaluate asset management blackbox problem in development at Hydro-Québec (HQ). It reproduces the overall behaviour of this blackbox by simulating the application of electrical equipments maintenance strategies on power grids. For a given maintenance strategy, the simulation returns an operational cost to be minimized in an optimization context. During an evaluation of Micro-PRIAD, information is made available to a BBO solver at different key moments. Moreover, mechanisms that allow the solver to manipulate this intermediate information and interrupt the evaluation are provided. These features allow the benchmarking of solvers that will ultimately be used for difficult real-world HQ problems that possess these properties. Illustrative numerical results on Micro-PRIAD are presented to showcase complexity and its use for benchmarking.

Keywords: Benchmark problem; Monte Carlo (MC) simulation; asset management; electrical equipment maintenance; Blackbox optimization (BBO); Derivative-free optimization (DFO)

Résumé : Le domaine de l'optimisation de boîte noire (BBO) manque de problèmes d'ingénierie réalistes et disponibles au public pour l'analyse comparative d'algorithmes. Ce travail propose Micro-PRIAD, une collection de dix boîtes noires stochastiques faites pour les analyses comparatives. Neuf de ces dix instances sont des problèmes d'industrie réaliste sous contraintes. Les dix boîtes noires sont basées sur des simulations de Monte-Carlo (MC). Micro-PRIAD offre une version simplifiée d'un problème complexe et coûteux à évaluer de gestion d'actifs qui est en développement à Hydro-Québec (HQ). Elle reproduit le comportement général de cette boîte noire en simulant l'application de stratégies de maintenance des équipements électriques sur des réseaux électriques. La simulation retourne un coût d'opération relié à la stratégie de maintenance qui doit être minimisé dans un contexte d'optimisation. Au cours de l'évaluation de Micro-PRIAD, de l'information est rendu disponible aux algorithmes de BBO à différents moments clés de l'évaluation. De plus, des mécanismes permettant à ces algorithmes de manipuler l'information intermédiaire et d'interrompre l'évaluation prématurément sont intégrés à Micro-PRIAD. Ces fonctionnalités permettent l'étude d'algorithmes qui seront ultimement utilisés sur des problèmes difficiles d'application réelle de HQ qui possèdent ces propriétés. Des résultats illustratifs sur Micro-PRIAD sont présentés pour mettre en avant sa complexité et son utilité dans un contexte d'analyses comparatives.

Mots clés : problème pour analyse comparative; simulation Monte-Carlo (MC); gestion d'actifs; maintenance d'équipements électriques; Optimisation de boîte noire (BBO); optimisation sans dérivées (DFO)

1 Introduction

Blackbox optimization (BBO) refers to problems where the objective or constraint functions do not have an accessible analytical formulation. This kind of problem is often simulation-based and presents optimization challenges, such as simulation crashes, being costly to evaluate, containing local optima, and non-smoothness. Additionally, derivative-based optimization methods cannot be applied to this class of problems, where derivatives may not exist, are inaccessible, or are impractical to compute. Accordingly, derivative-free optimization (DFO) methods are used to solve this kind of problem. An introduction to BBO and DFO is found in the textbook [8].

Blackbox optimization problems are increasingly numerous and diverse, especially in the engineering domain [3]. With diverse problems come diverse optimization algorithms. In BBO and DFO, benchmark problems are used to compare optimization algorithms. By testing different algorithms under the same conditions on a collection of problems, it is possible to compare the efficiency, speed, and convergence of these algorithms [10, 15, 23].

1.1 Context and motivation

This work proposes an inequality-constrained stochastic blackbox optimization problem of the form

$$\begin{aligned} \min_{\boldsymbol{x}\in\mathcal{X}\subseteq\mathbb{R}^n} \; & f(\boldsymbol{x}) \\ \text{s.t. } \; & c_i(\boldsymbol{x}) \leq 0, i \in \{1,2,\ldots,m\}, \end{aligned} \tag{$\mathcal{P}$}$$

where

$$\mathcal{X} = \left\{\boldsymbol{x} = \{x_1, x_2, \ldots, x_n\} \in [\boldsymbol{\ell}, \boldsymbol{u}] : x_j \in \mathbb{N} \text{ if } j \in \mathcal{J}, \text{ and } x_j \in \mathbb{R} \text{ otherwise}\right\},$$

and $f : \mathcal{X} \to \overline{\mathbb{R}}$ is the objective function with $\overline{\mathbb{R}} = \mathbb{R} \cup \{\infty\}$, subject to inequality constraints. The functions f and $\boldsymbol{c}$ can take the value of ∞ if the blackbox failed to evaluate them. The m inequality constraints are quantifiable and relaxable [22], and are noted $\boldsymbol{c} : \mathcal{X} \to \overline{\mathbb{R}}^m$ where $\boldsymbol{c}(\boldsymbol{x}) = (c_1(\boldsymbol{x}), c_2(\boldsymbol{x}), \ldots, c_m(\boldsymbol{x}))$. The function f and $\boldsymbol{c}$ are, in reality, stochastic observations. The input vector $\boldsymbol{x}$ is bounded between the lower bound vector $\boldsymbol{\ell} \in \mathbb{R}^n$ and the upper bound vector $\boldsymbol{u} \in \mathbb{R}^n$. The set $\mathcal{J} \subset \{1, 2, \ldots, n\}$ indicates the integer variables.

This work introduces Micro-PRIAD, a collection of ten stochastic blackbox problem instances. Nine of these ten instances are realistic engineering applications that simulate asset management in a power grid. The tenth instance is considerably simplified compared to the nine others and is meant for fast benchmarking. The ten instances present the inherent difficulties of BBO.

The primary motivation of this work is a blackbox engineering optimization problem from the PRIAD project at Hydro-Québec. This blackbox simulates the Québec electrical transmission grid with the goal of testing and optimizing maintenance strategies of the equipment on the power grid [12, 13, 14, 18, 21]. In this document, *the PRIAD blackbox* is used as a shorthand for the blackbox problem at the core of this asset management problem. This blackbox requires up to 145 days for an evaluation with a single CPU [21]. These prohibitive costs motivate the development of benchmarking problems with similar properties. As such, Micro-PRIAD is a realistic problem that reproduces, at a smaller scale, the overall behaviour of the PRIAD blackbox. Among other changes between Micro-PRIAD and the PRIAD blackbox, the power flow simulation, which is time-consuming, is replaced by a simplified electrical cost computation function.

In Micro-PRIAD, the objective function and some of the inequality constraints are stochastic. Accordingly, their value is averaged over a certain number of Monte Carlo (MC) draws.

1.2 Literature review

There are few realistic engineering benchmarking problems in blackbox optimization. Indeed, [4, 7] states that realistic BBO application problems are often computationally expensive, have restricted access, are difficult to install, or rely on expensive or proprietary libraries. This lack of realistic engineering benchmark problems in the literature is illustrated by the largest benchmarks of DFO algorithms, where the problems used are analytical [26, 27].

Blackbox optimization works are often compared using BBO problems created for benchmarking purposes, such as the styrene production process simulation from [5], the optimization of subsystem replacement dates to avoid system failures without replacing subsystems too frequently from [11], the optimization of the design of a simplified wing from [28], the groundwater supply problem, or the hydraulic capture problem from [17]. These realistic engineering blackbox problems all offer only a single blackbox instance, whereas to evaluate different algorithms, it is advantageous to compare them across a variety of optimization instances.

Analytical test problems, such as the *cutest* problems [19], Beale, Griewank, Rosenbrock, or the problems proposed in [20], are also used for BBO algorithm benchmarking. Analytical problems offer various optimization instances that may be treated as different blackboxes. However, they severely lack the inherent difficulties encountered in real BBO applications.

The five realistic blackbox problems presented have only one instance, and the analytical problems lack the inherent difficulties encountered in BBO, these issues are common and limit the scope of use for these benchmark problems. The BBO benchmark problem collection solar [4, 7] addresses both of these gaps. It presents different optimization problems centered around the design of a concentrated solar power plant. This collection of blackboxes simulates different parts of the solar plant to create various optimization problems. The solar collection offers a variety of BBO problems implementing different problem types, including multifidelity, multiobjective, constrained, and unconstrained problems.

1.3 Contributions

This work presents Micro-PRIAD, a collection of realistic engineering BBO benchmarking problems. The contributions these problems offer are twofold. First, the gap identified in the literature is answered by providing a benchmark collection of problems. Ten instances are described herein, and the user may create their own. Second, Micro-PRIAD simulates an electrical grid where the aim is to optimize an equipment maintenance strategy such that the associated monetary cost is minimized. Micro-PRIAD offers features proper to electrical network optimization in order to benchmark solvers before applying them to real-world electrical network BBO problems. Namely, stochasticity, an embedded intermediate outputs mechanism that allows dynamic interruption of the simulator, and a division into different processing blocks.

While using the dynamic interruption mechanism, at different moments during a blackbox evaluation at $\boldsymbol{x} \in \mathcal{X}$, Micro-PRIAD provides *intermediate outputs* [1, 2] returning $f(\boldsymbol{x})$ and $\boldsymbol{c}(\boldsymbol{x})$. This allows numerical tests where the DIDS algorithm leverages intermediate information [1, 2].

In Micro-PRIAD, information is passed from one processing block to another. This allows sub-sampling of the output of the stochastic processing block before passing the information to the last deterministic block. In Micro-PRIAD, the sub-sampling saves time spent on this last processing block.

Embedded intermediate outputs and allowing the sub-sampling within the evaluation of the blackbox are two features integrated in Micro-PRIAD to mimic the behaviour of the PRIAD blackbox. Note that, to the author's knowledge, no BBO benchmarking problem currently offers these last two features.

1.4 Outline

A general overview of the blackbox and its operation is presented in Section 2, followed by an explanation of the main features of Micro-PRIAD and how to use them in Section 3. A few illustrative results of the behaviour of Micro-PRIAD and the optimization using it are presented in Section 4, followed by a short discussion in Section 5 about general subjects covered herein. Micro-PRIAD is open-source on GitHub[1], and a user guide is presented in the Appendix.

2 The Micro-PRIAD blackbox

The Micro-PRIAD blackbox simulates a power grid from electricity production down to the consumers, who will be referred to as clients. The considered clients are classified into four categories: residential clients, small businesses, large businesses, and hospitals. Micro-PRIAD simulates the main causes of failure for various electrical equipment within an electrical transmission and distribution grid. Micro-PRIAD does not include power flow simulations because they are time-consuming [21] and do not fit the benchmarking purpose.

2.1 Main blocks of Micro-PRIAD

The blackbox returns a monetary cost caused by failures, equipment maintenance, and energy not supplied (ENS) to the clients on the grid over a 40-year period. The objective is to find the maintenance periodicities associated with each degradation mechanism, for every type of equipment, that minimize this monetary cost while satisfying established constraints. These periodicities constitute a maintenance strategy that a power utility could implement to minimize operational costs.

The Micro-PRIAD blackbox is divided into several processing and simulation blocks to evaluate the objective and the constraint functions at a given $\boldsymbol{x} \in \mathcal{X}$. These functions are detailed in Section 2.5. To evaluate those functions, N MC draws are done. In Micro-PRIAD, the total N MC draws are separated into η groups of s MC draws, hence the product $N = \eta \times s$. Figure 1 illustrates the execution order of the different blocks composing Micro-PRIAD, which is called by selecting one of three power grids and giving an $\boldsymbol{x}$ of size $n \in \{13, 15, 28\}$. Other principal parameters such as N, η, s and seed are specified when calling Micro-PRIAD. The command-line options to set those parameters are presented in Section 2.4.

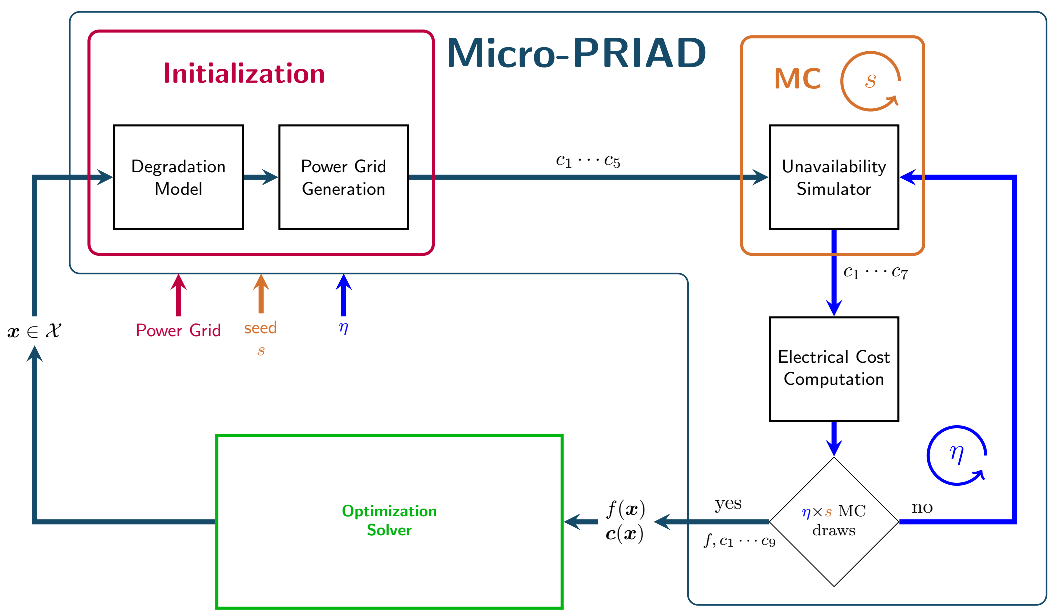


Figure 1: Internal structure of Micro-PRIAD divided into processing and simulation blocks [14]. The orange box represents the part of the blackbox subject to MC draws. MC draws are done by groups of s draws. The blue loop is repeated η times. Thus, at each call of Micro-PRIAD, a total of $N = \eta \times s$ draws are done.

[1] https://github.com/bbopt/Micro-PRIAD

The different blocks of Figure 1 are briefly described below.

Degradation Model: Computes the failure probability for each failure type of each equipment type listed in Table 1 based on the maintenance periodicities.

Power Grid Generation: Defines the power grid structure according to the power grid, which involves:

- The initialization of each client, including their type.
- The initialization of the equipment, including their maintenance periodicity, failure probability, and maintenance and repair times.
- The grouping of equipment into substations.
- The connection between clients and substations to create the links.

Unavailability Simulator: For each MC draw, it simulates unavailability periods of each electrical equipment in the grid over a 40-year period, i.e., periods representing when the equipment is non-functional due to maintenance or breakdown. It calculates the substation unavailability period over 40 years, which involves:

- The calculation of unavailability intervals due to maintenance for each equipment.
- The drawing of the unavailability intervals due to failures for each equipment.
- The aggregation of those intervals.
- The aggregation of the unavailability intervals of all equipment in each substation.
- The calculation of the unavailability period over 40 years in each substation.

Electrical Cost Computation: Calculates the cost of ENS to clients who have experienced power outages, the cost of equipment repairs, and the cost of equipment maintenance.

A more detailed description of the initialization processing blocks shown in Figure 1 is presented in Section 2.2. A more detailed version of the processing blocks of the repeated blue MC simulation, in blue in Figure 1, is presented in Section 2.3. Sections 2.2 to 2.5 present the nine realistic instances. The simplified instance is independent of those sections and is presented in Section 2.6.

2.2 Simulated power grids

Micro-PRIAD simulates small-scale power grids, representing a small town or an agglomeration of villages. The simulation of small power grids is faster, which makes the blackbox optimization problem more suitable for benchmarking. Despite the small size of the considered towns, Micro-PRIAD offers an optimization complexity true to industry blackbox problems such as the PRIAD blackbox [13], enabling the development of optimization algorithms for these blackboxes by testing them on Micro-PRIAD. This complexity is showcased in Section 4.1.

The present section first presents the generation of the power grid that uses probabilities calculated in the degradation model. The degradation model is presented after the power grid generation for easier explanation.

To generate a power grid, six parameters are used, namely

- the number of each type of client,
- the number of each type of substation,
- the number of redundant equipment for each type of equipment that can be redundant in each type of substation,
- the generation seed which controls the random number generation used in the initialization of the grid,
- the expected required time for maintenance associated with each degradation mechanism,

- the expected required time for repair associated with each degradation mechanism.

These parameters will be referred to as *power grid parameters*.

Figure 2 illustrates a generated power grid resulting from an example set of parameters. The HV, MV, and LV blocks respectively denote high-voltage, medium-voltage, and low-voltage transportation substations. A forward slash denotes a transformation substation between two voltage levels, and *prod* denotes energy production.

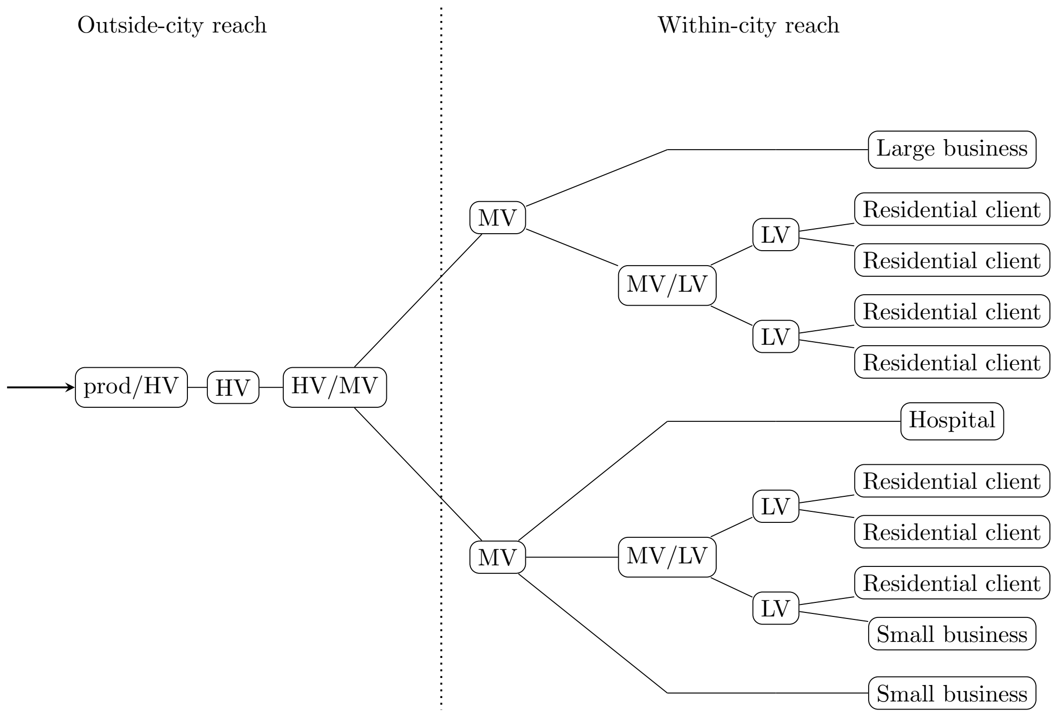


Figure 2: Illustration of an example of a simulated electrical network. On each side of the dotted line are illustrated stations outside the city reach and within the city reach.

In each substation in Figure 2, several equipments are considered, for which several degradation mechanisms are taken into account. An equipment is characterized by the substation type it appears in and by its equipment type. In each substation, multiple equipments of the same type create redundancy. For each degradation mechanism from each equipment type, the input vector $\boldsymbol{x} \in \mathcal{X}$ assigns a maintenance periodicity to all equipments of that type.

In Micro-PRIAD, substations outside the city reach affect all the clients and cause a general power outage when not functional. Conversely, the substation within the city reach affects only a portion of the clients.

Three power grids are proposed, each simulating a different power grid with a similar structure to the one illustrated in Figure 2. Each power grid is characterized by a set of power grid parameters and is labelled with a letter.

- Power grid "u" simulates the power grid of a *dense urban city*, characterized by a high density of clients, and has medium repair and maintenance time.
- Power grid "s" simulates the power grid of a *suburban city*, characterized by a medium client density and shorter repair and maintenance times compared to the dense urban city. In this power grid, all equipment that allows redundancy has a redundant piece of equipment.
- Power grid "r" simulates the power grid of a *rural town*, characterized by a much lower and more dispersed client density, and by longer maintenance and repair times than the other two

power grids. It has a high density of small businesses and no large businesses. There is also no redundancy of equipment in each substation.

Different input sizes are available, independently of the power grid parameters. Each input size characterizes a different optimization problem, as illustrated in Table 1. Micro-PRIAD thus offers a total of 9 distinct realistic engineering application blackbox optimization instances: 3 power grids, each with 3 different input sizes n.

Each blackbox decision variable is characterized by a trio of a substation type, an equipment type, and a degradation mechanism. A blackbox decision variable corresponds to a maintenance periodicity paired to that trio. Table 1 lists all trios of substation type, an equipment type, and a degradation mechanism in Micro-PRIAD.

Table 1: Enumeration of degradation mechanisms by equipment type and substation type, and description of the input vector x for the three input sizes n. [14]

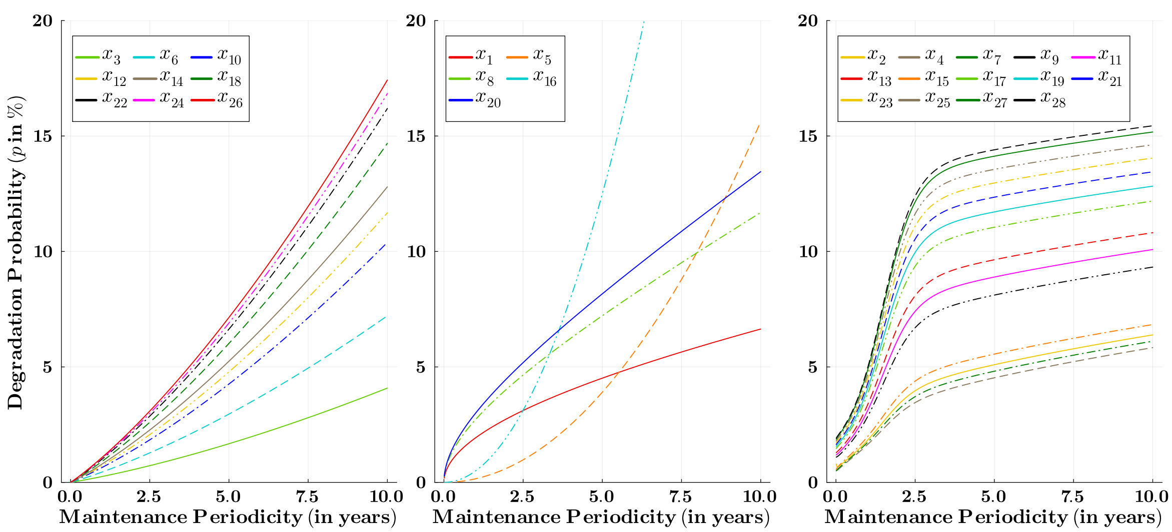
Input vector for $n =$			Nature	Station type	Equipment type	Degradation mechanism
13	15	28				
		x_1	$\mathbb{N}$	prod/HV	transformer	obstruction of electric radiators
	x_1	x_2	$\mathbb{R}$	prod/HV	transformer	theft of grounding cables
		x_3	$\mathbb{N}$	prod/HV	insulator	corrosion
	x_2	x_4	$\mathbb{R}$	prod/HV	insulator	copper theft
	x_3	x_5	$\mathbb{R}$	HV	cable	obstruction of power lines
		x_6	$\mathbb{N}$	HV	insulator	corrosion
	x_4	x_7	$\mathbb{R}$	HV	insulator	copper theft
		x_8	$\mathbb{N}$	HV/MV	transformer	obstruction of electric radiators
	x_5	x_9	$\mathbb{R}$	HV/MV	transformer	theft of grounding cables
		x_{10}	$\mathbb{N}$	HV/MV	disconnector	corrosion
	x_6	x_{11}	$\mathbb{R}$	HV/MV	disconnector	copper theft
		x_{12}	$\mathbb{N}$	HV/MV	breaker	corrosion
	x_7	x_{13}	$\mathbb{R}$	HV/MV	breaker	copper theft
		x_{14}	$\mathbb{N}$	HV/MV	insulator	corrosion
	x_8	x_{15}	$\mathbb{R}$	HV/MV	insulator	copper theft
x_1		x_{16}	$\mathbb{N}$	MV	cable	obstruction of power lines
x_2	x_9	x_{17}	$\mathbb{R}$	MV	cable	copper theft
x_3		x_{18}	$\mathbb{N}$	MV	insulator	corrosion
x_4	x_{10}	x_{19}	$\mathbb{R}$	MV	insulator	copper theft
x_5		x_{20}	$\mathbb{N}$	MV/LV	transformer	obstruction of electric radiators
x_6	x_{11}	x_{21}	$\mathbb{R}$	MV/LV	transformer	theft of grounding cables
x_7		x_{22}	$\mathbb{N}$	MV/LV	disconnector	corrosion
x_8	x_{12}	x_{23}	$\mathbb{R}$	MV/LV	disconnector	copper theft
x_9		x_{24}	$\mathbb{N}$	MV/LV	breaker	corrosion
x_{10}	x_{13}	x_{25}	$\mathbb{R}$	MV/LV	breaker	copper theft
x_{11}		x_{26}	$\mathbb{N}$	MV/LV	insulator	corrosion
x_{12}	x_{14}	x_{27}	$\mathbb{R}$	MV/LV	insulator	copper theft
x_{13}	x_{15}	x_{28}	$\mathbb{R}$	LV	cable	copper theft
Within-city reach	Partial	Complete				

To associate a maintenance periodicity with each degradation mechanism of each equipment type, two periodicity assignment systems are implemented. For each equipment type, one input variable is an absolute periodicity associated with the maintenance of a degradation mechanism, illustrated by an ($\mathbb{R}$) in Table 1. However, for each additional degradation mechanism of an equipment, an integer variable is assigned. This integer variable corresponds to the maintenance frequency relative to the first degradation mechanism on that same equipment. In Table 1, relative periodicity input variables are illustrated by an ($\mathbb{N}$). Typically, for each equipment type that has theft as a degradation mechanism, an absolute periodicity is assigned to the theft-related maintenance, and a relative periodicity is assigned to the other degradation mechanisms, which depend on the maintenance periodicity preventing theft.

For example, if the first variable is assigned 2.5 years and the second is assigned 2 years, then the equipment undergoes maintenance every 2.5 years for degradation mechanism 1 and every 5 years for degradation mechanism 2.

In problems with $n = 28$, a *complete* optimization of the maintenance strategy is done, meaning that all optimization variables are present. With $n = 15$, a *partial* optimization of the maintenance strategy is done, meaning that each relative maintenance periodicity is fixed at 2, resulting in a continuous BBO problem. This scenario represents a power utility that wants to perform complete equipment maintenance half the time and partial maintenance the other half. With $n = 13$, a *within-city reach* optimization of the maintenance strategy is done, meaning that substations outside the city reach (prod/HV, HV, and HV/MV substations) are not considered. In Figure 2, substations simulated in problems with $n = 13$ are illustrated on the right-hand side of the dotted line. This represents a scenario where maintenances of substations on the left-hand side of the dotted line in Figure 2 are done by the energy production utility; thus, the energy distribution utility only needs to take care of substations at medium and low voltage. The command-line option `-PG i` sets the power grid simulated to $i \in \{$“s”, “r”, “u”$\}$ indexing a city type listed above.

In Micro-PRIAD, the degradation model from Figure 1, which is constructed over historic data in the PRIAD blackbox [14], is modelled with analytical functions. The degradation model returns a probability of failure p associated with a degradation mechanism. It is calculated from the absolute periodicity of maintenance associated with that degradation mechanism. The analytical functions are illustrated in Figure 3.



(a) Corrosion degradation probability curves. **(b) Line obstruction and radiator obstruction degradation probability curves.** **(c) Theft-related degradation probability curves.**

Figure 3: Degradation model of Micro-PRIAD, represented with analytical curves associating an absolute maintenance periodicity to a failure probability. The degradation probability curves are separated into four categories, each modelled with a different analytical form.

2.3 Repeated Monte Carlo simulations

The stochastic nature of Micro-PRIAD is completely due to the unavailability simulator block from Figure 1. This section presents the blue loop illustrated in Figure 1. This loop repeatedly (η times) calls the block responsible of the MC simulation every time doing groups of s MC draws. This division of the total N MC draws in η groups will be exploited in Section 3 to dynamically interrupt the simulation.

For each evaluation, Micro-PRIAD performs N MC draws and returns the averages of its stochastic outputs. The values of the objective function f and the constraint functions $c_6, c_7, \ldots, c_9$ depend on the number of MC draws N and on the random seed. In contrast, the values of $c_1, c_2, \ldots, c_5$ are deterministic and are independent of both parameters.

The stochasticity of Micro-PRIAD originates from three sources: (i) equipment failures; (ii) repair durations; and (iii) maintenance durations.

For each equipment, at most one failure can occur per year. The occurrence of failures is modeled by a geometric distribution,

$$Y_1 \sim \text{Geo}(p),$$

where the parameter p is provided by the probability curves of the degradation model shown in Figure 3. Whenever a failure occurs, its occurrence time within the year is sampled from the uniform distribution

$$Y_2 \sim \mathcal{U}(0,\, 365.25 \times 24).$$

As described in Section 2, each power-grid instance specifies the expected required time for repair and maintenance for every equipment type.

These mean values are used to define lognormal distributions

$$T \sim \log\mathcal{N}(\mu, \sigma^2),$$

with variance fixed to $\mathbb{V}[T] = 0.25$ and expected value $\mathbb{E}[T]$ equal to the corresponding instance parameter [25]. The lognormal parameters μ and σ^2 can be obtained [16] from the variance and the mean value:

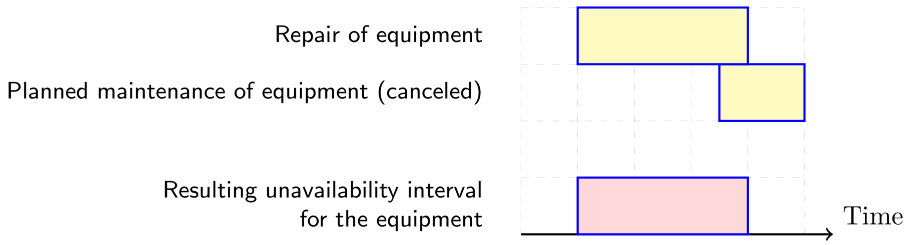
$$\sigma^2 = ln\left(\frac{\mathbb{V}[T]}{\mathbb{E}^2[T]}\right) \text{ and } \mu = ln\left(\mathbb{E}[T]\right) - \frac{\sigma^2}{2}.$$

At each MC draw, for each substation in the power grid, the Unavailability Simulator of Figure 1 draws a set of unavailability intervals from the three presented distributions. To compute the unavailability intervals of a given equipment, those due to failure and those due to maintenance are considered.

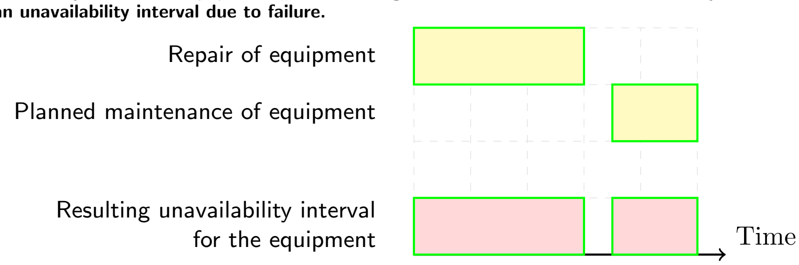
The scenario of Figure 4a illustrates a case where the planned maintenance should start while the equipment is under repair. In this case, the maintenance is not done. It is considered that the equipment has been maintained during the repair. Conversely, in the scenario of the Figure 4b, the maintenance starts after the end of the repair, and in this case, the maintenance is done even if the repairs have been done on this equipment not long ago. To compute the unavailability intervals of two redundant equipments, the unavailability intervals of those two equipments are compared to see if both equipments are unavailable at the same time. If they are unavailable at the same time, the substation where they are is experiencing a power outage. In Figure 4c, the aggregation of the resulting unavailability intervals of redundant equipment #1 and #2 is illustrated. To compute the set of unavailability intervals of a substation, the processing work done in Figure 4 is done for each equipment type present in that substation.

Figure 4 is constructed to illustrate specific scenarios. In Micro-PRIAD, the unavailability intervals of the equipment are dispersed over a longer period.

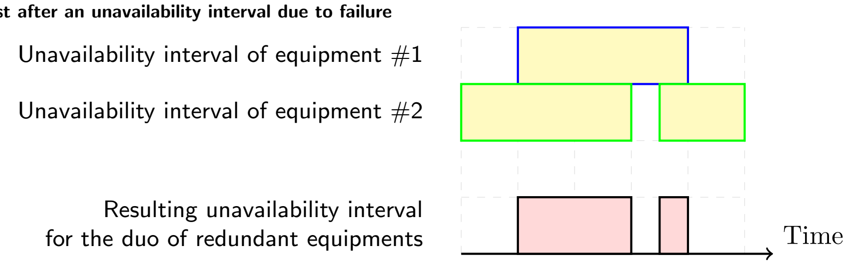
The stochastic output of the Unavailability Simulator is used to call the electrical cost simulator, which is deterministic. To avoid the computational time of the power flow simulation of the electric simulation, in Micro-PRIAD, the electrical cost computation block is simplified [14]. To compute the electrical cost for each substation, the program iterates over all clients linked to it and sums the cost of the ENS using the price listed in Table 2. It then adds to that cost the cost of corrective maintenance, which involves employee salaries, equipment replacement, and the use of a car and tools.



(a) Calculation of unavailability interval of equipment #1 illustrating the scenario where the unavailability interval due to planned maintenance occurs during an unavailability interval due to failure.



(b) Calculation of unavailability interval of equipment #2 illustrating the scenario where the unavailability interval due to planned maintenance occurs just after an unavailability interval due to failure



(c) Calculation of unavailability intervals of the duo of redundant equipment #1 and #2 represented respectively by blue contour and green contour, and illustrating the aggregation of the two resulting unavailability interval sets of equipment #1 and #2.

Figure 4: Special case handling for unavailability intervals of pieces of equipment calculated in the unavailability simulator of Figure 1, and aggregation of those unavailability intervals of redundant pieces of equipment.

Table 2: Price of undelivered energy by outage duration and client type. Large businesses have an annual consumption of more than 50MWh/year, and small businesses have an annual consumption of less than or equal to 50MWh/year [18].

Outage Duration (in hours)	**Price of Undelivered Energy (in $/kWh) for** large businesses and hospitals	small businesses	residential clients
≤ 0.5	12.70	258.00	1.30
$(0.5, 1]$	12.90	267.30	1.40
$(1, 4]$	12.10	214.30	1.60
$(4, 8]$	21.80	295.00	3.30
$(8, 16]$	37.40	474.10	5.90
> 16	190.70	2255.00	30.90

In Micro-PRIAD, the prices of Table 2 are static, and the amount of energy not supplied (ENS) to a client during a service outage is given by

$$ENS\ [kWh] = client's\ annual\ consumtion\ [kWh/year] \times outage\ duration\ [years]$$

where the annual consumption of the client is scaled in kWh instead of MWh, and the outage duration is scaled in years instead of hours.

The random seed determines all draws from the geometric, uniform, and lognormal distributions, and therefore all stochastic outputs of the unavailability simulator. The electrical cost computation block uses those stochastic outputs to compute the remaining constraints and the objective function. Consequently, the random seed determines all stochastic outputs of Micro-PRIAD.

2.4 Principal parameters of Micro-PRIAD

The principal parameters used to call Micro-PRIAD and the command-line options used to pass them to Micro-PRIAD are presented in Table 3.

Table 3: Command-line options used to specify the principal parameters of Micro-PRIAD. Two parameters among `-N`, `-eta` and `-s` are given to Micro-PRIAD; the third one is deduced from $N = \eta \times s$. If only one is given, default values of other parameters are used.

Command-line option	Default value	Admissible values	Definition of the parameter
`-PG i`	`s`	$i \in \{$"u", "s", "r", "t"$\}$	select the power grid simulated among three presented in Section 2 ("u", "s", "r") and the simplified Micro-PRIAD instance presented in Section 2.6 ("t").
`-N k`	`10000`	$k \in \mathbb{N}$	Total number of MC draws done to average f and $\boldsymbol{c}$. The option `-N 0` returns values of $c_1 ... c_5$ detailed in Section 2.5; the other constraints, as well as the objective function, are fixed at ∞.
`-s b`	`1000`	$b \in \mathbb{N}$ with $0 < b \leq N$	number of MC draws done in a group; one group is evaluated at each call of the unavailability simulator of Figure 1.
`-eta a`	`10`	$a \in \mathbb{N}$ and $0 < a \leq N$	Number of group of s MC draws done at each call of Micro-PRIAD. Set the number of iteration of the blue loop from Figure 1.
`-seed d`	`0`	$d \in \mathbb{N} \cup \{-1\}$	Random seed used to draws random variables presented in Section 2.3. When `-seed -1` is specified, the seed is generated automatically from the operating system's entropy source, producing different stochastic realizations at each execution.

In Table 3, the parameters `-N`, `-eta` and `-s` are redundant ($N = \eta \times s$). By providing two of the three parameters, the user indirectly fixes the third one. When calling Micro-PRIAD at $x \in \mathcal{X}$ without N being specified directly or indirectly with η and s, a default number N of MC draws is performed to average $f(\boldsymbol{x})$ and $\boldsymbol{c}(\boldsymbol{x})$.

To choose this number of MC draws, $f(\boldsymbol{x}_{\texttt{best}})$ is evaluated with different values of N at the best known input for each instance $\boldsymbol{x}_{\texttt{best}} \in \mathcal{X}$. For each value of N, $f(\boldsymbol{x}_{\texttt{best}})$ is evaluated a hundred times. Then, the mean value ($\bar{f}$) and the standard deviation (σ_f) of each of those hundred $f(\boldsymbol{x}_{\texttt{best}})$ values are computed.

Figure 5 illustrates the convergence of $\bar{f}$ to a stabilized value and the convergence of the standard deviation to 0 while increasing N. In blue, the value of $\bar{f}$ is plotted, and in red, the value of $\bar{f} \pm 3\sigma_f$ is plotted. Assuming that f is normally distributed, in $\approx 99.7\%$ of the cases, the value of f evaluated with an N given on the horizontal axis of Figure 5 will give a value between the two red bounds. The vertical dotted lines indicate the minimum value of N required for the ratio $(3\sigma_f/\bar{f})$ to fall below a given threshold; the thresholds shown are 1%, 0.5%, and 0.1%.

The tolerated threshold chosen for the $(3\sigma_f/\bar{f})$ ratio is 0.1%. The obtained statistics show that the minimum number of MC draws required to respect this threshold across the nine instances is $N = 10^4$. Therefore, in Micro-PRIAD, the default number of MC draws is $N = 10^4$. The statistics displayed in Figure 5 show that by fixing the number of MC draws to $N = 10^3$ or $N = 10^2$, the $(3\sigma_f/\bar{f})$ ratio respectively falls below 0.5% and 1%.

The advantage of using a lower value of N in a stochastic blackbox is to save execution time at the expense of the accuracy of the returned values. Figure 6 illustrates the time savings by showing the execution time when calling the blackbox with different values of N.

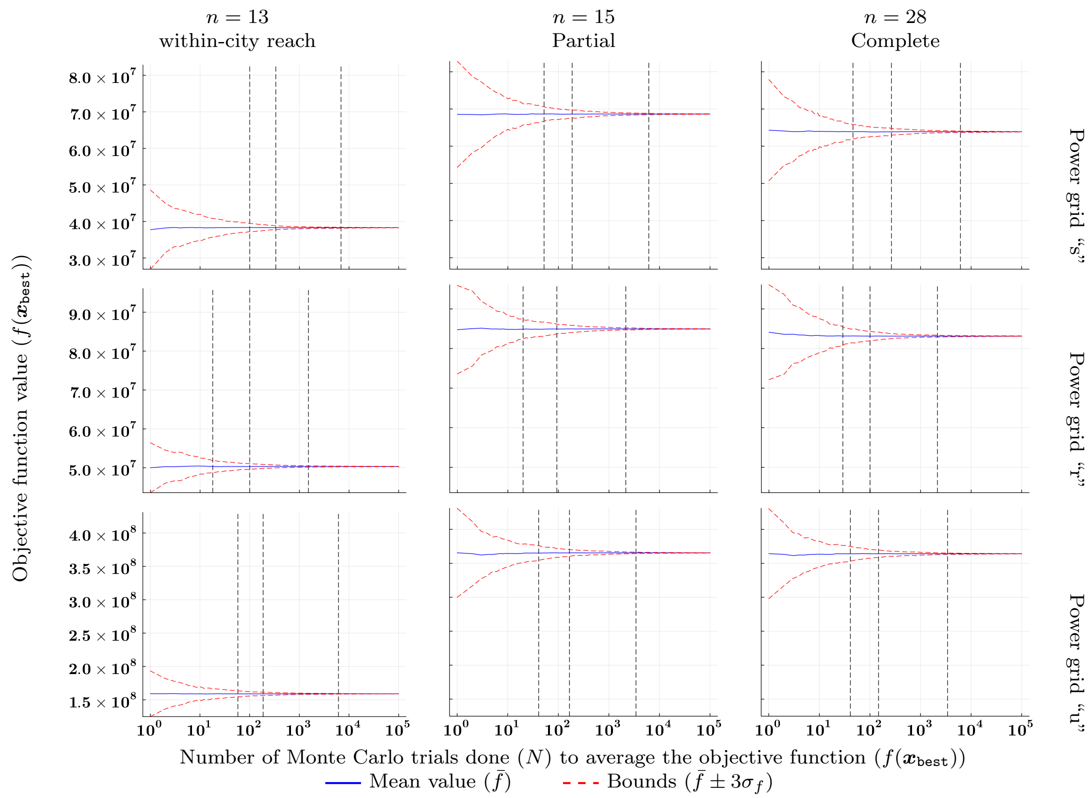


Figure 5: Horn representing the average objective function value ($\bar{f}$) over 100 samples with sizes ranging from 10 to 100 000, as well as bounds enclosing 99.7% of the mean objective function values for a given number of Monte Carlo trials ($\pm 3\sigma_f$) [24].

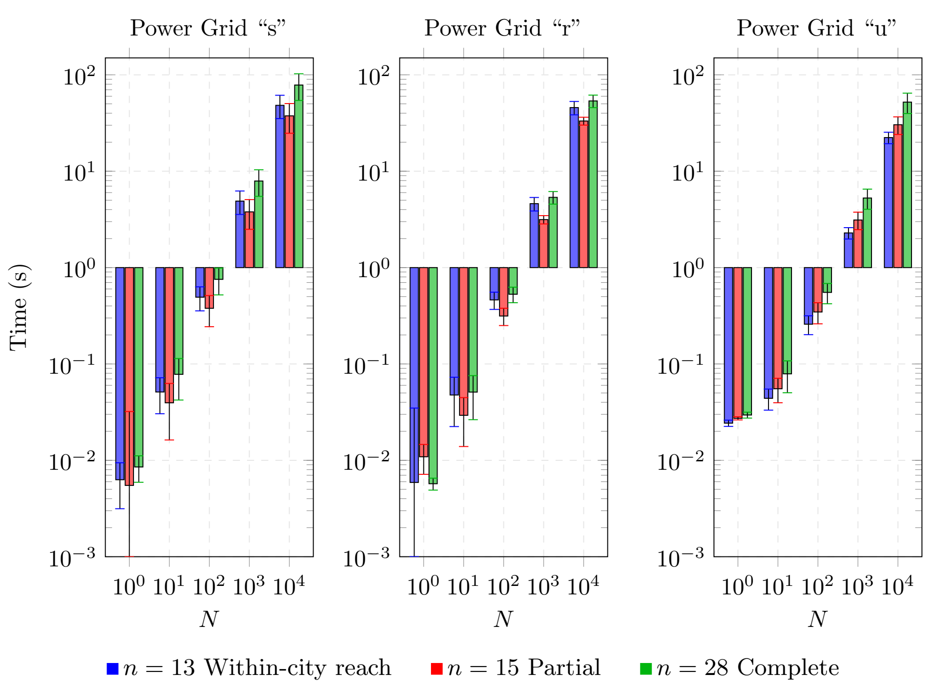


Figure 6: Mean value of the executional time and the standard deviation in seconds (s) for each instance according to the number of MC trials done.

2.5 Formulation of the optimization instances

In the optimization problem $\mathcal{P}$, the decision variables $\boldsymbol{x}$ are detailed in Table 1, and the bounds $\boldsymbol{\ell} = (\ell_1, \ell_2, \dots, \ell_n)$ and $\boldsymbol{u} = (u_1, u_2, \dots, u_n)$ are given, for all $i \in \{1, 2, \dots, n\}$, by

$$(\ell_i, u_i) = \begin{cases} (1, 9) & \text{if } i \in \mathcal{J}, \\ (0.1, 10.0) & \text{otherwise.} \end{cases}$$

The set $\mathcal{J}$ of integer variables can be deduced from the fourth column of Table 1 and is defined by

$$\mathcal{J} = \begin{cases} \{1, 3, 5, 7, 9, 11\} & \text{if } n = 13, \\ \{1, 3, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26\} & \text{if } n = 28, \\ \emptyset & \text{otherwise.} \end{cases}$$

Regardless of the problem size n, there are always $m = 9$ constraints function $\boldsymbol{c} = \{c_1, c_2, \dots, c_9\}$. They are the same in all nine instances. The Table 4 describes those constraints and classifies them with the taxonomy of constraints from [22].

Table 4: Classification according to [22] and description of the nine inequality constraints.

	Classification	Description
c_1	QRAK	A priori analytical constraint verifying the number of planned maintenance tasks performed over 40 years
c_2	QRAK	A priori analytical constraint verifying the number of planned maintenance hours performed over 40 years
c_3	QRSK	Deterministic constraint verifying the sum of the failure probabilities of each degradation mechanism
c_4	QRSK	Deterministic constraint verifying the failure probabilities of each degradation mechanism in a weighted manner to avoid probabilities greater than 5% and to ensure that nothing exceeds the 10% failure probability
c_5	QRSK	Deterministic constraint verifying the failure probabilities of each equipment type in a weighted manner to avoid probabilities greater than 5% and to ensure that nothing exceeds the 10% failure probability
c_6	QRSK	Stochastic constraint on the total service interruption time
c_7	QRSK	Stochastic constraint on the number of repair tasks over 40 years
c_8	QRSK	Stochastic constraint on the total amount of unsupplied energy over 40 years
c_9	QRSK	Stochastic constraint on the amount of unsupplied energy to hospitals

Constraints c_1 and c_2 are a priori analytical constraints [22]. Their analytical equations are defined by

$$c_1(\boldsymbol{x}) = \left(\sum_{j \in \mathcal{J}} \frac{40}{x_j \times x_{j+1}} + \sum_{i \in \{1,2,\dots,n\} \setminus \mathcal{J}} \frac{40}{x_i} \right) 1.4 \times \mathcal{K} - 500 \tag{1}$$

$$\text{and} \quad c_2(\boldsymbol{x}) = \left(\sum_{j \in \mathcal{J}} \frac{40}{x_j \times x_{j+1}} \times RTM_j + \sum_{i \in \{1,2,\dots,n\} \setminus \mathcal{J}} \frac{40}{x_i} \times RTM_i \right) 0.715 \times \mathcal{K} - 500. \tag{2}$$

Where the vector of average required time for each maintenance, denoted by RTM, is known before running the blackbox, and $\mathcal{K}$ is a power grid scaling factor that is given by

$$\mathcal{K} = \begin{cases} \frac{13}{28} & \text{if } n = 13, \\ 1 & \text{otherwise.} \end{cases}$$

Table 6 in the appendix lists the vectors of the mean time required (RTM) for each of the maintenance tasks for the three power grids. In Equations (1) and (2), the 1.4 and 0.715 are scaling factors to keep the inequality threshold at zero while providing a minimum value of -500, like for all other inequality constraints in Micro-PRIAD. In Micro-PRIAD, all constraints have a lower bound of -500 and an inequality threshold of 0, this similarity helps for the utilization of the constraints violation function [6].

2.6 The simplified Micro-PRIAD instance

Micro-PRIAD offers a considerably simplified instance compared to the nine others. It simulates only one transformation substation and is significantly faster to evaluate than the other instances. This instance is not presented above with the others because it has restricted functionality and is meant for fast benchmarking. It is unconstrained ($m = 0$), only includes the intermediate outputs related to equipment availability, and does not loop, meaning that $\eta = 1$ is fixed. For each step of the Micro-PRIAD simulation, each processing block is replaced with a simplified version. It is a stochastic simulation where the outputs of the unavailability simulator can be sub-sampled before calling the electrical cost computation block. To use this instance, the command-line option `-PG t` sets the power grid to a *single transformation substation*.

In the optimization problem $\mathcal{P}$, the decision variable $\boldsymbol{x}$, its lower and upper bounds $\boldsymbol{\ell}$ and $\boldsymbol{u}$ are described in Table 5. In this instance, all decision variables are continuous, meaning that all variables are absolute maintenance periodicities. This additional instance does not have any constraints; hence $m = 0$.

Table 5: Formulation of the additional problem instance.

$\boldsymbol{x}$	Nature	$\boldsymbol{\ell}$	$\boldsymbol{u}$	Equipment type	Degradation mechanism
x_1	$\mathbb{R}$	0.0	10.0	breaker	corrosion
x_2	$\mathbb{R}$	0.0	10.0	transformer	obstruction of electric radiators
x_3	$\mathbb{R}$	0.0	10.0	transformer	theft of grounding cables
x_4	$\mathbb{R}$	0.0	10.0	disconnector	corrosion

As illustrated in Table 5, this instance always has an $\boldsymbol{x}$ of length 4 and has different but similar variables than the other instances.

3 Time-saving features of the Micro-PRIAD blackbox

Micro-PRIAD is a stochastic blackbox function. It incorporates mechanisms specifically designed to handle stochasticity, notably through its separation into several processing blocks and the use of intermediate outputs. These features are interconnected, and their objective is to allow a BBO solver to reduce blackbox evaluation times during an optimization. The separation into different processing blocks provides access to the outputs of the unavailability simulator, which constitutes the stochastic component of Micro-PRIAD. Those outputs are sent to the *subsampler*, which decides what to send to the electrical cost computation block. The recommended usage of this process is to take the original sample of size s MC draws and return a sub-sample of size $\hat{s} < s$ MC draws. Intermediate outputs are returns of information to the solver before the blackbox evaluation is completed, thereby allowing the simulation to be dynamically interrupted before the total N MC draws are completed.

Micro-PRIAD calls up to three different interrupters during the simulation. Each interrupter has the power to terminate the simulation. In Micro-PRIAD, the default interrupters do not stop the simulation, making the interrupter invisible to the eyes of the solver as shown in Figure 1. The Energy interrupter has one mandatory condition for the interruption, which is the completion of the total N MC draws.

No sub-sampling is performed by default, but it allows the solver to add a subsampler. It allows the sub-sampling of the s MC draws passed from the unavailability simulator to the electrical cost computation block.

Micro-PRIAD thus introduces two mechanisms that the solver can use as levers for the optimization. Figure 7 builds upon Figure 1. It illustrates the internal structure of Micro-PRIAD and the interaction between the processing block of Micro-PRIAD and the different levers in green.

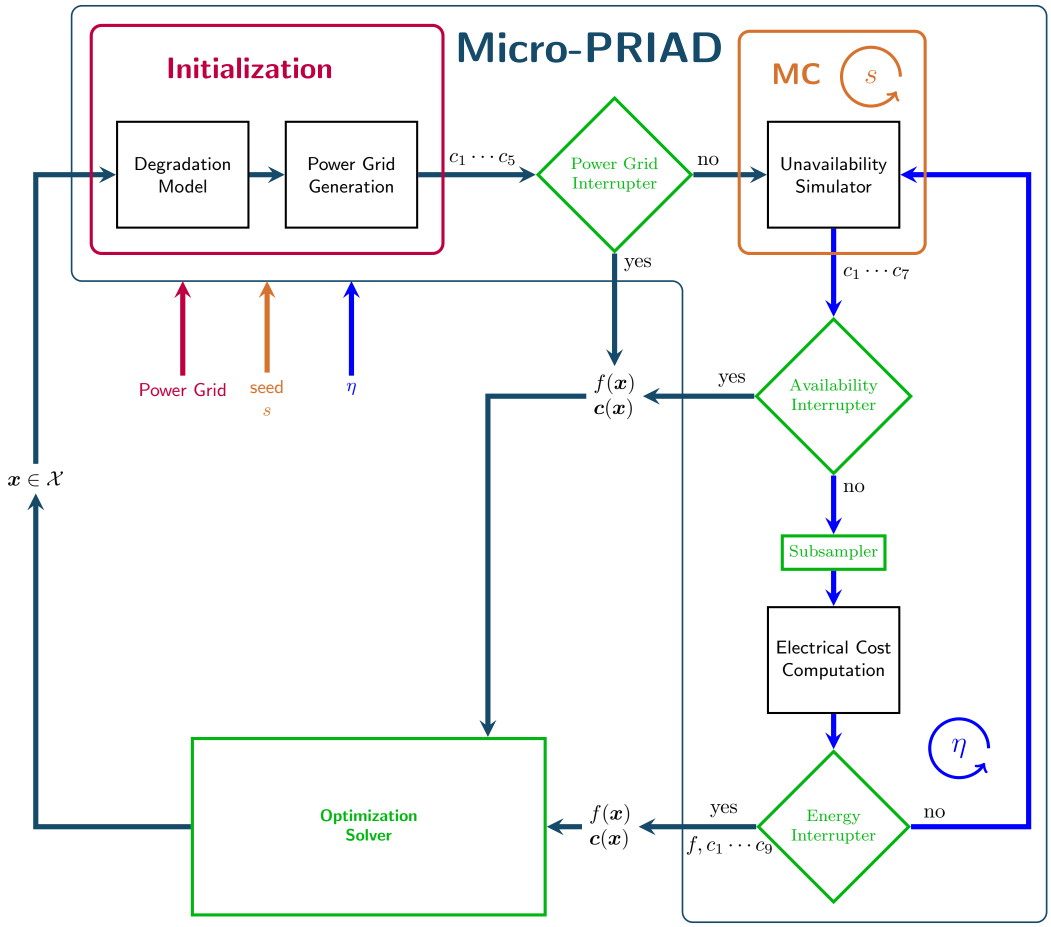


Figure 7: Internal structure of Micro-PRIAD and interaction between processing blocks and optimization levers in green [14]. The Unavailability processing block does s MC draws before passing information to the interrupter. The blue loop composed of the unavailability simulator and the electrical cost computation block is repeated $\eta = N/s$ times.

3.1 Intermediate outputs

Micro-PRIAD implements intermediate outputs and allows for the dynamic interruption of the blackbox at certain key moments. Solvers utilize controllers called *dynamic interrupters* that make the decision, based on the intermediate outputs provided by Micro-PRIAD, to either continue the evaluation or interrupt it without proceeding with the remainder of the evaluation [1, 2]. The dynamic interrupters are part of the solver and are provided by the user.

In Figure 7, the green diamonds and boxes are user-defined and can communicate with one another. In Micro-PRIAD, by default, the intermediate outputs are disabled, and no dynamic interruptions are performed.

In Micro-PRIAD, the intermediate outputs occur at three different places in the blackbox illustrated by green diamonds in Figure 7. The first intermediate output, namely the one returned to the power grid interrupter, returns the value of the deterministic constraints $(c_1, c_2, \dots, c_5)$ and fixes the values of the other constraints and the objective function to ∞. The second and third places where intermediate outputs are returned are at the availability interrupter and the energy interrupter. An intermediate output is returned to those two interrupters at each of the η cycles of the blue loop of Figure 7. The intermediate output provided to the availability interrupter is composed of the deterministic constraints $(c_1, c_2, \dots, c_5)$, the updated values of c_6 and c_7, and the values of f, c_8 and c_9 at the last iteration of the blue loop. The values of f, c_8 and c_9 are computed in the electrical cost computation block. The first time the availability interrupter receives an intermediate output, the electrical cost computation block has never been called; at this return, the values of f, c_8 and c_9 are fixed at ∞. The intermediate output provided to the energy interrupter is composed of the up-to-date values of f and $\boldsymbol{c}$.

To indicate how many MC draws have been completed at a certain moment, two notations are introduced: (i) the iteration counter, which counts the number of times the blue loop has been started; it is denoted by $i \in \{0, 1, \ldots, \eta\}$; and (ii) the number of MC draws completed at iteration i; denoted N_i. Hence, the equality below gives the value of N_i:

$$N_i = i \times s.$$

The counter i can also be interpreted as the number of times that the unavailability simulator block has been called.

The intermediate outputs are composed of two vectors, one with the values of f and $\boldsymbol{c}$, and the other with the respective number of MC draws done to obtain their value. Even if the deterministic constraints do not depend on N_i, their corresponding value in the second vector is fixed at N_i. The corresponding values of c_6 and c_7 in the second vector are also fixed at N_i. The corresponding values of f, c_8 and c_9 in the second vector are either fixed at N_{i-1} or at N_i respectively in the intermediate output sent to the availability interval and sent to the energy interrupter. Indeed, at the availability interrupter, the electrical cost computation block has not been called in the present iteration of the blue loop; the values of f, c_8 and c_9 are those of the last iteration; hence the corresponding values of f, c_8 and c_9 in the second vector are fixed at N_{i-1}.

3.2 Sub-sampling of the MC draws

Micro-PRIAD is divided into four different processing and simulation blocks. This division gives access to the output of the unavailability simulator block, which is the stochastic MC simulation part of the blackbox. The outputs of the block are used to call the electrical cost computation block. Those outputs can be sub-sampled before being passed to the electrical cost computation block.

By sub-sampling the outputs of the unavailability simulator block, the number of inputs of the electrical cost computation block is reduced, leading to a gain of time when calling Micro-PRIAD. In the case of Micro-PRIAD, this gain is negligible because the electrical cost computation block is less costly to evaluate than the unavailability simulator block. Micro-PRIAD offers this feature to mimic the behaviour of the PRIAD blackbox, which has an electrical cost computation block that is extremely costly to evaluate. Micro-PRIAD thus allows benchmarking of optimization solvers interested in this kind of problem.

The unavailability simulator produces a list of vectors indicating unavailability periods of each substation over 40 years. Before returning its outputs, it transforms this list into a vector containing the total number of hours that each substation will be offline over 40 years, for each of the six outage duration periods described in the first column of Table 2. As a result, for power grids “s” and “u”, the input of the electrical cost computation block is a vector of length 66, and for power grid “r”, it is of length 180. For the simplified Micro-PRIAD instance, this vector is of length 4.

At each call of the unavailability simulator block, s vectors of length 66, 180 or 4 are produced to call the electrical cost computation block. At each evaluation of the Micro-PRIAD blackbox, those blocks are called η times.

After the electrical cost computation block, the values of the stochastic outputs are always averaged over the number of MC draws used to evaluate each of them. In the case where sub-sampling is used, the values of f, c_8, and c_9 are averaged over $\hat{N}_i = i \times \hat{s}$, where $\hat{s}$ is the size of the sub-sample, and the values of c_6 and c_7 are averaged over N_i. The values of $c_1, c_2, \ldots, c_5$ are deterministic and are not influenced by the sub-sampling of the stochastic outputs of the unavailability simulator.

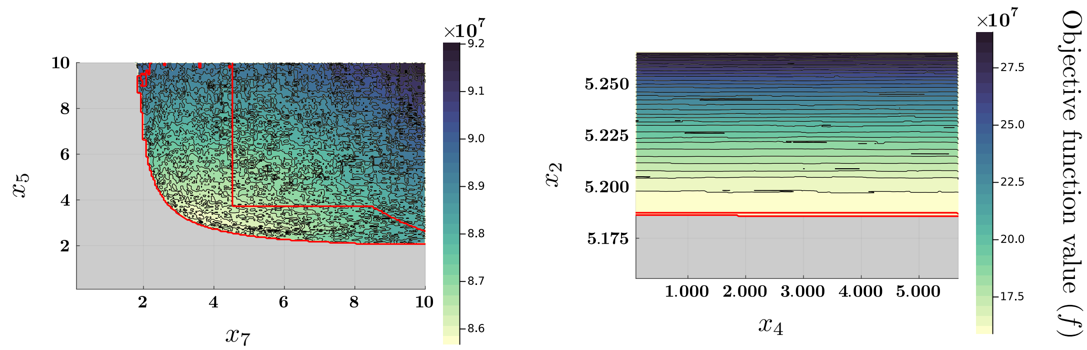
4 Micro-PRIAD behaviour illustrations

Micro-PRIAD is a collection of problems meant for benchmarking BBO solvers. This section showcases the general behaviour and complexity of Micro-PRIAD. It shows that Micro-PRIAD meets its goal of being a benchmark problem. Section 4.1 illustrates the influence of $\boldsymbol{x}$ and N on the feasibility and objective function value. Section 4.2 illustrates the use of Micro-PRIAD in an optimization context.

The results of the present section are done on a Dell Precision 3680 workstation equipped with an Intel Core i9-14900K processor (24 cores, 32 threads, up to 6.0 GHz) and 128 GB of RAM.

4.1 Cross-sectional illustrations

Micro-PRIAD is a realistic engineering benchmarking bank of problems. It involves the difficulties of BBO, such as discontinuity, non-smoothness, and small feasible regions. To illustrate the complexity of the blackbox, the objective function value is plotted in Figure 8 by varying only two input variables at a time.



(a) Presents the behaviour of the partial optimization of the maintenance on the rural town power grid problem (PG="r", $n = 15$). It shows a large feasible region.

(b) Presents the behaviour of the within-city reach optimization of the maintenance on the dense urban city power grid problem, where a small interval of values of x_2 is studied (PG="u", $n = 13$). It shows a small feasible region.

Figure 8: Illustration of different feasible region sizes in different Micro-PRIAD instances. The objective function value is illustrated with contour lines when varying two input variables.

Figure 8 illustrates the objective function value with the a priori infeasible region in gray and the feasible region boundary in red while varying the value of two input variables. It presents the behaviour of the objective function averaged over $N = 500$ MC draws. Each of the two plot from Figure 8 are defined by a 200×200 grid and takes more than 10 hours to generate.

By comparing the right plot and the left plot of Figure 8, the *within-city reach* optimization of the maintenance on the *dense urban city* power grid instance presents a bigger optimization challenge than the *partial* optimization of the maintenance on the *rural town* power grid instance. Indeed, simply by comparing the size of the feasible region, this conclusion is supported. In fact, to generate the right plot of Figure 8, the biggest challenge is finding a feasible point. The *within-city reach* optimization of the maintenance on the *dense urban city* power grid instance is the hardest of the instances presented in this work in terms of feasibility. Conversely, the easier instance is either *complete* or *partial* optimization of the maintenance on the *suburban city* power grid.

Figure 9 presents the behaviour of the objective function averaged over different value of N while varying the same two input variables. Each of the four plot from Figure 9 are defined by a 200×200 grid.

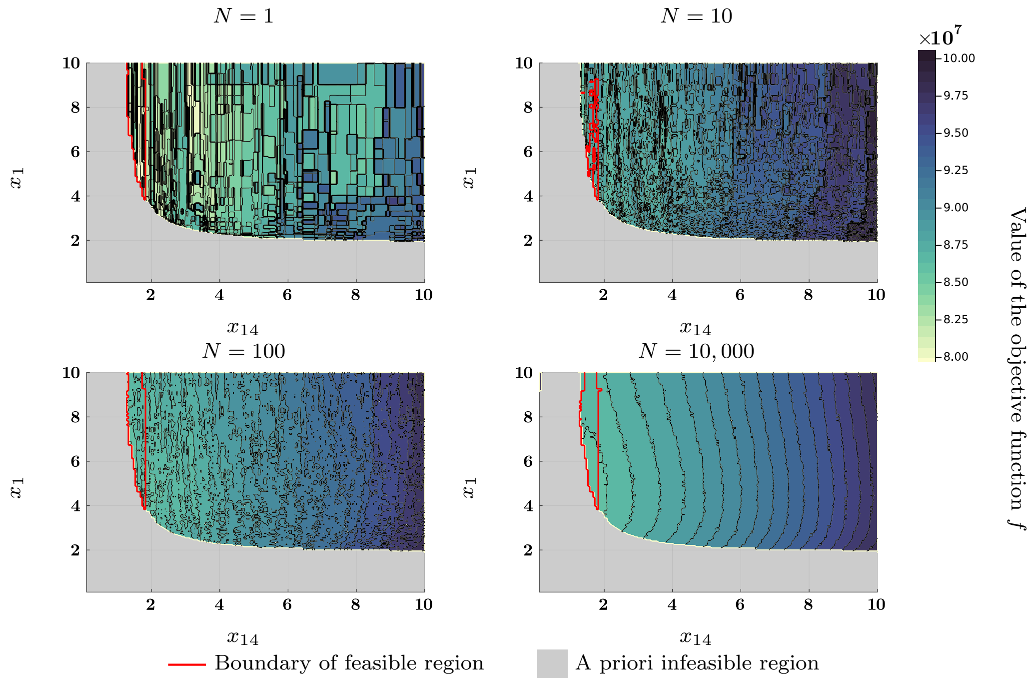


Figure 9: Illustration of the influence of N on the smoothness and the discontinuity of f. These plots represent the behaviour of the partial optimization of the maintenance on the rural town power grid problem.

Figure 9 shows that the objective function value becomes smoother when the value of N increases. At low values of N, the stochasticity of f is apparent. The outline of the feasible region becomes clear at $N = 100$. This illustrates that the random draws influence the feasibility. Conversely, the a priori feasible region in grey is independent of the MC draws. The bottom-right plot of Figure 9 with $N = 10,000$ required a week to generate. Every time the value of N is multiplied by a factor of 10, the computation time is approximately multiplied by this factor.

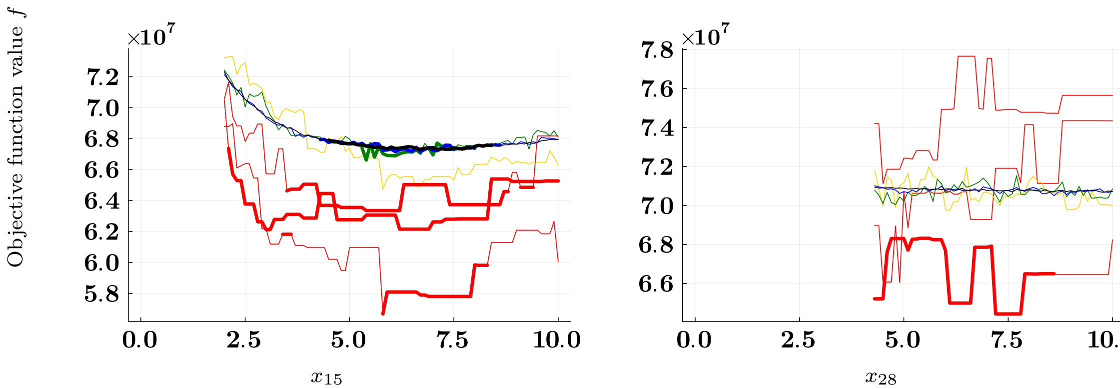
Figures 8 and 9 illustrates the general complexity of Micro-PRIAD, such as discontinuity and non-smoothness. Those BBO difficulties make Micro-PRIAD a good candidate for benchmarking.

Next, Figure 10 illustrates the influence of the number of MC draws done on the objective function and the feasibility, but by varying only one input variable at a time. In Figure 10, the colors represent the number of MC draws used to evaluate Micro-PRIAD. Feasible points and infeasible points are distinguished with different line styles. Additionally, the curves are shown only if the point does not violate the a priori constraints.

Figure 10 explains why the feasible region from Figure 9 at $N = 10$ is significantly different compared to the others values of N. Indeed, the left plot of Figure 10 shows that, sometimes, at low values of N, the point is deemed feasible when it is not at high values of N and deemed infeasible when it is feasible at high values of N. Moreover, the right plot of Figure 10 shows that depending on the seed, the value of f with $N = 1$ can be over or under the value of f with $N = 10000$.

4.2 Optimization benchmarking results

This section illustrate the use of Micro-PRIAD in the context of DFO and BBO benchmarking on 32 problem instances. A problem instance is defined by the starting point, the seed given to Micro-PRIAD, and one of the ten Micro-PRIAD instances. Data profiles [23] presented in this section are generated with RunnerPost[2] [9]. For all data profiles in this section, $\tau = 0.005$ and f is evaluated with the default value of $N = 10^4$.

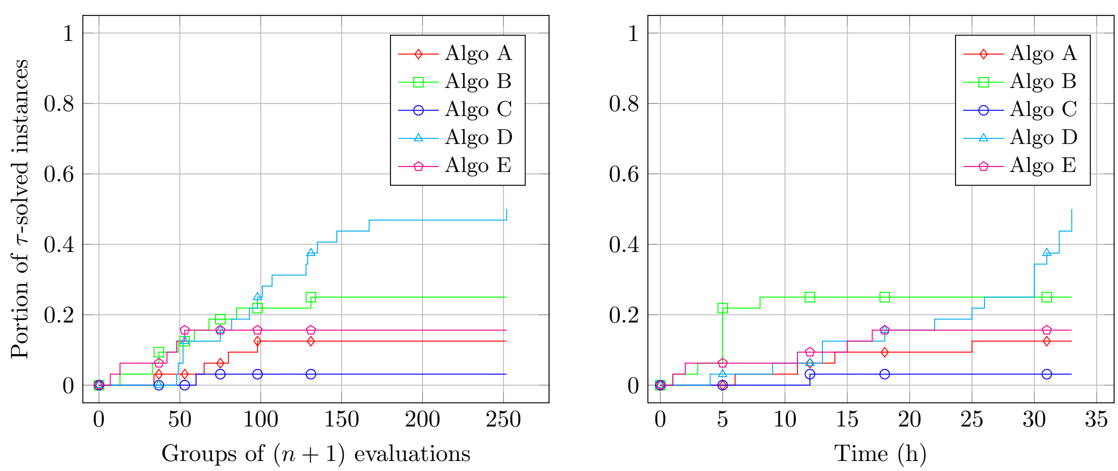


(a) Presents the behaviour of the complete optimization of the maintenance on the rural town power grid problem when varying x_{15} (x_{15} free on PG="r", $n = 28$).

(b) Presents the behaviour of the complete optimization of the maintenance on the dense urban city power grid problem when varying x_{28} (x_{28} free on PG="u", $n = 28$).

Figure 10: Objective function value plotted when varying only one input variable and fixing the others. The three red lines illustrate f evaluated with only one MC draw with three different seeds.

Optimizations are conducted on the 32 instances with five anonymized BBO solvers. Figure 11 illustrates these results on two plots of data profiles [23]. On Figure 11a, the horizontal axis is the number of group of $n+1$ evaluation of Micro-PRIAD, and on the right, it is the cumulative time spent to evaluate Micro-PRIAD. In both plots, the algorithms τ-solve between 3% and 50% of the problem instances after using their full budget. Further analysis revealed similar trends with different values of τ. This indicates that the instances of Micro-PRIAD are not trivial optimization problems.



(a) Data profile with groups of $n+1$ evaluations as horizontal axis.

(b) Data profile with time in hours as horizontal axis

Figure 11: Illustration of data profiles with five BBO solvers on 32 problem instances from Micro-PRIAD and $\tau = 0.005$. Solvers all have a 32 hours budget for the optimization.

Between the two plots of Figure 11, the shape of the curves of Algo B and Algo D significantly changes. Indeed, at low budget, Algo B τ-solves more instances when using the time scale compared to the groups of $n+1$ evaluations scale. Conversely, Algo D is better when using the groups of $n+1$ evaluations scale than the time scale. Using the time scale instead of the groups of $n+1$ evaluation

[2]https://github.com/bbopt/RunnerPost

scale makes Algo A better and Algo D worse. Indeed, Algo B explored a region of $\mathcal{X}$ where the points are faster to evaluate than normal, and Algo D explored a region where the points are longer to evaluate, it explains the shifting of the two curves.

Figure 12 illustrate the comparison of an algorithm that can use a static surrogate of the objective function in a case where one is provided and in the case that it is not. The objective function is considered to be f evaluated with $N = 10^4$ MC draws and the static surrogate to be f evaluated with $N = 10^3$ MC draws. To make the data profile fair, each 10 call of the surrogate count as one evaluation of the objective function [9].

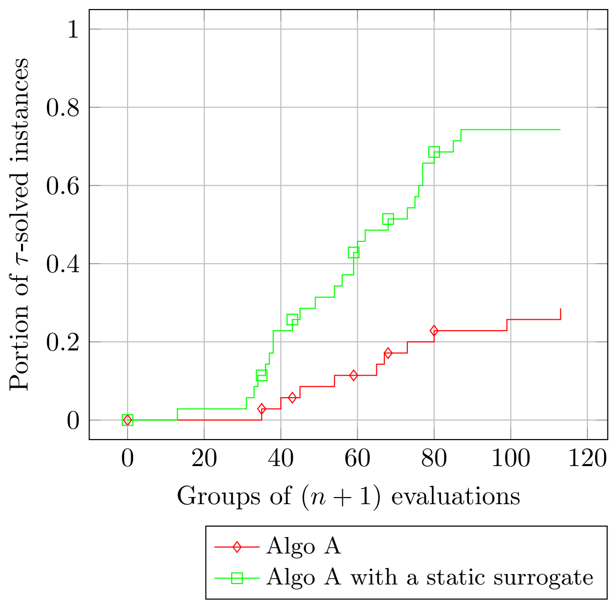


Figure 12: Illustration of a data profile with $\tau = 0.005$ where the same BBO solver is compared with itself in the case where no static surrogate of f is provided and in the case where the solver can call f with $N = 10^3$. The function f with $N = 10^3$ is used as a static surrogate of the function f with $N = 10^4$. The data profile is generated over 32 problems, where each Algo has 32 hours of budget.

When providing the surrogate to Algo A, the portion of τ-solved problems is significantly higher. Further analysis revealed the same conclusion with different values of τ. This observation show that using lower N values information can help a solver to conduct an optimization, suggesting that the use of intermediate outputs is advantageous.

5 Discussion

Micro-PRIAD is a collection of ten blackbox instances made for benchmarking. Nine of those are intended to be realistic engineering problems, and the tenth one is made for fast benchmarking. Each of the ten instances partially or fully simulates a power grid with different input decision variables. All ten are stochastic MC simulation-based and are separated into different processing blocks. The division into blocks comes from a blackbox in development at Hydro-Québec (HQ). Indeed, Micro-PRIAD reproduces the structure and behaviour of the PRIAD blackbox at a smaller scale. Recall that this problem involves prohibitive computational costs. This work is motivated by the need for an alternative simulator that is suitable for comparative analysis of BBO methods. Illustrative results show that Micro-PRIAD meets its goal of being a well-fitted problem for benchmarking, presenting several optimization challenges. Moreover, information between the processing blocks is made available to the solver and can be used as levers to improve the optimization. The optimization challenges and the levers available allow the benchmarking of solvers that aim to solve complex HQ BBO problems, and to a broader extent, power utility problems in general.

6 Appendix

Required Time for Maintenance

The vector containing the average required time for maintenance of each degradation mechanism listed in Table 1 (RTM) is presented in Table 6 for each of the three Micro-PRIAD power grids. The Maintenance index $= k$ corresponds to the maintenance associated with the degradation mechanism of the k^{th} line of Table 1.

Table 6: Required time in hours for each maintenance associated with degradation mechanisms listed in Table 1, for each power grid instance.

Station type	Equipment type	Degradation mecanism	Maintenance Index	RTM vector for Power Grid Instance		
				1	2	3
prod/HV	transformer	obstruction of electric radiators	1	4.0	5.5	5.000
prod/HV	transformer	theft of grounding cables	2	0.5	1.0	0.750
prod/HV	insulator	corrosion	3	3.5	4.5	5.000
prod/HV	insulator	copper theft	4	0.5	1.0	0.750
HV	cable	obstruction of power lines	5	6.0	6.5	5.500
HV	insulator	corrosion	6	3.5	4.5	5.000
HV	insulator	copper theft	7	0.5	1.0	0.750
HV/MV	transformer	obstruction of electric radiators	8	4.0	5.5	5.000
HV/MV	transformer	theft of grounding cables	9	0.5	1.0	0.750
HV/MV	disconnector	corrosion	10	3.5	4.5	5.000
HV/MV	disconnector	copper theft	11	0.5	1.0	0.750
HV/MV	breaker	corrosion	12	3.5	4.5	5.000
HV/MV	breaker	copper theft	13	0.5	1.0	0.750
HV/MV	insulator	corrosion	14	3.5	4.5	5.000
HV/MV	insulator	copper theft	15	0.5	1.0	0.750
MV	cable	obstruction of power lines	16	8.0	8.5	6.000
MV	cable	copper theft	17	0.5	0.5	0.333
MV	insulator	corrosion	18	3.5	3.5	2.500
MV	insulator	copper theft	19	0.5	0.5	0.333
MV/LV	transformer	obstruction of electric radiators	20	3.5	3.5	2.500
MV/LV	transformer	theft of grounding cables	21	0.5	0.5	0.333
MV/LV	disconnector	corrosion	22	3.5	3.5	2.500
MV/LV	disconnector	copper theft	23	0.5	0.5	0.333
MV/LV	breaker	corrosion	24	3.5	3.5	2.500
MV/LV	breaker	copper theft	25	0.5	0.5	0.333
MV/LV	insulator	corrosion	26	3.5	3.5	2.500
MV/LV	insulator	copper theft	27	0.5	0.5	0.333
LV	cable	copper theft	28	0.5	0.5	0.333

Micro-PRIAD's User Guide

The blackbox can be used in three different ways:

1. by calling the blackbox using a terminal command and keyword argument;
2. by calling the blackbox using a terminal command and a text file containing the parameters;
3. or by implementing the optimization algorithm directly in Julia, where it will be possible to call the Julia function directly;

With the first method (1.), $\boldsymbol{x}$ is specified in a text file, and the arguments are directly specified in the terminal command. An example of the command line that is used to call the blackbox in the terminal is presented below:

```
Julia $MICRO_PRIAD_HOME/src/run.jl -N 1000 -eta 100 -seed 0 pathToX/x.txt
```

Or In A More General Way

```
Julia $MICRO_PRIAD_HOME/src/run.jl -param1 value1 -param2 value2 ...  pathToX/x.txt
```

where `MICRO_PRIAD_HOME` represents the location of the directory containing the blackbox, and `pathToX` represents the location of the x.txt file. An example of an x.txt file is illustrated in Figure 13 at the end of the user guide.

With the second method (2.), $\boldsymbol{x}$ and the parameters must be specified in separate text files. Parameters are optional and can be initialized in a parameter file. If none of the optional parameters are set, the parameter file is not required. An example of the command line that is used to call the blackbox in the terminal is presented below:

```
Julia $MICRO_PRIAD_HOME/src/run.jl pathToARGS/ARGS.txt pathToX/x.txt
```

OR

```
Julia $MICRO_PRIAD_HOME/src/run.jl pathToX/x.txt
```

where `pathToARGS` represents the location of the parameter file, and `ARGS.txt` represents the parameter file. In the parameter file, a line can be commented out by adding a # at the beginning of a line. An example of a parameter file for calling the blackbox is illustrated in Figure 14, and an example of an x.txt file is illustrated in Figure 13; both examples are at the end of the user guide.

With the third method (3.), the input variable $\boldsymbol{x}$ must be passed to the Julia function. The other parameters are optional arguments to the Julia function and can be set by setting a value to the parameter when calling Micro-PRIAD.

The information given in the terminal command (1.) and the one given in the parameter file (2.) are the same. The couple (`param1 value1`) would be represented by a line in the parameter file if `param1` is set to `value1`.

The return of the outputs at the end of the evaluation for the given point will be displayed in the terminal as a message followed by a line break.

Input (x)

The $\boldsymbol{x}$ is provided to the blackbox in a text file. It represents the point to be evaluated and follows the input format defined in Figure 13 at the end of the user guide. Depending on n, the length of the input vector, a different optimization problem is executed. The text file must exclusively contain $\boldsymbol{x}$, with each component of the vector separated by a space.

Principal parameters

This section present parameters listed in Section 2.4.

`PG`

The `PG` parameter selects which power grid to execute. There are three power grids, described in Section 2.2. Each combination of power grid and n represents a different optimization problem instance.

The `PG` parameter can be provided to the blackbox by writing in the command line:

```
-PG s,
```

it runs the optimization on the suburban city power grid. The default value of the `PG` parameter is "s". This parameter can take the values "u", "s", or "r" to run power grids described in Section 2.2. It can also take the value t to run the limited fast benchmarking additional problem instance presented at the end of Section 2.5.

In this section, it is assumed that the first method (1.) is employed to call the blackbox, namely by using a command in the terminal to call Micro-PRIAD, and by giving arguments in the command line. To use the second method (2.), the example given in this section can be adapted for the parameter file by removing the -̈änd writing the example in a line of the parameter file.

`N`

The `N` parameter controls the number of Monte Carlo samples done during a blackbox call.

The `N` parameter can be provided to the blackbox by writing in the command line:

`-N 10000`,

it assigns 10 000 to the parameter `N`. The default value of the parameter `N` is 10 000.

The parameter `N` must be a positive integer. Furthermore, modifying this parameter while using the fidelity feature influences the fidelity calculation (e.g., `fidelity = 0.0001` will no longer necessarily represent exactly 1 Monte Carlo sample).

`s`

The `s` parameter controls the size of the group evaluated between each subsample of the MC draws and each intermediate output to the energy interrupter.

The `s` parameter can be provided to the blackbox by writing in the command line:

`-s 1000`,

it assigns the value 1000 to `s`. The default value of the parameter `s` is 1000. This parameter must be a strictly positive integer less than or equal to `N`.

`eta`

The `eta` parameter controls the number of groups of size `s` necessary to complete an evaluation.

The `eta` parameter can be provided to the blackbox by writing in the command line:

`-eta 10`,

it assigns the value 10 to `eta`. The default value of the parameter `eta` is 10. This parameter must be a strictly positive integer less than or equal to `N`.

`seed`

The `seed` parameter is the random seed controlling the Monte Carlo sampling within the blackbox. Changing the `seed` yields different results because the random draws vary. Conversely, maintaining the default value (0) ensures that the outputs for a given input remain identical, which allows for deterministic comparison of blackbox optimization algorithms.

The `seed` parameter can be provided to the blackbox by writing in the command line :

`-seed 0`,

it assigns a value of 0 to `seed`. The default value of the `seed` parameter is 0. This parameter must be a positive integer or set to -1. If set to -1, the seed is generated automatically from the operating system's entropy source, producing different stochastic realizations at each execution.

Parameters for the features of Micro-PRIAD

`availInterrupter`

The `availInterrupter` parameter allows enabling or disabling the intermediary outputs to the availability Interrupter illustrated in Figure 7. At this intermediary output, not all the constraints and objective functions are evaluated at the same N_i. Therefore, the information communicated to the algorithm consists of a primary vector containing the objective function and the nine constraints described in Table 4, alongside a secondary ten-element vector indicating the N_i at which each component of the primary vector was evaluated. This intermediary output can be enabled by setting `availInterrupter` to *true* or *1*, or disabled by setting it to *false* or *0*.

The `availInterrupter` parameter can be provided to the blackboxby writing in the command line:

`-availInterrupter true`,

it assigns the value *true* to `availInterrupter` parameter. The default value of the `availInterrupter` parameter is *true*.

This parameter does not influence the optimization if the intermediary outputs are not used by the solver.

`Interrupter`

The `Interrupter` parameter specifies to the blackbox the location and name of the function responsible for deciding whether to continue or interrupt the evaluation of a point at a given intermediary output. To continue the current evaluation, the function must return *true*. To interrupt the evaluation and return the intermediate values, it must return *false*. Four `Interrupter` functions are already implemented within the Micro-PRIAD source files and are described below:

***basicInterrupter*:** This function always returns *true* to Micro-PRIAD, meaning that regardless of the received f and $\boldsymbol{c}$, the evaluation will proceed to completion. This function is intended for optimization algorithms that do not utilize intermediary outputs. It does not require any additional parameters.

***printInterReturnInterrupter*:** This function implements a communication method with an external algorithm outside the Julia language environment. At each intermediary output, the algorithm writes the intermediate result, along with the fidelity level of each constraint and the objective function evaluation, into a primary text file (e.g., `result.txt`). Following the write operation to this primary file, the *solver* makes its decision based on this information and returns its conclusion in text format within a secondary text file (e.g., `decision.txt`) by writing a boolean value (*true* or *false*). Clearing the file contents is handled by Micro-PRIAD. To use this function, two additional parameters must be provided: the path and filename of these two text files. These supplementary parameters are passed via `AnyParamForInterrupter` parameter.

***DeterministicInfoInterrupter*:** This function returns *false* whenever one of the 5 deterministic constraints presented in Section 2.5 is violated, and returns *true* otherwise. This function requires no additional parameters.

The additional parameters required by the *printInterReturnInterrupter* function are passed to the blackbox via the subsequent parameter (`AnyParamForInterrupter`). If none of these predefined functions satisfy the requirements of a specific algorithm, or if the external communication method implemented in *printInterReturnInterrupter* is too rudimentary, a custom `Interrupter` function can be constructed and provided to the blackbox.

The `Interrupter` parameter can be provided to the blackbox by writing in the command line the example of line below:

`-Interrupter MICRO_PRIAD_HOME/src/InterrupterFunctions.jl`

`printInterReturnInterrupter`.

Before specifying the name of the chosen `Interrupter` function, its file path must be indicated. This method enables the use of external Julia functions developed outside the Micro-PRIAD framework. The default value of the `Interrupter` function is *basicInterrupter*.

When using the *printInterReturnInterrupter* function, set the `AnyParamForInterrupter` parameter to the specified value described below.

`AnyParamForInterrupter`

The `AnyParamForInterrupter` parameter is the argument passed to the selected `Interrupter` function. This parameter is a *string* in Julia that is transferred directly to the `Interrupter` function. The `AnyParamForInterrupter` parameter for the *printInterReturnInterrupter* function can be provided to the blackbox by writing in the command calling Micro-PRIAD a line similar to the two below:

`-AnyParamForInterrupter PathToResult/InterReturn.txt`, or

`-AnyParamForInterrupter pathToResult/InterReturn.txt pathToResult/decision.txt`.

By default, this parameter is set to an empty *string*, in this case, the text file location and name are *MICRO_PRIAD_HOME/src/* and *interReturnLog.txt*.

`SubSampler`

The `SubSampler` parameter specifies to the blackbox the location and the name of the `SubSampler` function of the stage one output before calling stage two. This function receives the output of stage one by group of η outputs. It then returns the indices of the output that will be evaluated in stage two. A `SubSampler` function is already implemented within Micro-PRIAD source files and is described below:

***basicSubSampler*:** This function does not subsample. It simply returns all indices. It does not require any additional parameter.

`AnyParamForSubSampler`

The `AnyParamForSubSampler` parameter is the argument passed to the selected `SubSampler` function. This parameter is a *string* and is transferred directly to the `SubSampler` function. Its default value is an empty *string*.

Main benchmarking parameters

`loggingTime`

The `loggingTime` parameter enables the recording of the blackbox evaluation execution time. This time represents the duration required to evaluate a single point and excludes any time taken by the optimization algorithm to make decisions. Indeed, whether between the evaluation of different points or via intermediary communication or subsampling between the two stages, the time required for the algorithm to make these decisions is omitted from this logging. Execution times are logged in a text file. The recording of execution times can be enabled or disabled by providing `true` or `1`, or `false` or `0`. If set to `true`, a text file named `loggingTime.txt` will be created in the Micro-PRIAD directory. Alternatively, the user can specify the directory where the text file will be created and its name.

The `loggingTime` parameter can be provided to the blackbox by writing in the command line:

`-loggingTime false`,

`-loggingTime PathToResult/result/loggingTime.txt`, or

`-loggingTime PathToResult/result`.

Where `PathToResult` represents the location of the `result` folder. The default value of the parameter `loggingTime` is `false`.

`loggingN`

The `loggingN` parameter enables the recording of the number of MC trials done at each call of the blackbox. These numbers are logged in a text file. The logging can be enabled or disabled by providing `true` or `1`, or `false` or `0`. If set to `true`, a text file named `loggingN.txt` will be created in the Micro-PRIAD directory. Alternatively, the user can specify the directory where the text file will be created and its name.

The `loggingN` parameter can be provided to the blackbox by writing in the command line:

`-loggingN false`,

`-loggingN PathToResult/result/loggingN.txt`, or

`-loggingN PathToResult/result`.

Where `PathToResult` represents the location of the `result` folder. The default value of the parameter `loggingN` is `false`.

Other parameters

`single_MC_info_return`

The `single_MC_info_return` parameter specifies to the blackbox the location and name of the function resonsable for the communication of information between the solver and Micro-PRIAD at each M.C. trial. Three `single_MC_info_return` functions are already implemented within Micro-PRIAD source files and are described blow:

***basic_single_MC_info_return*:** This function does not return anything. This function is intended for solvers that do not need information at each M.C. trial. It does not require additional parameters.

***print_single_MC_info_return*:** This function return the value of f, c_6, c_7, c_8 and c_9 in a text file at each M.C. trial. To use this function η must be set to 1. While using this function, an additional parameter indicating the logging file path and the logging file name.

***print_single_MC_info_return_for_subSampling*:** This function return the value of f, c_8 and c_9 in a text file at each M.C. trial. It returns only values that depend on stage two of Micro-PRIAD. While using this function, an additional parameter indicating the logging file path and the logging file name.

While using a subsampling algorithm, the *print_single_MC_info_return* cannot be used.

The value of the *single_MC_info_return* function can be provided to the blackbox by writing in the command line :

`-single_MC_info_return MICRO_PRIAD_HOME/src/Single_MC_info_return_Functions.jl`

`print_single_MC_info_return`,

this line tells the program what file to include and what is the name of the function used. The default value of the `single_MC_info_return` function is *basic_single_MC_info_return*, and its path is already included.

`AnyParamForSingle_MC_Info_ReturnFunction`

The `AnyParamForSingle_MC_Info_ReturnFunction` parameter is the argument passed to the selected `Single_MC_Info_Return` function. This parameter is a *string* in Julia that is transferred directly to the `Single_MC_Info_Return` function. To set the path, the name or both for the logging time file of *print_single_MC_info_return* or *print_single_MC_info_return_for_subSampling* function, a line similar to the two below can be used:

`-AnyParamForSingle_MC_Info_ReturnFunction pathToResult/info.txt`, or

`-AnyParamForSingle_MC_Info_ReturnFunction pathToResult`.

Where *pathToResult* is the file location of the logging file. By default, this parameter is an empty *string*, in this case, the location and name of the logging file are *MICRO_PRIAD_HOME/src/* and *single_MC_info_log.txt*.

`first_s_to_one`

The `first_s_to_one` parameter enables or disables a functionality that Interrupter can leverage. It sets the size of the first group evaluated in the blue loop of Figure 7 to one if enabled. The `first_s_to_one` parameter can be provided to the blackbox by writing in the command line:

`-first_s_to_one true`,

it assigns *true* to the `first_s_to_one` parameter. The default value of `first_s_to_one` is *false*.

This loop with only one MC draw does not count in the η iterations of the loop, and the next iteration will be done over $s - 1$ MC draws.

`fidelity`

The `fidelity` parameter controls the fidelity level at which the final output is returned. Maximum fidelity (1.0) corresponds to the maximum number of Monte Carlo samples (N), which defaults to 10^4. Thus, `fidelity` can be assigned values between 0.0 and 1.0 with a resolution of 10^{-4}. The mapping between the fidelity and the number of MC trials done is linear, meaning that increasing the fidelity by 10^{-4} increment the number of MC trials done to average $f(\boldsymbol{x})$ by 1.

The `fidelity` parameter can be provided to the blackbox by writing in the command line:

`-fidelity 0.1`,

it assigns 0.1 to the `fidelity` parameter, thereby evaluating Micro-PRIAD using 10% of the maximum number of samples. The default value of `fidelity` is 1.0.

`loggingPhi`

The `loggingPhi` parameter enables the recording of the fidelity level at which each call to the blackbox is made. These fidelity levels are logged in a text file. The logging can be enabled or disabled by providing `true` or `1`, or `false` or `0`. If set to `true`, a text file named `loggingPhi.txt` will be created in the Micro-PRIAD directory. Alternatively, the user can specify the directory where the text file will be created and its name.

The `loggingPhi` parameter can be provided to the blackbox by writing in the command line:

`-loggingPhi false`,

`-loggingPhi PathToResult/result/loggingPhi.txt`, or

`-loggingPhi PathToResult/result`.

Where `PathToResult` represents the location of the `result` folder. The default value of the parameter `loggingPhi` is `false`.

Output description

Micro-PRIAD print the outputs in the terminal with a line break, when the calling methode (1.) or (2.) is used. If the calling methode (3.) is used then it returns a *string*.

The output is of the form:

$$countEvalFlag\ f\ c_1\ c_2\ c_3\ c_4\ c_5\ c_6\ c_7\ c_8\ c_9.$$

The countEvalFlag is set to 0.0 if the a priori constraints are violated and is set to 1.0 otherwise. The objective function and the nine constraints are printed in scientific format if too large. For each of the nine constraint, if the constraint's value is above 0.0 then it is dimed violated, and the point classified as infeasible.

x.txt
1 3.92181 1 6.77591 2 2.00488 2 1.75796 1 3.36608 2 1.69558 9.97996

Figure 13: Example of an x.txt file for a problem with $n = 13$.

```
ARGS.txt
seed 0

PG r

N 10000

eta 10

s 1000

Interrupter ../src/MicroPRIAD.jl basicInterrupter

# Interrupter ../src/MicroPRIAD.jl printInterReturnConinueEval
# AnyParamForInterrupter ../result/InterReturn(r).txt

# Interrupter ../src/MicroPRIAD.jl printInterReturnConinueEval
# AnyParamForInterrupter ../result/InterReturn(r).txt ../result/decision.txt

SubSampler ../src/SubSamplingFunctions.jl basicSubSampler

loggingTime false
# loggingTime ../result/loggingTime(r).txt
# loggingTime ../result

availInterrupter 1

first_s_to_one false

fidelity 1.0

single_MC_info_return false
# single_MC_info_return ../result
# single_MC_info_return ../result/single_MC_info_logging(r).txt
```

Figure 14: Example of an ARGS file where lines starting with # are commented line (not considered by Micro-PRIAD).

References

[1] S. Alarie, C. Audet, M. Diago, S. Le Digabel, and X. Lebeuf. Inter-DS: A cost saving algorithm for expensive constrained multi-fidelity blackbox optimization. Computational Optimization and Applications, 90(3):607–629, 2025. doi:10.1007/s10589-024-00645-w.

[2] S. Alarie, C. Audet, M. Diago, S. Le Digabel, and X. Lebeuf. Multi-fidelity constraints in blackbox optimization. Technical Report G-2026-01, Les cahiers du GERAD, 2026. doi:10.48550/arXiv.2601.06321.

[3] S. Alarie, C. Audet, A.E. Gheribi, M. Kokkolaras, and S. Le Digabel. Two decades of blackbox optimization applications. EURO Journal on Computational Optimization, 9:100011, 2021. doi:10.1016/j.ejco.2021.100011.

[4] N. Andrés-Thió, C. Audet, M. Diago, A.E. Gheribi, S. Le Digabel, X. Lebeuf, M. Lemyre Garneau, and C. Tribes. Solar: a solar thermal power plant simulator for blackbox optimization benchmarking. Optimization and Engineering, 26(3):1815–1861, 2025. doi:10.1007/s11081-024-09952-x.

[5] C. Audet, V. Béchard, and S. Le Digabel. Nonsmooth optimization through Mesh Adaptive Direct Search and Variable Neighborhood Search. Journal of Global Optimization, 41(2):299–318, 2008. doi:10.1007/s10898-007-9234-1.

[6] C. Audet and J.E. Dennis, Jr. A Progressive Barrier for Derivative-Free Nonlinear Programming. SIAM Journal on Optimization, 20(1):445–472, 2009. doi:10.1137/070692662.

[7] C. Audet, M. Diago, S. Le Digabel, X. Lebeuf, P. Brillion, and C. Tribes. Benchmarking derivative-free optimization solvers for CSP plant design using the SOLAR simulator. Technical Report G-2025-70, Les cahiers du GERAD, 2025. URL: https://www.gerad.ca/fr/papers/G-2025-70.

[8] C. Audet and W. Hare. Derivative-Free and Blackbox Optimization. Springer Series in Operations Research and Financial Engineering. Springer, Cham, Switzerland, 2nd edition, 2026. doi:10.1007/978-3-032-00906-7.

[9] C. Audet, W. Hare, and C. Tribes. A summary of benchmarking constrained, multi-objective and surrogate-assisted optimization methods. Optimization Letters, 2026. doi:10.1007/s11590-026-02302-z.

[10] V. Beiranvand, W. Hare, and Y. Lucet. Best practices for comparing optimization algorithms. Optimization and Engineering, 18(4):815–848, 2017. doi:10.1007/s11081-017-9366-1.

[11] T. Browne, B. Iooss, L. Le Gratiet, and J. Lonchampt. Stochastic simulators based optimization by Gaussian process metamodels – Application to maintenance investments planning issues. Technical Report 1509.03880, arXiv, 2015. doi:10.48550/arXiv.1512.07060.

[12] A. Côté, O. Blancke, S. Alarie, A. Dems, D. Komljenovic, and D. Messaoudi. Combining Historical Data and Domain Expert Knowledge Using Optimization to Model Electrical Equipment Reliability. In 2020 International Conference on Probabilistic Methods Applied to Power Systems (PMAPS), pages 1–6, Liege, Belgium, 2020. doi:10.1109/PMAPS47429.2020.9183620.

[13] M. Diago, X. Lebeuf, T.J.M. Meango, A. Côté, and C. Rajotte. Risk-Based Optimization of Periodic Maintenance for Power Grid Equipment. In 2025 Annual Reliability and Maintainability Symposium (RAMS), pages 1–6. IEEE, 2025. doi:10.1109/RAMS48127.2025.10935127.

[14] M. Diago, A. Mailhot, J.-M. Meango, A. Côté, and C. Rajotte. Optimisation des périodicités de maintenance préventive pour l'appareillage électrique. In Congrès Lambda Mu 23, 2022.

[15] E.D. Dolan and J.J. Moré. Benchmarking optimization software with performance profiles. Mathematical Programming, 91(2):201–213, 2002. doi:10.1007/s101070100263.

[16] W.L. Dunn and J. Kenneth Shultis. Appendix A – Some Common Probability Distributions. In Exploring Monte Carlo Methods, pages 447–495. Elsevier, second edition, 2023. doi:10.1016/B978-0-12-819739-4.00019-6.

[17] K.R. Fowler, J.P. Reese, C.E. Kees, J.E. Dennis Jr., C.T. Kelley, C.T. Miller, C. Audet, A.J. Booker, G. Couture, R.W. Darwin, M.W. Farthing, D.E. Finkel, J.M. Gablonsky, G. Gray, and T.G. Kolda. Comparison of derivative-free optimization methods for groundwater supply and hydraulic capture community problems. Advances in Water Resources, 31(5):743–757, 2008. doi:10.1016/j.advwatres.2008.01.010.

[18] M. Gaha, B. Chabane, D. Komljenovic, A. Côté, C. Hébert, O. Blancke, A. Delevari, and G.Abdul-Nour. Global Methodology for Electrical Utilities Maintenance Assessment Based on Risk-Informed Decision Making. Sustainability, 13(16):9091, 2021. doi:10.3390/su13169091.

[19] N.I.M. Gould, D. Orban, and Ph.L. Toint. CUTEst: a Constrained and Unconstrained Testing Environment with safe threads for mathematical optimization. Computational Optimization and Applications, 60(3):545–557, 2015. Code available at https://ccpforge.cse.rl.ac.uk/gf/project/cutest/wiki. doi:10.1007/s10589-014-9687-3.

[20] N. Hansen, S. Finck, R. Ros, and A. Auger. Real-parameter black-box optimization benchmarking 2009: Noiseless functions definitions. Technical Report RR-6829, INRIA, 2009. URL: https://hal.inria.fr/inria-00362633/.

[21] D. Komljenovic, D. Messaoudi, A. Côté, M. Gaha, L. Vouligny, S. Alarie, A. Dems, and O. Blancke. Asset Management in Electrical Utilities in the Context of Business and Operational Complexity. In 14th WCEAM Proceedings, pages 34–45, 2021. doi:10.1007/978-3-030-64228-0_4.

[22] S. Le Digabel and S.M. Wild. A taxonomy of constraints in black-box simulation-based optimization. Optimization and Engineering, 25(2):1125–1143, 2024. doi:10.1007/s11081-023-09839-3.

[23] J.J. Moré and S.M. Wild. Benchmarking Derivative-Free Optimization Algorithms. SIAM Journal on Optimization, 20(1):172–191, 2009. doi:10.1137/080724083.

[24] K. Murphy, C. Carter, and L. Wolfe. How long should I simulate, and for how many trials? A practical guide to reliability simulations. Annual Reliability and Maintainability Symposium. 2001 Proceedings. International Symposium on Product Quality and Integrity, pages 207–212, 2001. doi:10.1109/RAMS.2001.902468.

[25] S. Murphy, J. Apt, J. Moura, and F. Sowell. Resource adequacy risks to the bulk power system in North America. Applied Energy, 212:1360–1376, 2018. doi:10.1016/j.apenergy.2017.12.097.

[26] N. Ploskas and N.V. Sahinidis. Review and comparison of algorithms and software for mixed-integer derivative-free optimization. Journal of Global Optimization, 82(3):433–462, 2022. doi:10.1007/s10898-021-01085-0.

[27] L.M. Rios and N.V. Sahinidis. Derivative-free optimization: a review of algorithms and comparison of software implementations. Journal of Global Optimization, 56(3):1247–1293, 2013. doi:10.1007/s10898-012-9951-y.

[28] C. Tribes, J.-F. Dubé, and J.-Y. Trépanier. Decomposition of multidisciplinary optimization problems: formulations and application to a simplified wing design. Engineering Optimization, 37(8):775–796, 2005. doi:10.1080/03052150500289305.