HOKKAIDO UNIVERSITY

Title	Stochastic MRAS-Based Strategy Optimization: Integration into PyCATSHOO, a Platform for Probabilistic Performance Assessment of Complex Systems
Author(s)	Chraibi, Hassane; Houdebine, Jean-Christophe
Relation	Proceedings of the 10th International Symposium on Reliability Engineering and Risk Management (ISRERM 2026), June 28 - July 1, 2026, Sapporo, Japan
Citation	Proceedings of the 10th International Symposium on Reliability Engineering and Risk Management (ISRERM 2026)
Issue Date	2026-07-01
Doc URL	https://hdl.handle.net/2115/100420
Type	conference paper
File Information	C000003_1.pdf

Hokkaido University Collection of Scholarly and Academic Papers : HUSCAP

10th International Symposium on Reliability Engineering and Risk Management (ISRERM 2026),
28 June–01 July 2026, Sapporo, Japan
Kitahara, T., Miyamori, Y. & Beer, M. (editors)

Stochastic MRAS-Based Strategy Optimization: Integration into PyCATSHOO, a Platform for Probabilistic Performance Assessment of Complex Systems

Hassane Chraibi[1,*] and Jean-Christophe Houdebine[2]

[1]*EDF Lab Paris-Saclay, 7, boulevard Gaspard Monge, 91120 Palaiseau, France.*
E-mail: hassane.chraibi@edf.fr
[2]*ARISTE, 2, rue du Marché, 92160 Antony, France.*
E-mail: jean-christophe.houdebine@wanadoo.fr
[*]*Corresponding Author*

This paper investigates the integration of the stochastic Model Reference Adaptive Search (SMRAS) algorithm with a modeling framework based on piecewise deterministic Markov processes (PDMPs) for the analysis and optimization of complex hybrid systems. The proposed approach is implemented within the PyCATSHOO platform, which enables the modeling and simulation of systems combining discrete stochastic events with continuous deterministic dynamics.
A case study is used to evaluate the proposed framework through the optimization of the maintenance policy of a multi-component system subject to aging effects. In this model, component failure rates depend on both the physical age and an apparent age representing cumulative operating time. Maintenance actions, such as replacement and refurbishment, may partially or completely reset these ages and thus reduce failure rates.
The optimization problem accounts for several economic factors, including the costs of preventive maintenance actions, the costs of corrective maintenance following failures, and the revenues generated during system operation. These elements are combined into a global performance criterion that must be maximized over the system's operational lifetime.
The SMRAS algorithm is used to explore the space of possible maintenance policies and to identify near-optimal policies. The results demonstrate the relevance of combining SMRAS with PDMP-based simulation models for maintenance optimization. They also highlight the sensitivity of the algorithm to the choice of certain hyperparameters, emphasizing the importance of appropriate calibration to ensure stable and reliable optimization results.

Keywords: MRAS, optimization, complex systems, PDMP, PyCATSHOO.

1 Introduction

Ensuring that complex industrial systems operate efficiently, safely, and at reasonable cost is a major challenge for modern industries, particularly those involving large infrastructures or high-risk operations. Maintenance policies therefore play a central role in guaranteeing system availability and economic performance over long operational periods.

Traditional maintenance policies, such as reactive maintenance (repairing equipment after failure) or preventive maintenance (performing maintenance at fixed time intervals), are increasingly complemented or replaced by more advanced approaches. In particular, condition-based, predictive, and prescriptive maintenance policies rely on data analysis, artificial intelligence, and connected devices associated with the Industrial Internet of Things (IIoT) to improve maintenance planning and decision-making.

Despite these advances, managing complex systems composed of interacting and aging components remains challenging, especially when the objective is to optimize economic performance over long time horizons. Previous research has provided significant contributions to maintenance optimization and reliability modeling (de Jonge & Scarf, 2020). However, many existing approaches rely on simplified or static modeling assumptions that do not fully capture the dynamic nature of industrial systems. As a result, they may fail to accurately represent the temporal evolution of system reliability and operational performance (Zhao & Luo, 2024).

In practice, system evolution is driven by several mechanisms, including degradation processes, failures, maintenance actions, and possible system reconfigurations. These mechanisms introduce both structural complexity and uncertainty into system behavior. In such contexts, sequential stochastic optimization methods provide a relevant framework for identifying efficient maintenance policies.

Markov Decision Processes (MDPs) are widely used for maintenance optimization because they allow system degradation to be represented through a sequence of states and support sequential decision-making (Andersen, 2022). In this framework, the health state of a component is typically represented by a Markov chain whose states correspond to different degradation levels. At each state, a maintenance decision can be selected, generating a cost and influencing the future evolution of the system. The objective is then to determine the maintenance policy that minimizes the expected long-term cost.

However, when applied to complex systems composed of many interacting components, such discrete models may suffer from severe combinatorial explosion. This limitation motivates the use of simulation-based optimization approaches capable of exploring large decision spaces without requiring an explicit analytical model of the objective function.

In this context, this paper investigates the integration of the stochastic Model Reference Adaptive Search (SMRAS) optimization algorithm into the PyCATSHOO platform, which is dedicated to the probabilistic performance assessment of complex hybrid systems. The proposed approach combines simulation-based evaluation with an adaptive stochastic search procedure in order to identify efficient maintenance policies for systems exhibiting both continuous deterministic dynamics and discrete stochastic events.

The remainder of the paper is organized as follows. Section 2 introduces the case study used throughout the paper to illustrate the proposed approach. Section 3 presents the PyCATSHOO platform. Section 4 describes the principles of the SMRAS algorithm, the modifications introduced into PyCATSHOO to enable its integration, and the optimization results obtained for the case study. Finally, Section 5 concludes the paper.

2 Case study: optimizing a factory's maintenance policy

2.1 System Description

We consider a factory that must meet a production target (e.g., supplying a given amount of electricity) over a 20-year horizon. The factory consists of three workshops with fixed capacity, which are activated progressively according to production demand (cold start when the load increases). Each workshop contains three machine pools and operates only if all pools are operational. Each pool contains two machines operating in passive redundancy, so the pool remains operational as long as at least one machine functions.

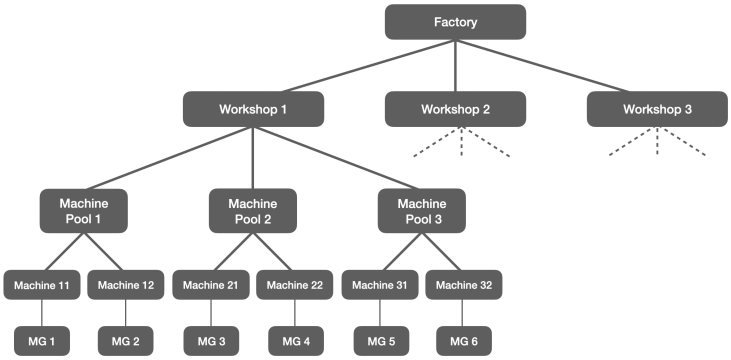


Figure 1. Hierarchical composition of the factory

2.2 Degradation process and maintenance actions

Each machine has two age measures that affect its reliability: **absoluteAge**, defined as the time elapsed since the last replacement, and **apparentAge**, corresponding to the cumulative operating time.

$$apparentAge(t) = \int_{\text{lastRefurbishmentTime}}^{t} ON(u)\, du, \tag{1}$$

where ON(u) is a boolean function equal to 1 when the machine is operating at time u and 0 otherwise.

These ages determine the machines' failure rates through equations of the following form:

$$\lambda(t) = \lambda_0(1 + a \cdot apparentAge(t) + b \cdot absoluteAge(t)) \tag{2}$$

This simplified linear form is chosen for illustration purposes. In practice, the relationship between age and failure rate may be more complex.

At each year over a 20-year horizon (19 decision points), a manager must choose one of three actions for each machine. The possible maintenance actions are refurbishment, replacement, and corrective maintenance. Refurbishment leaves the absolute age unchanged but resets the apparent age to zero. Replacement resets both the absolute age and the apparent age to zero. Corrective maintenance leaves both the absolute age and the apparent age unchanged.

2.3 Objective function and problem complexity

The goal is to maximize net gain over 20 years:

$$netGain = productionRevenue - maintenanceCosts - operatingCosts \tag{3}$$

where **productionRevenue** denotes the revenue generated when the workshops operate and meet demand, **maintenanceCosts** represents the costs of corrective repairs, refurbishments, and replacements, and **operatingCosts** corresponds to the normal operating costs of the equipment.

The number of possible policies is $3^{(18\times19)} = 3^{342}$, which is astronomical. The gain function has no analytical form; it must be estimated through Monte-Carlo simulation with uncertainty about failures. This combination makes any exhaustive enumeration or classical deterministic optimization approach impossible.

3 PyCATSHOO platform

PyCATSHOO (Chraibi et al., 2016) is a freeware platform developed within the R&D division of EDF for the probabilistic evaluation of the performance of complex systems, including hybrid systems. Such systems are characterized by the coexistence of continuous deterministic behaviors and discrete stochastic events.

PyCATSHOO is based on the theoretical framework of Piecewise Deterministic Markov Processes (PDMPs) (Davis, 1984). It is intended for the modeling of complex systems composed of interacting components and represents PDMPs as distributed hybrid stochastic automata organized into unit models, which are conceptually similar to classes in programming.

Each unit model describes the behavior of a class of components in the studied system using a largely declarative approach. It defines state variables, automata representing the component dynamics, and differential equations linking these variables. The automata consist of discrete states and transitions. These transitions may be deterministic, triggered by Boolean conditions, or stochastic, activated according to probability distributions.

During simulation, unit models evolve independently while interacting through a simple communication mechanism. This approach preserves the internal structure of each model and facilitates the scalability of large systems.

PyCATSHOO provides several quantification methods, the primary one being Monte Carlo simulation. It also includes algorithms for sequence exploration and can generate fault trees in the Open-PSA format (Epstein & Rauzy, n.d.), which can subsequently be analyzed using specialized tools.

In addition, PyCATSHOO integrates several optimization capabilities, including integer and mixed-integer linear programming methods, as well as the SMRAS algorithm, whose implementation is discussed in this article.

4 SMRAS principles and application to the case study

4.1 Case study optimization problem

The objective is to determine π^*, the optimal maintenance policy. This policy corresponds to a sequence of decisions for each machine that maximizes the expected profit of the factory over a 20-year operating period. The term expected is used deliberately because the system behavior is stochastic, and the optimization must therefore be expressed in terms of expectations. In the following, this expectation is denoted $G = \mathbb{E}(\text{netGain})$.

The problem can therefore be formulated as the following optimization problem:

$$\pi^* \in \underset{\pi\in\mathcal{P}}{\operatorname{argmax}} G(\pi) \tag{4}$$

where $\mathcal{P}$ denotes the set of all possible maintenance policies, and $G(\pi)$ represents the expected net gain associated with policy π. A policy is defined by a combinatorial representation expressed as a matrix of size $N_{\text{machines}} \times D$, where $N_{\text{machines}} = 18$ is the number of machines in the system and $D = 19$ is the number of decision points.

$$\pi = (a_{t,m})_{0\le t<19,\, 0\le m<18} \tag{5}$$

Each entry $a_{t,m}$ of this matrix corresponds to a maintenance action—refurbishment (0), replacement (1), or corrective maintenance (2)—for machine m at decision point t.

4.2 Keys of the SMRAS algorithm

The SMRAS algorithm (Stochastic Model-Reference Adaptive Search) is an adaptive population-based optimization algorithm (Hu et al., 2007). Its main idea is to progressively construct a probability distribution over the solution space, concentrating probability mass on better solutions over time.

More precisely, SMRAS generates a sequence of parameterized distributions $\{f(\cdot,\theta_k)\}_{k\ge0,\,\theta_k\in\Theta}$ such that:

$$f(\cdot, \theta_k) \xrightarrow[k \to +\infty]{} f(\cdot, \theta^*) \tag{6}$$

where $f(\cdot, \theta^*)$ is a distribution concentrated on optimal solutions, meaning that $\mathbb{E}_{f(\cdot,\theta^*)}[G(\Pi)] = G(\pi^*)$, i.e., the expected gain under $f(\cdot, \theta^*)$ equals the optimal expected gain.

To estimate the distributions $\{f(\cdot, \theta_k)\}_{k \geq 0}$, SMRAS relies on a sequence of reference distributions $\{f_k^{\text{ref}}(\cdot)\}_{k \geq 0}$ that is known to converge to the optimal distribution. This second sequence is also constructed iteratively using the following update rule, which is designed to ensure convergence.

$$f_k^{ref}(.) = \frac{H(G(\cdot)) f_{k-1}^{\text{ref}}(\cdot)}{\mathbb{E}_{f_{k-1}^{\text{ref}}}[H(G(\Pi))]} \tag{7}$$

where $H(\cdot)$ is a non-zero positive increasing function. It therefore follows that:

$$\mathbb{E}_{f_k^{\text{ref}}}[H(G(\Pi))] = \frac{\mathbb{E}_{f_{k-1}^{\text{ref}}}[H(G(\Pi))^2]}{\mathbb{E}_{f_{k-1}^{\text{ref}}}[H(G(\Pi))]} \geq \mathbb{E}_{f_{k-1}^{\text{ref}}}[H(G(\Pi))] \tag{8}$$

Consequently, the expected gain under the reference distributions increases with the iterations. We thus obtain an increasing and bounded sequence, which converges to its supremum, as shown by (Hu et al., 2007).

$$\mathbb{E}_{f_k^{ref}}[H(G(\Pi))] \xrightarrow[k \to +\infty]{} H(G(\pi^*)) \tag{9}$$

However, a direct use of this reference distribution is limited because its analytical form is unknown. Consequently, we still have to rely on the sequence of parametric distributions $\{f(\cdot, \theta_k)\}_{k \geq 0}$. Fortunatly In our case, these distributions belong to the natural exponential family, since they correspond to multinomial distributions. Their explicit analytical form facilitates parameter estimation from samples of policies generated through simulation.

$$f_k(., \theta_k) = h(.) exp({}^t\theta_k C(.) - K(\theta_k)) \tag{10}$$

The objective is therefore to estimate the parameters θ_k so that the parametric distribution is as close as possible to the reference distribution in terms of the Kullback–Leibler divergence.

$$\theta_k = \underset{\theta \in \Theta}{\operatorname{argmin}} \mathcal{D}(f_k^{\text{ref}}(.) \| f(., \theta)) \tag{11}$$

where:

$$\mathcal{D}(f_k^{\text{ref}}(.) \| f(., \theta)) = \mathbb{E}_{f_k^{\text{ref}}}\left[ln(\frac{f_k^{\text{ref}}(\Pi)}{f(\Pi, \theta)})\right] \tag{12}$$

The parameter θ_k is an abstract concept whose structure can be defined in the way that is most appropriate for the problem. In our case, it is represented as a matrix specifying the probability of each of the three possible decisions for every decision point and each machine. It can therefore be written as follows:

$$P_k = (p_k(t, m, d))_{0 \leq t < 19,\, 0 \leq m < 18,\, 0 \leq d < 3} \tag{13}$$

The minimization of the Kullback–Leibler divergence can be carried out analytically because $f(\cdot, \theta_k)$ belongs to the natural exponential family.

Finally, we can adopt the following expression to compute P_k from a sample of policies generated through simulation:

$$p_{k+1}(t, m, d) = \nu \cdot p_k(t, m, d) + (1 - \nu) \cdot \frac{\sum_{i=0}^{N_k} \frac{[H(G(\pi_k^i))]^k}{g(p_k(t,m,d))} \mathcal{I}_k(H(G(\pi_k^i))) \mathbb{1}_{\{\pi_k^i(t,m)=d\}}}{\sum_{i=0}^{N_k} \frac{[H(G(\pi_k^i))]^k}{g(p_k(t,m,d))} \mathcal{I}_k(H(G(\pi_k^i)))} \tag{14}$$

where N_k is the number of policies generated at iteration k, π_k^i denotes the i-th policy generated at iteration k, and $\mathcal{I}_k(\cdot)$ represents a high-pass filter that evolves over the iterations to progressively focus on the best policies. This filter is characterized by a quantile ρ_k, which evolves over the iterations, and by a tolerance parameter ε. The function $g(\cdot)$, in turn, is introduced to balance exploration and exploitation, thereby preventing the algorithm from concentrating too quickly on policies that may be suboptimal.

In addition, a smoothing parameter ν is used to moderate the updates of P_k, which helps prevent premature convergence and ensures a more stable optimization process.

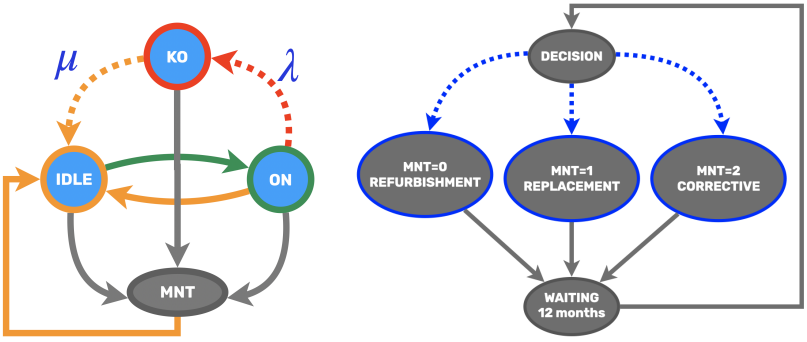
4.3 PyCATSHOO adaptations for SMRAS

We do not detail the modeling of system behavior based on the specifications given above, as it follows the usual PyCATSHOO modeling approach (Chraibi et al., 2016). Instead, we present the main concepts related to decision-making and optimization that were introduced into PyCATSHOO to enable the integration of the SMRAS algorithm.

In PyCATSHOO, component dynamics are modeled using automata, which are discrete-state machines with stochastic or deterministic transitions. Figure 2a illustrates an example of such an automaton representing the dynamics of a machine.

The integration of the optimization algorithm required the introduction of a new type of transition, called decision transitions, which are triggered by PyCATSHOO according to the policy selected by the optimization algorithm at each decision point.

Figure 2b illustrates the automaton describing the dynamics of the maintenance decision. In this automaton, three transitions originate from the DECISION state and are designated as decision transitions. These transitions correspond to the three possible maintenance actions: refurbishment, replacement, and corrective maintenance. The optimization algorithm determines which of these transitions is activated at each decision point.



(a) Automaton representing the dynamics of a machine. Dashed transitions are stochastic, whereas the others are triggered by deterministic rules.

(b) Automaton representing the dynamics of the maintenance decision. Dashed transitions correspond to decision transitions.

Figure 2. Automata used in the PyCATSHOO model of the case study.

The other modification introduced in PyCATSHOO to integrate the optimization algorithm concerns the nature of the variables. To allow the algorithm to search for optimal policies, it is necessary to designate one of the state variables as the optimization objective. In our case study, this variable is **netGain**, which represents the net profit of the factory.

4.4 Optimization results

To evaluate the MRAS algorithm integrated into PyCATSHOO, we applied it to two scenarios. The first scenario was designed to exclude preventive maintenance. To achieve this, the costs of replacement and refurbishment were significantly increased so that any maintenance action other than corrective maintenance would substantially reduce profitability.

```
<?xml version="1.0" encoding="UTF-8"?>
<STRATEGY REWARD="7777.21">
    <TR NAME="MGroup_1.DECISION">
        <CH I="2" ST="corrective"/>
        <CH I="2" ST="corrective"/>
        <CH I="2" ST="corrective"/>
        <CH I="0" ST="refurbishment"/>
        <CH I="2" ST="corrective"/>
        <CH I="0" ST="refurbishment"/>
        <CH I="2" ST="corrective"/>
        <CH I="0" ST="refurbishment"/>
        <CH I="2" ST="corrective"/>
        <CH I="2" ST="corrective"/>
        <CH I="0" ST="refurbishment"/>
        <CH I="2" ST="corrective"/>
        <CH I="2" ST="corrective"/>
        <CH I="2" ST="corrective"/>
        <CH I="0" ST="refurbishment"/>
        <CH I="2" ST="corrective"/>
        <CH I="0" ST="refurbishment"/>
        <CH I="2" ST="corrective"/>
        <CH I="2" ST="corrective"/>
    </TR>
    ....
</STRATEGY>
```

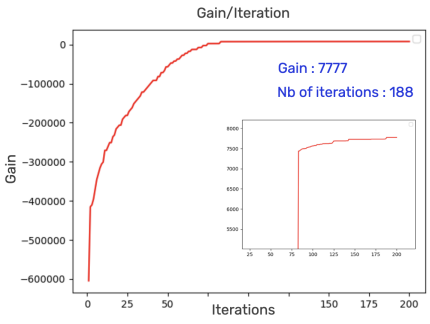


Figure 3. Illustrative results from the second scenario
Example of decision sequences and gain evolution over iterations

In the second scenario, the costs of replacement and refurbishment were moderated, while the impact of both actual and apparent machine ages was increased.

The policy obtained for the first scenario was optimal, as it matched the expected result. For the second scenario, the algorithm produced a highly profitable policy; however, its optimality cannot be formally established, although no modification of the obtained policy resulted in a higher profit.

In addition to these scenarios, several experiments were conducted to analyze the sensitivity of the algorithm to its hyperparameters. A noticeable sensitivity was observed with respect to parameters such as the coefficient controlling the balance between exploration and exploitation and the smoothing coefficient governing the evolution of the matrix P_k. Regarding the quantile defining the elite policies and the number of policies generated at each iteration, the algorithm proved to be relatively robust, since these parameters are dynamically adapted during the iterations.

The choice of the initial distribution also proved particularly important, as it affects the number of simulations required for convergence. The closer this distribution is to the optimal one, the faster the convergence. However, if it is too close to a local optimum, the algorithm may remain trapped there for several iterations. In practice, the exploration–exploitation mechanism proved effective in reducing this risk, although it may sometimes increase the number of iterations required for convergence.

5 Conclusion

This paper investigates the integration of the stochastic Model Reference Adaptive Search (MRAS) algorithm within the PyCATSHOO platform for the optimization of maintenance policies in complex hybrid systems. The proposed approach combines simulation-based system modeling with a stochastic optimization procedure capable of exploring extremely large decision spaces.

The results obtained for the case study show that the algorithm is able to identify highly profitable maintenance policies despite the combinatorial complexity of the problem and the absence of an analytical expression for the objective function. These results confirm that combining SMRAS with PDMP-based simulation models is a promising approach for maintenance optimization in complex industrial systems.

The experiments also revealed a noticeable sensitivity of the algorithm to the choice of certain hyperparameters and to the initial probability distribution. These factors influence the convergence behavior of the optimization process and therefore require careful tuning in practical applications.

It should be emphasized that the main objective of this work was not to provide a comprehensive benchmark of the MRAS algorithm itself. Rather, the goal was to evaluate whether such an optimization method could be effectively integrated into the PyCATSHOO modeling environment and used in practice by system modelers. From this perspective, the results are encouraging, since only a limited number of modifications were required to enable interaction between the optimization algorithm and the simulation models.

The integration approach presented in this work therefore opens the way for integrating additional simulation-based optimization algorithms into the PyCATSHOO platform in future developments.

One remaining limitation concerns parallelization. Although the original PyCATSHOO framework allows efficient parallel execution of Monte Carlo simulations, the current optimization workflow introduces a trade-off between parallel policy evaluation and parallel simulation within each policy. Future work will therefore focus on developing a parallel architecture capable of exploiting both levels simultaneously: policy evaluation and Monte Carlo simulation.

References

de Jonge, B., & Scarf, P. A. (2020).
A review on maintenance optimization. Operational Research, 285(3), 805-824. https://doi.org/10.1016/j.ejor.2019.09.0475,

Zhao X, Luo Y.
Dynamic reliability decision-making frameworks: trends and opportunities. Complex Eng Syst 2024;4:22. https://dx.doi.org/10.20517/ces.2024.54,

Andersen, Jesper Fink. "Maintenance optimization for multi-component systems using Markov decision processes." (2022),

Jiaqiao Hu, Michael C. Fu, Steven I. Marcus, (2007) A Model Reference Adaptive Search Method for Global Optimization. Operations Research 55(3):549-568. https://doi.org/10.1287/opre.1060.0367

H. Chraibi, J. C. Houdebine, A. Sibler. PyCATSHOO: Toward a new platform dedicated to dynamic reliability assessments of hybrid systems. PSAM 13. 13^{th} International Conference on Probabilistic Safety Assessment and Management. Oct 2-7 2016. Seoul, Korea

Epstein Steven, Rauzy Antoine Open-PSA Model Exchange Format. The Open-PSA Initiative. https://open-psa.github.io/j

M. H. A. Davis. "Piecewise-Deterministic Markov Processes: A general class of non-diffusion stochastic models", J. R. Stat. Soc. B 46:353-388, (1984).