

UNIVERSITÉ DE MONTRÉAL

OPTIMISATION DE BOÎTES NOIRES MULTI-FIDÉLITÉ SOUS CONTRAINTES

XAVIER LEBEUF

DÉPARTEMENT DE MATHÉMATIQUES APPLIQUÉES ET GÉNIE INDUSTRIEL  
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION  
DU DIPLÔME DE MAÎTRISE EN SCIENCES APPLIQUÉES  
(MATHÉMATIQUES APPLIQUÉES)

AVRIL 2023

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé :

OPTIMISATION DE BOÎTES NOIRES MULTI-FIDÉLITÉ SOUS CONTRAINTES

présenté par : LEBEUF Xavier

en vue de l'obtention du diplôme de : Maîtrise en sciences appliquées

a été dûment accepté par le jury d'examen constitué de :

M. NOM Prénom, Doct., président

M. AUDET Charles, PhD., membre et directeur de recherche

M. LE DIGABEL Sébastien, PhD., membre et codirecteur de recherche

M. DIAGO-MARTÍNEZ Miguel, PhD., membre et codirecteur de recherche

M. NOM Prénom, PhD., membre

## REMERCIEMENTS

Texte.

## RÉSUMÉ

Le résumé est un bref exposé du sujet traité, des objectifs visés, des hypothèses émises, des méthodes expérimentales utilisées et de l'analyse des résultats obtenus. On y présente également les principales conclusions de la recherche ainsi que ses applications éventuelles. En général, un résumé ne dépasse pas quatre pages.

Le résumé doit donner une idée exacte du contenu du mémoire ou de la thèse. Ce ne peut pas être une simple énumération des parties du document, car il doit faire ressortir l'originalité de la recherche, son aspect créatif et sa contribution au développement de la technologie ou à l'avancement des connaissances en génie et en sciences appliquées. Un résumé ne doit jamais comporter de références ou de figures.

## ABSTRACT

Written in English, the abstract is a brief summary similar to the previous section (Résumé). However, this section is not a word for word translation of the French.

## TABLE DES MATIÈRES

REMERCIEMENTS . . . . .	iii
RÉSUMÉ . . . . .	iv
ABSTRACT . . . . .	v
TABLE DES MATIÈRES . . . . .	vi
LISTE DES TABLEAUX . . . . .	viii
LISTE DES FIGURES . . . . .	ix
LISTE DES SIGLES ET ABRÉVIATIONS . . . . .	x
LISTE DES ANNEXES . . . . .	xi
CHAPITRE 1 INTRODUCTION . . . . .	1
1.1 Contexte . . . . .	1
1.2 Objectifs de recherche et plan du mémoire . . . . .	2
CHAPITRE 2 REVUE DE LITTÉRATURE . . . . .	4
2.1 Optimisation de boîtes noires . . . . .	4
2.2 Gestion des contraintes . . . . .	7
2.3 Multi-fidélité . . . . .	10
2.4 Boîtes noires stochastiques . . . . .	13
2.5 PRIAD . . . . .	14
2.6 Réduction du temps d'optimisation . . . . .	16
CHAPITRE 3 ÉTUDE DE L'IMPACT DE LA FIDÉLITÉ . . . . .	18
3.1 Algorithme d'évaluations hiérarchiques interrompues . . . . .	18
3.2 Représentativité des contraintes . . . . .	18
3.3 Satisfaction des contraintes . . . . .	18
3.4 Temps d'évaluation . . . . .	18
CHAPITRE 4 PARTITION DES CONTRAINTES . . . . .	19
4.1 Modèle d'optimisation de la partition . . . . .	19

4.2	Dépendance entre la satisfaction des contraintes et la fidélité . . . . .	19
4.3	Réductions de la taille du modèle . . . . .	19
CHAPITRE 5 RÉSULTATS . . . . .		20
5.1	Algorithmes implémentés . . . . .	20
5.2	Résultats numériques . . . . .	20
5.2.1	Données à priori . . . . .	20
5.2.2	Partition optimale . . . . .	20
5.2.3	Optimisations avec NOMAD . . . . .	20
5.2.4	Comparaison des partitions . . . . .	20
5.3	Discussion . . . . .	20
CHAPITRE 6 CONCLUSION . . . . .		21
6.1	Synthèse des travaux . . . . .	21
6.2	Limitations de la solution proposée . . . . .	21
6.3	Améliorations futures . . . . .	21
RÉFÉRENCES . . . . .		22
ANNEXES . . . . .		28

**LISTE DES TABLEAUX**

2.1	Définitions liées à l'optimisation de boîtes noires . . . . .	5
2.2	Définitions liées à la multi-fidélité . . . . .	12
2.3	Temps d'évaluation du simulateur de PRIAD par Komljenovic et al. (2019) . . . . .	15



**LISTE DES FIGURES**

2.1	Représentation en arbre de la taxonomie des contraintes QRAK traduite de Le Digabel et Wild (2015) . . . . .	11
2.2	Représentation graphique du simulateur de PRIAD . . . . .	14

## LISTE DES SIGLES ET ABRÉVIATIONS

IREQ	Institut de Recherche en Électricité du Québec
GERAD	Groupe d'Études et de Recherche en Analyse des Décisions
PRIAD	Programme de Robustesse, d'Intégration et d'Aide à la Décision
DFO	Optimisation sans dérivée ( <i>Derivative Free Optimisation</i> )
BBO	Optimisation de boîtes noires ( <i>Black Box Optimisation</i> )
NOMAD	Optimisation non linéaire par recherche directe sur treillis adaptifs ( <i>Nonlinear Optimisation by Mesh Adaptive Direct search</i> )
BP	Barrière Progressive
BE	Barrière Extrême
MADS	Recherche directe sur treillis adaptifs ( <i>Mesh Adaptive Direct Search</i> )
MC	Monte-Carlo
PEB	Barrière progressive à extrême ( <i>Progressive-to-Extreme Barrier</i> )

**LISTE DES ANNEXES**

Annexe A	DÉMO . . . . .	28
Annexe B	ENCORE UNE ANNEXE . . . . .	29
Annexe C	UNE DERNIÈRE ANNEXE . . . . .	30

## CHAPITRE 1 INTRODUCTION

Ce premier chapitre vise à exprimer la motivation derrière la recherche, les objectifs du projet ainsi que le plan du mémoire.

### 1.1 Contexte

La recherche effectuée est motivée par un problème rencontré par l'Institut de recherche en électricité du Québec (IREQ), dans le cadre du projet Programme de robustesse, d'intégration et d'aide à la décision (PRIAD) pour la gestion d'actifs d'Hydro-Québec TransÉnergie. L'Institut souhaite optimiser les périodicités de maintenance sur les équipements électriques du réseau électrique d'Hydro-Québec. L'un des objectifs de PRIAD est d'élaborer un simulateur du réseau de transport électrique d'Hydro-Québec qui prend en entrée les intervalles de temps où chaque type de maintenance est effectuée sur chaque sous-famille d'équipements de la partie du réseau simulée (ou du réseau en entier), pour associer un coût à cet ensemble d'intervalles, aussi appelé les périodicités de maintenances. À partir de ces dernières, le simulateur effectue des simulations d'indisponibilité des équipements ainsi que des simulations électriques en considérant les indisponibilités. Le coût est ensuite calculé en considérant, entre autres, le coût de la maintenance, le nombre de défaillances concourantes, la quantité d'énergie non livrée et la gravité des pannes. Le simulateur est détaillé et illustré dans la section 2.5. L'IREQ souhaite trouver les périodicités qui minimisent le coût. L'intuition derrière cet objectif est qu'il est possible d'imaginer qu'effectuer toutes les maintenances sur le réseau au complet quotidiennement coûterait beaucoup trop cher, et inversement, trop peu de maintenance causerait beaucoup de défaillances sur le réseau. L'objectif est d'optimiser les périodicités pour trouver celles qui font le meilleur compromis, c'est-à-dire celles qui minimisent le coût.

Il n'est pas possible d'extraire de ce simulateur un ensemble d'équations formant un modèle mathématique qui définit le comportement du simulateur. Il n'est donc pas possible d'utiliser les méthodes d'optimisation qui font usage d'expressions analytiques ou qui supposent l'existence des dérivées. Conséquemment, les méthodes d'optimisation sans dérivée : *Derivative Free Optimisation* (DFO) et d'optimisation de boîtes noires : *Black Box Optimisation* (BBO) doivent être mises à l'usage. En effet, le simulateur peut être vu comme un procédé qui prend des périodicités en entrée et qui renvoie un coût et des conditions, considérés respectivement comme l'objectif à optimiser et des contraintes mathématiques. Par exemple, une contrainte pourrait être que lors des simulations électriques, le nombre d'évènements où plusieurs équi-

pements sont indisponibles en même temps doit être plus petit ou égal à une certaine quantité. Formulé ainsi, le procédé peut être vu comme une boîte noire, et il s’agit d’un cas typique d’application de méthodes de BBO. Appliquer de telles méthodes avec un logiciel existant est généralement relativement simple, le simulateur de l’IREQ pose toutefois deux problèmes particuliers. Le premier est qu’il contient une simulation Monte-Carlo (MC), signifiant que la simulation est stochastique. À chaque lancement de la simulation, la méthode doit déterminer un niveau de précision avec lequel effectuer la simulation. Ce problème peut être contourné en demandant une haute qualité aux résultats en effectuant beaucoup de tirages MC, mais cette pratique cause le second problème. Les méthodes de BBO sont des méthodes itératives qui lancent la simulation à répétition. Si chaque simulation est longue lorsqu’on y effectue beaucoup de tirages, le temps total d’optimisation devient très grand. Dans le cas de PRIAD, le temps total estimé d’une méthode itérative qui lance la simulation du réseau de transport principal (soit 2000 équipements à simuler) 2000 fois avec  $5 \times 10^5$  tirages MC est d’environ 800 ans (voir le Tableau 2.3, où environ 800 ans correspond à 145 jours multiplié par 2000). Au moment de l’écriture de ce mémoire, le simulateur de PRIAD est toujours en développement. Le projet de maîtrise vise donc à développer une méthode générale, qui sera appliquée dans des travaux futurs à la problématique d’Hydro-Québec. La méthode présentée exploite l’aspect multi-fidélité ainsi que les contraintes d’une boîte noire pour réduire le temps d’optimisation. Sachant que la boîte noire de l’IREQ comportera plusieurs contraintes difficiles à satisfaire et un aspect stochastique provenant de la simulation MC, la méthode présentée dans ce mémoire s’appliquera à PRIAD.

## 1.2 Objectifs de recherche et plan du mémoire

L’objectif premier de ce projet de maîtrise est de créer une méthode d’optimisation de boîtes noires qui exploite l’aspect multi-fidélité dans le cas contraint pour sauver du temps d’optimisation. La motivation derrière cet objectif est le projet PRIAD d’Hydro-Québec, qui comprend une boîte noire très coûteuse à optimiser. Comme cette boîte noire comprend plusieurs contraintes difficiles à satisfaire, il arrive qu’un grand temps investi dans une évaluation est perdu car le point est finalement non réalisable. La méthode présentée dans ce mémoire vise à couper sur ce temps, en utilisant des évaluations à basse fidélité, donc plus rapides, pour déterminer si un point vaut les ressources d’une évaluation plus longue. Il en suit donc que plus les contraintes sont difficiles à satisfaire, plus il y a de temps à sauver avec la méthode proposée. Inversement, une boîte noire sans contraintes ou sans aspect multi-fidélité ne peut bénéficier de cette méthode.

Plusieurs autres travaux de recherche visent à effectuer des optimisations à précision variable.

En revanche, ces travaux s'intéressent le plus souvent à l'impact de la précision sur la valeur de l'objectif. Ces travaux visent à guider des algorithmes d'optimisation pour trouver à quelle fidélité effectuer chaque évaluation pour trouver de meilleures solutions. Le projet de maîtrise qui fait l'objet de ce mémoire s'intéresse plutôt à un autre aspect qui manque dans la littérature : l'impact de la précision variable sur les valeurs des contraintes, et l'utilisation de l'information que ces dernières peuvent donner à basse fidélité pour sauver du temps. En effet, la méthode présentée ici considère un ensemble discret de fidélités pour partitionner les contraintes en groupe, où chaque groupe correspond à une fidélité, et ensuite poser une hiérarchie sur ces groupes. Des travaux futurs pourront étudier la possibilité d'agencer la méthode de hiérarchie de groupes de contraintes aux méthodes d'optimisation multi-fidélité qui s'intéressent seulement à l'objectif pour proposer de nouvelles méthodes d'optimisation potentiellement très efficaces.

Le second objectif est de tester la méthode développée avec quelques implémentations différentes. La méthode des groupes de contraintes hiérarchisés est développée pour agir en tant que passerelle entre un algorithme d'optimisation et la boîte noire à optimiser. De ce fait, la méthode doit être couplée avec un autre algorithme d'optimisation de boîtes noires existant. De cette façon, les propriétés de convergence ainsi que les performances d'algorithmes qui ont déjà fait leur preuve peuvent être conservées, et la passerelle gère la partition des contraintes indépendamment de cet algorithme. Dans ce projet, la méthode est testée avec l'algorithme de recherche directe sur treillis adaptifs : *Mesh Adaptive Direct Search* (MADS), et avec différentes méthodes de gestion des contraintes. Le logiciel Optimisation non linéaire par recherche directe sur treillis adaptifs : *Nonlinear Optimisation by Mesh Adaptive Direct search* (NOMAD) est utilisé pour les tests.

\*\*\* plan du mémoire \*\*\* Je vais attendre de voir à quoi va ressembler la table des matières finale pour ce paragraphe.

## CHAPITRE 2 REVUE DE LITTÉRATURE

Ce second chapitre vise à établir les notions pré-requises à la compréhension de ce projet, ainsi qu'à décrire les avancements récents dans divers champs de l'optimisation de boîtes noires connexes à ce projet. Les notions de base sur l'optimisation de boîtes noires, la gestion des contraintes, la multi-fidélité, le projet d'Hydro-Québec PRIAD, l'étude des boîtes noires stochastiques ainsi que les méthodes de réduction du temps d'optimisation y sont couverts. Il est à noter que dans les chapitres subséquents, PRIAD ne sera que très peu mentionné pour les raisons mentionnées dans l'introduction. Il est tout de même important de couvrir le sujet dans la revue car ce sont les technicalités de ce projet qui ont forgé la méthode présentée dans ce mémoire. Les propriétés pouvant être exploitées par la méthode sont données par ce projet. La méthode est ensuite dite générale car elle s'applique à toute boîte noire qui comporte les mêmes propriétés.

### 2.1 Optimisation de boîtes noires

Cette section aborde le sujet de la BBO telle que décrite par Audet et Hare (2017). L'optimisation de boîtes noires est une branche de la recherche opérationnelle en mathématiques appliquées qui s'applique aux problèmes où certains éléments n'ont pas de formulation analytique. Aucune hypothèse n'est alors posée sur l'existence des fonctions décrivant le problème d'optimisation, et les hypothèses les plus faibles possibles sont posées sur les fonctions objectif et sur les contraintes. Une boîte noire prend un ou plusieurs paramètres en entrée, et applique un procédé qui renvoie une ou plusieurs valeurs de fonction objectif ainsi que les valeurs des contraintes, si applicable. Par exemple, un code informatique ou une expérience en laboratoire peut correspondre à cette définition. En particulier, ce document s'intéresse au cas mono-objectif contraint. Les problèmes multi-objectifs ne sont pas considérés.

Il est à noter que tout problème d'optimisation formulé en modèle analytique correspond également à cette définition, mais les méthodes qui exploitent l'information sur les dérivées sont généralement beaucoup plus efficaces que les méthodes de BBO. Une évaluation d'une boîte noire est définie comme l'action requise pour obtenir les valeurs des sorties étant donné certains paramètres. Le Tableau 2.1 liste les définitions des expressions mathématiques importantes.

Expression	Définition
$n \in \mathbb{R}$	Dimension du problème : nombre de paramètres d'entrée.
$m \in \mathbb{R}$	Nombre de contraintes relaxables du problème.
$X \subseteq \mathbb{R}^n$	Espace des paramètres.
$x \in X$	Point, vecteur contenant les valeurs des paramètres.
$f : X \rightarrow \mathbb{R}$	Fonction qui renvoie la valeur de l'objectif au point $x$ .
$c : X \rightarrow \mathbb{R}^m$	Fonction qui renvoie les valeurs des contraintes relaxables au point $x$ .
$c_j(x) \leq 0$	$j$ -ième contrainte relaxable, où $j \in J := \{1, 2, \dots, m\}$ .

TABLEAU 2.1 Définitions liées à l'optimisation de boîtes noires

Pour le reste du mémoire, par souci d'alléger l'écriture, l'abus de langage  $c_j$  est utilisé pour référer à une contrainte  $c_j(x) \leq 0$ . En résumé, une boîte noire prend en entrée  $x$  et renvoie  $f(x)$  et  $c(x)$ . L'équation (2.1) donne la formulation générale d'un problème d'optimisation.

$$\min_{x \in X} f \quad \text{s.c.} \quad x \in \Omega = \{x \in X : c_j(x) \leq 0, j \in J\} \quad (2.1)$$

Un point  $x$  est dit réalisable si toutes les contraintes y sont satisfaites. L'ensemble  $\Omega$  dénote l'ensemble des solutions réalisables. L'ensemble  $X$  est défini par les contraintes non-relaxables. Une contrainte est dite non-relaxable si elle doit être satisfaite par toute solution. Typiquement,  $X$  est borné par les limites inférieures et supérieures de chaque paramètre. Par exemple, un paramètre représentant une probabilité doit appartenir à l'ensemble  $[0, 1]$ . Ces contraintes ne renvoient pas nécessairement un nombre, elles pourraient renvoyer un message d'erreur par exemple. Lorsqu'elles ne sont pas respectées, toutes les sorties sont potentiellement erronées. Inversement, les contraintes relaxables n'ont pas d'impact sur la validité des autres sorties, et elle renvoient toujours une mesure de la réalisabilité d'un point. Posons à titre d'exemple une contrainte qui signifie qu'un budget ne doit pas dépasser  $b\$$ . Si un point  $x$  est tel que le budget excède  $b\$$ , la contrainte indiquera de combien le budget a été dépassé, et toutes les autres sorties sont tout de même valides. Il est posé par convention que l'objectif est minimisé. Sans perte de généralité, tout problème de maximisation peut se réécrire en problème de minimisation en changeant le signe de l'objectif.

Comme l'information sur les relations entre les entrées et les sorties d'une boîte noire sont supposées inaccessibles, les algorithmes d'optimisation de boîtes noires doivent obtenir l'in-



formation en exécutant la boîte noire. Plusieurs évaluations sont effectuées consécutivement, et chaque point à évaluer est déterminé par l'algorithme selon les sorties observées aux évaluations précédentes, à l'exception du premier point. La plupart des algorithmes sont itératifs avec un itéré  $x^k$ , où  $k$  représente le numéro de l'itération. Le point de départ est alors appelé  $x^0$ . À chaque itération, lorsqu'un point pour lequel la valeur de  $f$  est la plus petite tout en respectant les contraintes est trouvé,  $x^k$  est mis à jour. À l'atteinte d'un critère d'arrêt, l'algorithme renvoie le dernier itéré.

Les algorithmes de BBO sont divisés en trois grandes catégories. La première consiste en les méthodes qui génèrent des modèles de la boîte noire. Un modèle est un système qui tente de répliquer le comportement de la boîte noire, en ayant un coût d'évaluation faible. Le cadre d'optimisation de modèles entiers-mixtes : *Mixte-Integer Surrogate Optimisation framework* (MISO) présenté par Müller (2016) est un exemple d'implémentation d'une méthode qui correspond à cette catégorie. Il s'agit d'une implémentation MATLAB spécialisée pour les boîtes noires coûteuses en entier-mixte. La seconde comprend les méthodes de recherche directe, qui utilise directement l'information des évaluations. Quelques-unes de ces méthodes sont décrites ci-dessous. Plusieurs logiciels d'optimisation utilisent des algorithmes provenant de ces deux catégories conjointement. Finalement, il y a les méthodes heuristiques, qui ne sont pas discutées dans ce mémoire.

Il existe plusieurs algorithmes d'optimisation qui sont reconnus pour leur efficacité. Entre autres, l'algorithme MADS proposé par Audet et Dennis, Jr. (2006) se distingue par ses propriétés de convergence globale vers des minimums locaux. Il s'agit d'une amélioration directe de l'algorithme Recherche par motifs généralisé : *Generalized Pattern Search* proposé par Torczon (1992) où à partir de l'itéré courant, chaque nouveau point est généré en effectuant un pas dans la direction d'une coordonnée de l'espace des paramètres. À chaque itération,  $2n$  points à évaluer sont alors générés. Cette méthode est problématique lorsqu'un pas dans une combinaison de directions de différentes coordonnées doit être effectué pour atteindre un minimum. Avec MADS, les directions sont générées aléatoirement. Pour conserver les propriétés de convergence, les directions doivent former une base positive et donner des points positionnés sur un treillis adaptif. L'algorithme de Nelder et Mead (1965) place les points à évaluer sur un simplexe (un polytope à  $n+1$  arêtes). Dépendamment des évaluations, certains points du simplexe sont déplacés en effectuant une réflexion, une expansion, une contraction ou une réduction. Les résultats de cet algorithme sont souvent très dépendants du simplexe de départ. Huyer et Neumaier (1999) ont proposé l'algorithme Recherche par coordonnées multi-niveaux : *Multilevel Coordinate Search* qui consiste à diviser itérativement l'espace des paramètres en hyperrectangles. À chaque itération, un point est évalué et un hyperrectangle est divisé le long d'un axe auquel le point appartient et déterminé par le résultat de l'évalua-

tion. La quantité d’algorithmes de DFO et BBO est vaste, et la quantité d’implémentations logicielles différentes de ces algorithmes est toute aussi vaste. De ce fait, tel que démontré par Rios et Sahinidis (2013), il n’est pas évident de déterminer quelles implémentations de quels algorithmes sont les meilleurs. 22 logiciels différents ont été testés sur 502 problèmes pour découvrir que tout solveur a trouvé la meilleure solution pour au moins quelques problèmes, et qu’aucun solveur domine tous les autres. Plus récemment, une analyse similaire a été effectuée par Ploshkas et Sahinidis (2022) en s’intéressant aux problèmes entier-mixtes en particulier. Les conclusions ne sont pas les mêmes, l’étude a trouvé que NOMAD et MISO se démarquent de façon évidente.

Les quelques algorithmes présentés dans cette section sont plutôt vieux, mais des travaux de recherche sur ceux-ci continuent à générer des réflexions à leur sujet et à les améliorer ainsi que leurs implémentations. Conséquemment, ils sont encore fort pertinents aujourd’hui. Par exemple, Audet et al. (2022a) adaptent MADS pour tenir compte de problèmes où les meilleures solutions sont près de régions discontinues. Aussi, Ozaki et al. (2019) suggèrent un algorithme de parallélisation de la méthode de Nelder-Mead pour l’accélérer, tout en ajoutant des évaluations spéculatives basées sur un modèle, également effectuées en parallèle. Toujours au sujet de Nelder-Mead, Galántai (2022) effectue une preuve de convergence qui s’applique aux espaces dont la dimension n’excède pas huit. Alarie et al. (2021a) décrivent un large éventail d’applications en BBO qui ont eu lieu dans les 20 dernières années, principalement dans les domaines de l’énergie, de la science des matériaux et du génie informatique.

La méthode présentée dans de ce mémoire est testée avec le logiciel NOMAD présenté par Audet et al. (2022b). Il s’agit d’une implémentation de MADS qui vise particulièrement à optimiser des boîtes noires données par des programmes très coûteux en temps où le budget d’évaluations accordé à une optimisation est limité. Cette version récente est l’évolution du logiciel décrit par Le Digabel (2011).

## 2.2 Gestion des contraintes

Cette section décrit d’abord deux méthodes de gestion des points non réalisables au cours de l’application d’un algorithme d’optimisation. Elles permettent aussi de traiter les cas où le tout premier point, généralement donné par l’utilisateur, n’est pas réalisable. D’abord, définissons la fonction  $h : \mathbb{R}^n \mapsto \mathbb{R} \cup \{\infty\}$  appelée la fonction de violation des contraintes,

inspirée des méthodes par filtres de Gould et Toint (2010).

$$h(x) := \begin{cases} \sum_{j \in J} (\max\{c_j(x), 0\})^2 & \text{si } x \in X \\ \infty & \text{sinon.} \end{cases}$$

Avec cette définition,  $h$  est une fonction non négative qui renvoie 0 si  $x \in \Omega$ , et qui caractérise par quelle mesure  $x$  ne respecte pas les contraintes relaxables si  $x \notin \Omega$  et  $x \in X$ . Les valeurs des contraintes sont généralement mises à l'échelle pour que leur grandeur soit comparable. Si la fonction avait été définie sans le carré, mais plutôt avec une norme  $l_1$ , il y aurait introduction de non différentiabilité. Le carré permet à la fonction d'être non négative tout en conservant certaines propriétés de différentiabilité. De plus, la fonction avec le carré se comporte mieux dans un contexte algorithmique où l'on désire que la valeur de cette fonction atteigne zéro.

La première méthode se nomme la barrière extrême (BE). Cette méthode se définit par deux phases. La première vise à trouver un premier point réalisable en minimisant la fonction de violation des contraintes et en ignorant totalement la valeur de l'objectif. Cette première phase est utile uniquement lorsque le premier point n'est pas réalisable. La seconde phase optimise le problème en ignorant les points non réalisables. La méthode est détaillée à l'algorithme 2.1.

---

**Algorithme 2.1** : Barrière extrême à deux phases, tiré de Audet et Hare (2017)

---

Étant donné  $f_\Omega : X \rightarrow \mathbb{R} \cup \{\infty\}$ ,  $c : X \rightarrow \mathbb{R}^m$  et un point de départ  $x^0 \in X$

1. Phase de réalisabilité

Lancer un algorithme d'optimisation à partir de  $x^0$  pour résoudre  $\min_{x \in X} h(x)$ .

Terminer l'optimisation dès qu'un point réalisable  $\bar{x} \in \Omega$  est trouvé, puis aller à 2.

Si un autre critère d'arrêt est satisfait avant de trouver ce  $\bar{x}$ ,

terminer en concluant qu'aucun point appartenant à  $\Omega$  n'a été trouvé.

2. Phase d'optimisation

Définir  $f_\Omega := \begin{cases} f(x) & \text{si } x \in \Omega \\ \infty & \text{sinon} \end{cases}$

Lancer un algorithme d'optimisation à partir de  $\bar{x}$  pour résoudre  $\min_{x \in X} f_\Omega$

---

Cette façon de traiter les contraintes relaxables est très simple, et elle n'utilise pas l'information sur les points non-réalisables à partir de la seconde phase. La seconde méthode, proposée par Audet et Dennis, Jr. (2009), se nomme la barrière progressive, et elle tente d'utiliser cette information pour potentiellement trouver de meilleurs points réalisables tout au long de l'optimisation. La barrière progressive (BP) introduit un seuil  $h_{max}^k \in \mathbb{R}_+$ , mis à jour à chaque itération, où les itérations sont dénotées par  $k$ . Chaque  $x$  où  $h(x) > h_{max}^k$  est ignoré en attribuant à  $f(x)$  une valeur de  $\infty$ . Le seuil initial est donné par  $h_{max}^0 = \infty$ , et le seuil est monotone décroissant avec les itérations.

Au lieu de garder en mémoire un seul point itéré avec la meilleure valeur de  $f$ , la méthode propose de garder deux itérés : un point réalisable et un point non réalisable, nommés respectivement  $x^{fea}$  et  $x^{inf}$ . Les valeurs de l'objectif  $f(x^{fea})$  et  $f(x^{inf})$  sont initialisés à  $\infty$ , et les points sont mis à jour par l'algorithme 2.2.

---

**Algorithme 2.2 :** Mise à jour des points  $x^{fea}$  et  $x^{inf}$  de la barrière progressive

---

Étant donné  $x^{fea}, x^{inf}$  et un nouveau point  $x$  récemment évalué

Si  $h(x) = 0$  et  $f(x) < f(x^{fea})$

    |  $x^{fea} \leftarrow x$

Si  $f(x) \leq f(x^{inf})$  et  $h(x) \leq h(x^{inf})$  et ( $f(x) < f(x^{inf})$  ou  $(x) < h(x^{inf})$ )

    |  $x^{inf} \leftarrow x$

---

L'algorithme d'optimisation couplé avec la BP explore alors autour de  $x^{fea}$  et de  $x^{inf}$ . L'intérêt est que  $x^{inf}$  est souvent un point avec une bien meilleure valeur de  $f$  que  $x^{fea}$ , et en faisant progresser la valeur de  $h_{max}^k$  vers 0, il est possible de trouver un point réalisable près de  $x^{inf}$  très intéressant. La description de la barrière est intentionnellement vague car son application dépend de l'algorithme d'optimisation avec lequel elle est couplée.

Il est à noter que pour un même problème d'optimisation, il est possible d'appliquer différentes barrières aux différentes contraintes. Audet et al. (2010) présentent une troisième méthode, nommée la barrière progressive à extrême : *Progressive-to-Extreme Barrier* (PEB). Cette approche est utile pour les optimisations avec des points de départ non réalisables où l'on souhaite appliquer la BP seulement à la phase de réalisabilité de la BE. L'algorithme débute en appliquant la BP à toutes les contraintes. À chaque fois qu'un point évalué est tel qu'une contrainte qui n'était pas satisfaite précédemment le devient, cette contrainte est maintenant

traitée avec la BE. Une fois arrivé à la phase d’optimalité, toutes les contraintes sont traitées par la BE.

Dans la littérature récente, Papalexopoulos et al. (2022) proposent d’utiliser un réseau de neurones et un programme linéaire entier-mixte pour former un modèle qui peut traiter plus aisément les contraintes discrètes. Dans le cas de boîtes noires stochastiques, ces modèles sont souvent utilisés. Toutefois, Dzahini et al. (2022) suggèrent une modification à MADS où, au lieu d’utiliser des modèles, des estimations de fonctions et des bornes probabilistes avec des conditions suffisantes de descente permettent d’optimiser les cas contraints et stochastiques. Aussi, Audet et al. (2022c) analysent l’efficacité de la BP avec NOMAD en utilisant la plateforme COCO décrite par Hansen et al. (2021). Pour une meilleure utilisation de la BP, Audet et al. (2018a) proposent un algorithme de régions de confiance (construction de modèles locaux) où deux régions sont construites : une autour de  $x^{fea}$  et une seconde autour de  $x^{inf}$ . D’autre part, Bajaj et al. (2018) combinent un algorithme de régions de confiance avec une méthode comprenant une phase de réalisabilité et une phase d’optimisation très semblable à la BE.

D’un point de vue plus général, les contraintes peuvent être classifiées en neuf catégories selon Le Digabel et Wild (2015). La Figure 2.1 illustre en un arbre une série de questions à se poser pour classer une contrainte en partant de la racine. Chaque feuille de l’arbre correspond à une classe. D’abord, une contrainte peut être connue ou cachée. Une contrainte est connue si elle est explicitement donnée dans la formulation du problème. Une contrainte cachée peut s’agir d’un bogue informatique, ou elle peut simplement être  $x > 0$  dans le problème  $\min\{f(\log(x)) : x \in \mathbb{R}\}$  si la contrainte n’est pas indiquée au solveur. Une contrainte cachée est nécessairement une contrainte de simulation non relaxable et non quantifiable. Une contrainte est dite a priori si elle ne requiert pas le lancement d’une simulation pour vérifier sa réalisabilité. Cette définition inclut tous les problèmes formulés analytiquement, et les contraintes par simulation sont propres à l’optimisation de boîtes noires. La relaxabilité a déjà été discutée, et finalement une contrainte est quantifiable si elle renvoie une mesure de la réalisabilité de la contrainte. Une contrainte non quantifiable renvoie une quantité binaire, qui indique si elle est satisfaite ou non, sans savoir à quel degré.

### 2.3 Multi-fidélité

Cette section décrit le concept de multi-fidélité ainsi que ses implications dans un contexte d’optimisation de boîtes noires. Une fidélité est une mesure de la qualité des sorties d’un procédé et du coût en temps pour atteindre cette qualité. Une basse fidélité offre un résultat peu coûteux mais de qualité dégradée, alors qu’une haute fidélité correspond à des sorties

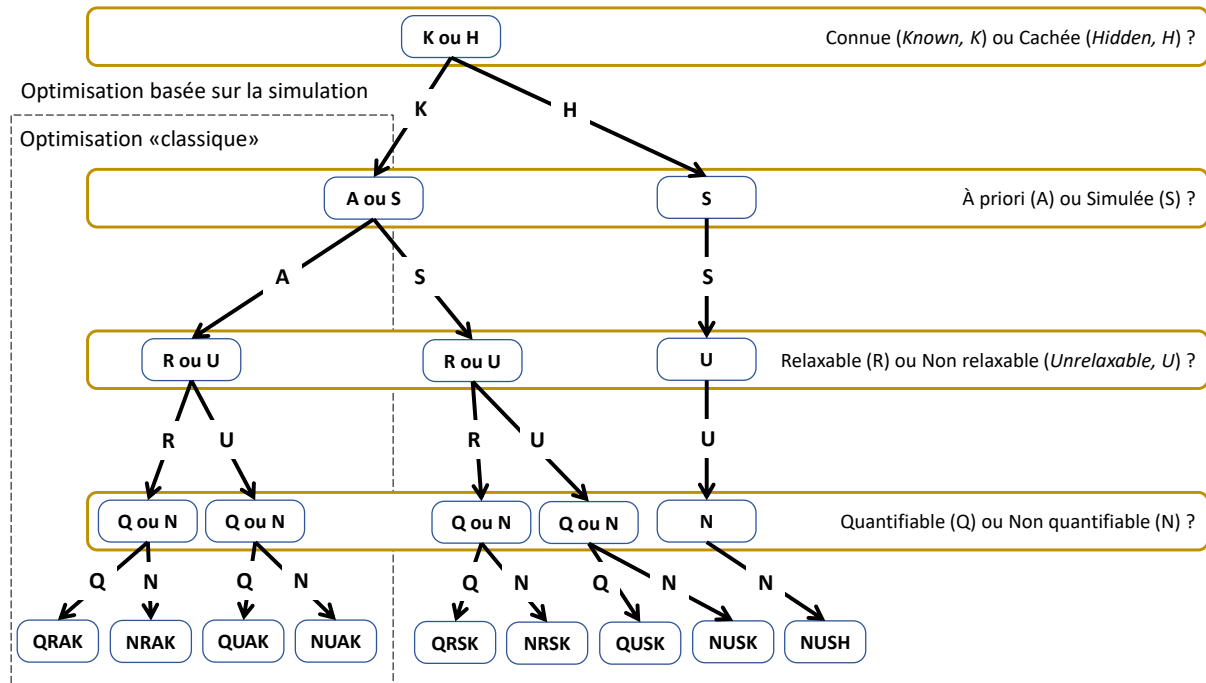


FIGURE 2.1 Représentation en arbre de la taxonomie des contraintes QRAK traduite de Le Digabel et Wild (2015)

très fiables mais coûteuses à obtenir. Il est à noter que la fidélité peut être définie de manière indépendante du coût, ce n'est cependant pas le cas pour ce projet de maîtrise. Un procédé est dit multi-fidélité s'il peut être effectué à différents niveaux de fidélité. Une grande branche de la BBO s'intéresse à la bi-fidélité, un cas particulier de la multi-fidélité où la vérité et une approximation de la vérité sont disponibles. Souvent, le procédé lui-même donne la vérité (haute-fidélité) et il existe un modèle qui donne une approximation (basse-fidélité). Il est aussi possible que la haute et la basse fidélité proviennent toutes deux du même procédé, en variant certains paramètres. En s'appliquant à la multi-fidélité, la méthode présentée dans ce mémoire s'applique aussi à la bi-fidélité, mais il faut savoir que les méthodes spécialisées en bi-fidélité comme celle de Balabanov et Venter (2004) risquent d'être plus efficaces lorsque seulement deux fidélités sont disponibles.

Pour citer un exemple de procédé multi-fidélité intéressant pour ce projet, une simulation MC peut être effectuée à différentes fidélités en variant le nombre de tirages effectués. En augmentant le nombre de tirages, la précision de la simulation et le temps de simulation augmentent et la fidélité est dite plus grande. Inversement, la fidélité diminue en diminuant le nombre de tirages. Le Tableau 2.2 liste les définitions importantes liées à la multi-fidélité, basées sur celles présentées par Sen et al. (2018).

Expression	Définition
$\phi \in [0, 1]$	Valeur d'une fidélité.
$f(x, \phi)$	Valeur de l'objectif suite à une évaluation au point $x$ et à une fidélité $\phi$ .
$\lambda : [0, 1] \mapsto \mathbb{R}^+$	Fonction coût monotone croissante selon $\phi$ .

TABLEAU 2.2 Définitions liées à la multi-fidélité

La plus haute fidélité correspond à  $\phi = 1$ , alors que  $\phi = 0$  est la plus basse. Ce que ces fidélités signifient concrètement pour un procédé doit être établi par l'utilisateur. Pour faire suite à l'exemple de la simulation Monte-Carlo donné plus haut,  $\phi = 1$  pourrait correspondre à  $2 \cdot 10^5$  tirages alors que  $\phi = 0$  pourrait correspondre à 100 tirages. Ces correspondances dépendent du problème auquel fait face l'utilisateur. La troisième définition du Tableau 2.2 impose que  $\lambda$  augmente avec la fidélité. Cette définition sera utile pour garantir l'optimalité d'un sous-problème d'optimisation détaillé à la Section 4, où une diminution de la fidélité doit impliquer une diminution du temps d'évaluation. La variable  $\lambda$  est définie comme le coût à des fins de généralité, or, pour ce projet de maîtrise, cette variable correspond au temps d'évaluation.

En BBO, une boîte noire peut être multi-fidélité. Le cas échéant, à chaque évaluation de la boîte noire, celle-ci doit recevoir en entrée non-seulement un point  $x \in X$ , mais également une valeur de fidélité  $\phi \in [0, 1]$  avec laquelle effectuer l'évaluation. En ce qui concerne  $\lambda$ , plusieurs champs de la recherche opérationnelle assument que cette fonction est connue. Toutefois, comme les méthodes d'optimisation de boîtes noires ont pour but de considérer un cadre très général, les hypothèses posées sur  $\lambda$  sont les plus faibles possible. De ce fait, mis à part la monotonie donnée par la définition du Tableau 2.2, aucune hypothèse n'est posée sur  $\lambda$ .

Pour lister quelques nouveaux exemples, une boîte noire multi-fidélité peut être constituée d'une simulation par éléments finis où la taille du maillage dicte la fidélité, ou d'un algorithme d'apprentissage machine dont on veut optimiser les hyperparamètres et où la fidélité est donnée par le nombre d'itérations d'apprentissage tel que montré par Wu et al. (2020). Pareillement, il pourrait s'agir d'une expérience en laboratoire où une fidélité plus basse est atteinte en se « dépêchant » d'effectuer les manipulations au risque d'effectuer des erreurs. Ce dernier exemple est plutôt excentrique, mais il illustre à quel point l'optimisation de boîtes

noires multi-fidélité est large par sa nature très générale. Bien sûr, toute boîte noire multi-fidélité peut être vue comme une boîte noire qui ne prend que  $x$  comme paramètre en fixant la fidélité à priori.

Dans la littérature récente, Belakaria et al. (2020) développent une méthode pour approximer un front de Pareto en multi-objectif multi-fidélité. Aussi, Wang et al. (2022) proposent une extension à l’algorithme d’arbre de recherche optimiste hiérarchique (*hierarchical optimistic tree search*) pour utiliser les évaluations à basse fidélité.

## 2.4 Boîtes noires stochastiques

La multi-fidélité est souvent le produit d’un processus stochastique. En revanche, les méthodes d’optimisation de boîtes noires stochastiques s’appliquent très différemment que les méthodes d’optimisation de boîtes noires multi-fidélité. En optimisation stochastique, il n’est généralement pas assumé que le contrôle sur la précision existe, seulement que les sorties d’une boîte noire sont bruitées. En optimisation multi-fidélité, la précision est contrôlable par un ou plusieurs paramètres mappés sur la fidélité qui appartient à  $[0, 1]$ . Les approches sont donc plutôt distinctes.

Tel que mentionné plus tôt, ce travail de maîtrise testera la méthode proposée avec une implémentation de l’algorithme MADS, pour lequel des adaptations pour les boîtes noires stochastiques existent déjà, mais qui ne s’appliquent pas à la multi-fidélité. Par exemple, Audet et al. (2018b) proposent une version de MADS nommé Robust-MADS qui construit une fonction lisse à partir des évaluations bruitées pour approximer l’objectif. Cette méthode adresse le problème des boîtes noires contaminées par un bruit numérique. Plus récemment, Audet et al. (2021) proposent StoMADS, une autre version de MADS, plutôt adaptée à un cadre stochastique, en imposant un seuil sur la probabilité que l’estimé d’une évaluation soit précise et une condition sur la variance de ces évaluations. En opposition aux autres méthode qui ne posent aucune hypothèse sur le bruit, Alarie et al. (2021b) suggèrent une modification à MADS qui assume un bruit Gaussien. L’algorithme proposé fait tendre vers zéro l’écart-type des estimations des valeurs de l’objectif, et comprend des propriétés de convergence fort intéressantes.

L’algorithme de Nelder-Mead a également vu plusieurs modifications pour être applicable à des systèmes stochastiques. Parmi les premiers à suggérer de telles modifications, Barton et Ivey, Jr. (1996) proposent d’ajouter des réévaluations de l’itéré courant, et une atténuation de l’étape de réduction. Aussi, Anderson et Ferris (2001) proposent un nouvel algorithme de BBO très semblable à Nelder-Mead, où les simplexes sont remplacés par d’autres structures



plus adaptées dans un contexte stochastique. Plus récemment, Chang (2012) propose de remplacer l'étape de réduction de Nelder-Mead par une recherche aléatoire adaptative.

## 2.5 PRIAD

PRIAD est le projet d'Hydro-Québec qui a motivé ce projet de recherche. Le simulateur qui sera vu comme une boîte noire est décrit par Komljenovic et al. (2019). PRIAD y est présenté comme une approche qui vise à maximiser la valeur des actifs d'Hydro-Québec, leur durabilité et leur résilience. Ces aspects sont abordés d'une part d'un point de vue technique, et d'autre part d'un point de vue organisationnel dans l'entreprise. Le simulateur d'intérêt pour ce projet est constitué de quatre modules appelés séquentiellement, illustrés à la Figure 2.2.

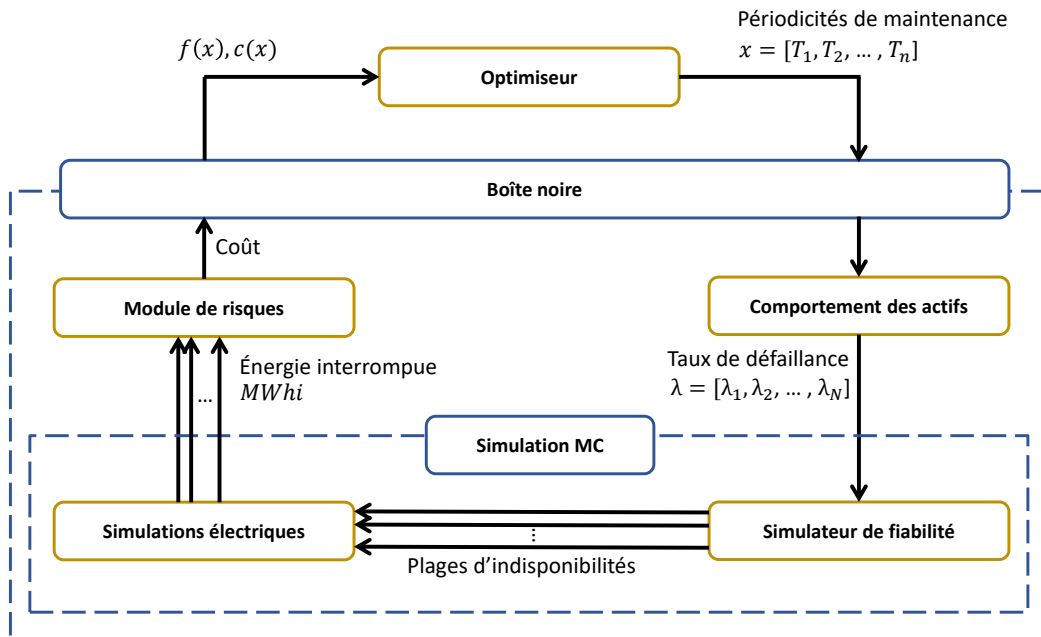


FIGURE 2.2 Représentation graphique du simulateur de PRIAD

D'abord, l'optimiseur envoie un point à la boîte noire. Ce dernier correspond aux périodicités de maintenance de chaque type de maintenance pour chaque famille d'équipement. L'ensemble des familles d'équipement est de taille  $N$ , et comprend les transformateurs, les disjoncteurs, etc. Il est également possible de diviser une famille en plusieurs familles, si par exemple il est jugé pertinent d'attribuer différentes périodicités aux transformateurs de distribution, de moyenne puissance et de haute puissance. Le premier module, qui est détaillé

par Côté et al. (2020), modélise les mécanismes de dégradation selon les périodicités données, pour produire un taux de défaillance  $\lambda$  pour chaque famille d'équipement. Ce module utilise un modèle basé sur la physique théorique et les connaissances des experts du domaine pour prédire de premiers taux de défaillances. Ce modèle est alors comparé à des données historiques pour calibrer le modèle. Cette calibration est effectuée par un modèle d'optimisation analytique. Les taux de défaillance, qui sont des probabilités que chaque équipement encoure une défaillance chaque année permettent au second module d'effectuer plusieurs simulations de fiabilités. Chaque simulation correspond à un tirage par la méthode MC, et renvoie une plage d'indisponibilités de chaque équipement individuel sur un horizon de 40 ans. Gaha et al. (2021) détaillent davantage les deux derniers modules. Pour chacun des tirages MC, l'avant-dernier module effectue une simulation de l'écoulement de puissance optimal, compte tenu du fait que certains équipements sont indisponibles durant certaines périodes. De plus, l'opération du réseau, c'est-à-dire, entre autres, les changements d'états de disjoncteurs ou de sectionneurs en cas de défaillances ou de maintenance planifiée respectivement, est également simulée. Chaque simulation renvoie l'énergie non-livrée sur le réseau, indiquée par l'acronyme *MWhi* qui signifie Mégawatts heure interrompus. Finalement, le module de risque traduit les sorties des différents modules en termes monétaires. Ce module prend en compte le coût de la maintenance planifiée, le nombre de pannes concourantes, le coût de l'énergie non livrée, l'impact des défaillances sur les infrastructures touchées, et bien plus. Par exemple, un grand coût est associé à une simulation où un hôpital est dépourvu d'électricité pendant une grande durée.

Komljenovic et al. (2019) montrent également le Tableau 2.3. Ce dernier présente des estimations des temps qu'une seule évaluation de la boîte noire requiert. Ces temps sont obtenus en supposant qu'aucune parallélisation n'est effectuée, et un nombre limité d'équipements sont simulés (il y a beaucoup plus que 2000 équipements électriques sur le réseau entier d'Hydro-Québec).

Cas	Nombre d'équipements	Nombre de tirages MC	Nombre d'évènements simulés	Temps d'une évaluation (jours)
Sous-station	80	$2 \times 10^4$	$5.6 \times 10^6$	0.23
Corridor	400	$1 \times 10^5$	$1.4 \times 10^8$	5.83
Réseau	2000	$5 \times 10^5$	$3.5 \times 10^6$	145

TABLEAU 2.3 Temps d'évaluation du simulateur de PRIAD par Komljenovic et al. (2019)

D'un point de vue plus général, Murphy et al. (2005) décrivent comment déterminer ce qui constitue la vérité dans le domaine d'étude des simulations de fiabilité, en comparant des approches théoriques et expérimentales. Aussi, Murphy et al. (2001) indiquent quelques règles générales à suivre pour conduire différents types de simulations de fiabilité.

## 2.6 Réduction du temps d'optimisation

En pratique, les méthodes de BBO sont souvent appliquées à des boîtes noires très coûteuses à évaluer. Cette section présente quelques des travaux qui visent la réduction du temps total d'optimisation. Razavi et al. (2010) divisent les types d'approches en quatre catégories :

- Le développement d'algorithmes particulièrement pour les problèmes coûteux. Le développement de NOMAD tombe dans cette catégorie ;
- L'utilisation du parallélisme pour lancer plusieurs simulations simultanément. Cette avenue a été explorée par Audet et al. (2008) et Alarie et al. (2018) dans le contexte du logiciel NOMAD ;
- L'identification des évaluations à simplement éviter. La méthode proposée dans ce mémoire tombe dans cette catégorie ;
- L'utilisation de modèles peu coûteux qui imitent la boîte noire, tel qu'abordés par Conn et al. (1997).

Du côté des algorithmes plus efficaces, Huot et al. (2019) suggèrent une approche qui combine MADS à l'algorithme de recherche dimensionné dynamiquement : *Dynamically Dimensioned Search* (DDS) pour former un nouvel algorithme hybride. Ce dernier a été testé sur la calibration de modèles hydrologiques et une amélioration significative du temps d'optimisation a été observé. Aussi, Wetter et Polak (2005) proposent une méthode qui s'apparente aux modèles pour résoudre des systèmes d'équations différentielles complexes. La méthode débute par effectuer de nombreuses approximations pour explorer le domaine des variables, puis la précision, et donc le temps d'évaluation aussi, sont augmentés graduellement. Les premières itérations sauvent suffisamment de temps pour observer une réduction significative du temps total pour les systèmes d'équations testés.

Finalement, en ce qui concerne l'évitement des mauvaises évaluations, Alarie et al. (2022) proposent deux algorithmes de hiérarchisation d'une boîte noire sous contraintes. La publication s'intéresse au cas où au cours d'une évaluation, différentes étapes donnent différentes informations intermédiaires. Il est alors possible de poser des contraintes sur ces informations intermédiaires. La boîte noire peut ainsi être considérée comme un ensemble de sous-boîtes

noires, où chacune renvoie la valeur d'une contrainte, sauf une qui donne la valeur de l'objectif. Le premier algorithme est une variation sur la BE. Durant la phase de réalisabilité, la fonction de violation des contraintes est calculée après chaque sous-boîte noire, en ne considérant que les contraintes évaluées. Dès que cette fonction atteint une valeur plus grande que celle de l'itéré courant, l'évaluation est interrompue. Durant la phase d'optimalité, dès qu'une sous-boîte noire renvoie une contrainte qui n'est pas satisfaite, l'évaluation est interrompue. Du temps est alors sauvé sur les évaluations de mauvaise qualité. Le second algorithme repose sur une hiérarchie des sous-boîtes noires. D'abord, les contraintes sont classées de la plus difficile à satisfaire à la plus facile. L'algorithme comporte également une phase de réalisabilité, où valeur de la première contrainte est minimisée, sans contraintes. Une fois qu'un point est tel que l'objectif est nul (la première contrainte est satisfaite), un second problème est optimisé, où la première contrainte est la seule contrainte et la seconde est l'objectif, et ainsi de suite. À la phase d'optimalité, le problème redevient le problème original en ajoutant l'objectif, et le principes des interruptions du premier algorithme est appliqué. Cette approche permet de traiter en priorité les contraintes particulièrement difficiles à satisfaire qui peuvent faire perdre beaucoup de temps à un algorithme. Ces deux algorithmes ont directement inspiré la méthode présentée dans ce mémoire.

## CHAPITRE 3 ÉTUDE DE L'IMPACT DE LA FIDÉLITÉ

### 3.1 Algorithme d'évaluations hiérarchiques interrompues

exemple avec priad où : phi à chaque étape et à chaque "batch" de simulations. simulations électriques c'est le plus lourd. elles sont effectuées en parallèles mais il y en a trop pour all in one shot alors il y a une queue de tâches.

Contraintes posées sur chaque étape. Histogrammes updatés au fur et à mesure que simulations électriques faites, puis contraintes sur moyennes, variances, max, min, etc. de VoLL, de N-1, N-2, etc.

### 3.2 Représentativité des contraintes

### 3.3 Satisfaction des contraintes

### 3.4 Temps d'évaluation

## CHAPITRE 4 PARTITION DES CONTRAINTES

### 4.1 Modèle d'optimisation de la partition

### 4.2 Dépendance entre la satisfaction des contraintes et la fidélité

### 4.3 Réductions de la taille du modèle

in preuve : avec  $\lambda$  du tableau 2.2, optimalité garantie du modèle d'opt analytique. Sans, il peut exister une haute fidélité pas rapport qui a un temps super bas où il faut partitionner ds contraintes pour partition optimale, et il est possible que ma méthode ne le voit pas.

## CHAPITRE 5 RÉSULTATS

Lemyre Garneau (2015)

### 5.1 Algorithmes implémentés

noter que PB/EB est différent de PEB

### 5.2 Résultats numériques

#### 5.2.1 Données à priori

#### 5.2.2 Partition optimale

#### 5.2.3 Optimisations avec NOMAD

#### Comparaison des cinq algorithmes

En assumant une partition basée sur données de qualité. Aka en utilisant des données biaisées, celles qui viennent des points d'une optimisation à  $\phi = 1$  de 1000 evals, en ayant réévalué les points à chaque fidélité de  $\Phi$ .

#### 5.2.4 Comparaison des partitions

Comparaison de différentes façons d'obtenir une partitions de manière réaliste dans une implémentation réelle, donc pas avec les données biaisées de la dernière section.

méthodes comparées : tout à  $\phi = 1$  (cas de base, partition sans analyse à priori), partition basée sur un HL, partition recalculée à chaque T evals.

### 5.3 Discussion

## CHAPITRE 6 CONCLUSION

Texte.

### **6.1 Synthèse des travaux**

Texte.

### **6.2 Limitations de la solution proposée**

### **6.3 Améliorations futures**

à tester avec autres logiciels que nomad et avec autres bb que solar.



## RÉFÉRENCES

S. Alarie, N. Amaioua, C. Audet, S. Le Digabel, et L.-A. Leclaire, “Selection of variables in parallel space decomposition for the mesh adaptive direct search algorithm”, *Les cahiers du GERAD*, Rapp. tech. G-2018-38, 2018. En ligne : [http://www.optimization-online.org/DB\\_HTML/2018/06/6660.html](http://www.optimization-online.org/DB_HTML/2018/06/6660.html)

S. Alarie, C. Audet, P.-Y. Bouchet, et S. Le Digabel, “Optimisation of stochastic blackboxes with adaptive precision”, *SIAM Journal on Optimization*, vol. 31, no. 4, pp. 3127–3156, 2021. DOI : 10.1137/20M1318894. En ligne : <https://dx.doi.org/10.1137/20M1318894>

S. Alarie, C. Audet, A. Gheribi, M. Kokkolaras, et S. Le Digabel, “Two decades of blackbox optimization applications”, *EURO Journal on Computational Optimization*, vol. 9, p. 100011, 2021. DOI : 10.1016/j.ejco.2021.100011. En ligne : <https://doi.org/10.1016/j.ejco.2021.100011>

S. Alarie, C. Audet, P. Jacquot, et S. Le Digabel, “Hierarchically constrained blackbox optimization”, *Operations Research Letters*, vol. 50, no. 5, pp. 446–451, 2022. DOI : 10.1016/j.orl.2022.06.006. En ligne : <https://doi.org/10.1016/j.orl.2022.06.006>

E. Anderson et M. Ferris, “A Direct Search Algorithm for Optimization with Noisy Function Evaluations”, *SIAM Journal on Optimization*, vol. 11, no. 3, pp. 837–857, 2001. DOI : 10.1137/S1052623496312848. En ligne : <https://dx.doi.org/10.1137/S1052623496312848>

C. Audet et J. Dennis, Jr., “A Progressive Barrier for Derivative-Free Nonlinear Programming”, *SIAM Journal on Optimization*, vol. 20, no. 1, pp. 445–472, 2009. DOI : 10.1137/070692662. En ligne : <https://dx.doi.org/10.1137/070692662>

—, “Mesh Adaptive Direct Search Algorithms for Constrained Optimization”, *SIAM Journal on Optimization*, vol. 17, no. 1, pp. 188–217, 2006. DOI : 10.1137/040603371. En ligne : <https://dx.doi.org/Doi:10.1137/040603371>

C. Audet et W. Hare, *Derivative-Free and Blackbox Optimization*, série Springer Series in Operations Research and Financial Engineering. Cham, Switzerland : Springer, 2017. DOI : 10.1007/978-3-319-68913-5. En ligne : <https://dx.doi.org/10.1007/978-3-319-68913-5>

C. Audet, J. Dennis, Jr., et S. Le Digabel, *Parallel Space Decomposition of the Mesh Adaptive Direct Search algorithm*, série The GERAD newsletters (Eleven articles published in leading Journals), 2008, vol. 5, no. 2, p. 3. En ligne : [https://www.gerad.ca/Sebastien.Le.Digabel/Publications/bulletin\\_gerad\\_2009-01-22.pdf](https://www.gerad.ca/Sebastien.Le.Digabel/Publications/bulletin_gerad_2009-01-22.pdf)

—, “Globalization strategies for Mesh Adaptive Direct Search”, *Computational Optimization and Applications*, vol. 46, no. 2, pp. 193–215, 2010. DOI : 10.1007/s10589-009-9266-1. En ligne : <https://dx.doi.org/10.1007/s10589-009-9266-1>

C. Audet, A. Conn, S. Le Digabel, et M. Peyrega, “A progressive barrier derivative-free trust-region algorithm for constrained optimization”, *Computational Optimization and Applications*, vol. 71, no. 2, pp. 307–329, 2018. DOI : 10.1007/s10589-018-0020-4. En ligne : <https://dx.doi.org/10.1007/s10589-018-0020-4>

C. Audet, A. Ihaddadene, S. Le Digabel, et C. Tribes, “Robust optimization of noisy blackbox problems using the Mesh Adaptive Direct Search algorithm”, *Optimization Letters*, vol. 12, no. 4, pp. 675–689, 2018. DOI : 10.1007/s11590-017-1226-6. En ligne : <https://dx.doi.org/10.1007/s11590-017-1226-6>

C. Audet, K. Dzahini, M. Kokkolaras, et S. Le Digabel, “Stochastic mesh adaptive direct search for blackbox optimization using probabilistic estimates”, *Computational Optimization and Applications*, vol. 79, no. 1, pp. 1–34, 2021. DOI : 10.1007/s10589-020-00249-0. En ligne : <https://doi.org/10.1007/s10589-020-00249-0>

C. Audet, A. Batailly, et S. Kojtych, “Escaping Unknown Discontinuous Regions in Blackbox Optimization”, *SIAM Journal on Optimization*, vol. 32, no. 3, pp. 1843–1870, 2022. DOI : 10.1137/21M1420915. En ligne : <https://doi.org/10.1137/21M1420915>

C. Audet, S. Le Digabel, V. Rochon Montplaisir, et C. Tribes, “Algorithm 1027 : NOMAD version 4 : Nonlinear optimization with the MADS algorithm”, *ACM Transactions on Mathematical Software*, vol. 48, no. 3, pp. 35 :1–35 :22, 2022. DOI : 10.1145/3544489. En ligne : <https://dx.doi.org/10.1145/3544489>

C. Audet, S. Le Digabel, L. Salomon, et C. Tribes, “Constrained Blackbox Optimization with the NOMAD Solver on the COCO Constrained Test Suite”, dans *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, série GECCO ’22. New York, NY, USA : Association for Computing Machinery, 2022, pp. 1683–1690. DOI : 10.1145/3520304.3534019. En ligne : <https://doi.org/10.1145/3520304.3534019>

I. Bajaj, S. Iyer, et M. Faruque Hasan, “A trust region-based two phase algorithm for constrained black-box and grey-box optimization with infeasible initial point”, *Computers & Chemical Engineering*, vol. 116, pp. 306–321, 2018, multi-scale Systems Engineering – in memory & honor of Professor C.A. Floudas. DOI : <https://doi.org/10.1016/j.compchemeng.2017.12.011>. En ligne : <https://www.sciencedirect.com/science/article/pii/S0098135417304404>

V. Balabanov et G. Venter, “Multi-fidelity optimization with high-fidelity analysis and low-fidelity gradients”, dans *10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, 2004, p. 4459.

R. Barton et J. Ivey, Jr., “Nelder-Mead simplex modifications for simulation optimization”, *Management Science*, vol. 42, no. 7, pp. 954–973, 1996. DOI : [10.1287/mnsc.42.7.954](https://doi.org/10.1287/mnsc.42.7.954). En ligne : <https://dx.doi.org/10.1287/mnsc.42.7.954>

S. Belakaria, A. Deshwal, et J. Doppa, “Multi-Fidelity Multi-Objective Bayesian Optimization : An Output Space Entropy Search Approach”, *CoRR*, vol. abs/2011.01542, 2020. En ligne : <https://arxiv.org/abs/2011.01542>

K. Chang, “Stochastic nelder-mead simplex method - a new globally convergent direct search method for simulation optimization”, *European Journal of Operational Research*, vol. 220, no. 3, pp. 684–694, 2012. DOI : [10.1016/j.ejor.2012.02.028](https://doi.org/10.1016/j.ejor.2012.02.028). En ligne : <https://dx.doi.org/10.1016/j.ejor.2012.02.028>

A. Conn, K. Scheinberg, et P. Toint, “On the convergence of derivative-free methods for unconstrained optimization”, dans *Approximation Theory and Optimization : Tributes to M.J.D. Powell*, M. Buhmann et A. Iserles, édés. Cambridge, United Kingdom : Cambridge University Press, 1997, pp. 83–108. En ligne : <http://us.cambridge.org/Titles/catalogue.asp?isbn=0521581907>

A. Côté, O. Blancke, S. Alarie, A. Dems, D. Komljenovic, et D. Messaoudi, “Combining Historical Data and Domain Expert Knowledge Using Optimization to Model Electrical Equipment Reliability”, dans *2020 International Conference on Probabilistic Methods Applied to Power Systems (PMAPS)*, Liege, Belgium, 2020, pp. 1–6. DOI : [10.1109/PMAPS47429.2020.9183620](https://doi.org/10.1109/PMAPS47429.2020.9183620). En ligne : <https://dx.doi.org/10.1109/PMAPS47429.2020.9183620>

K. Dzahini, M. Kokkolaras, et S. Le Digabel, “Constrained stochastic blackbox optimization using a progressive barrier and probabilistic estimates”, Les cahiers du

GERAD, Rapp. tech. G-2020-60, 2022, to appear in *Mathematical Programming*. DOI : 10.1007/s10107-022-01787-7. En ligne : [http://www.optimization-online.org/DB\\_HTML/2020/11/8101.html](http://www.optimization-online.org/DB_HTML/2020/11/8101.html)

M. Gaha, B.Chabane, D.Komljenovic, A. Côté, C. Hébert, O. Blancke, A. Delavari, et G. Abdul-Nour, “Global Methodology for Electrical Utilities Maintenance Assessment Based on Risk-Informed Decision Making”, *Sustainability*, vol. 13, no. 16, p. 9091, 2021.

A. Galántai, “Convergence of the Nelder-Mead method”, *Numerical Algorithms*, vol. 90, no. 3, pp. 1043–1072, 2022.

N. Gould et P. Toint, “Nonlinear programming without a penalty function or a filter”, *Mathematical Programming*, vol. 122, no. 1, pp. 155–196, 2010. DOI : 10.1007/s10107-008-0244-7. En ligne : <https://dx.doi.org/10.1007/s10107-008-0244-7>

N. Hansen, A. Auger, R. Ros, O. Mersmann, T. Tušar, et D. Brockhoff, “COCO : a platform for comparing continuous optimizers in a black-box setting”, *Optimization Methods and Software*, vol. 36, no. 1, pp. 114–144, 2021. DOI : 10.1080/10556788.2020.1808977. En ligne : <https://doi.org/10.1080/10556788.2020.1808977>

P.-L. Huot, A. Poulin, C. Audet, et S. Alarie, “A hybrid optimization approach for efficient calibration of computationally intensive hydrological models”, *Hydrological Sciences Journal*, vol. 64, no. 10, pp. 1204–1222, 2019. DOI : 10.1080/02626667.2019.1624922. En ligne : <https://dx.doi.org/10.1080/02626667.2019.1624922>

W. Huyer et A. Neumaier, “Global optimization by multilevel coordinate search”, *Journal of Global Optimization*, vol. 14, pp. 331–355, 1999.

D. Komljenovic, D. Messaoudi, A. Côté, M. Gaha, L. Vouligny, S. Alarie, et O. Blancke, “Asset Management in Electrical Utilities in the Context of Business and Operational Complexity”, dans *World Congress on Resilience, Reliability and Asset Management*, Singapore, 2019, pp. 148–151.

S. Le Digabel, “Algorithm 909 : NOMAD : Nonlinear Optimization with the MADS algorithm”, *ACM Transactions on Mathematical Software*, vol. 37, no. 4, pp. 44 :1–44 :15, 2011. DOI : 10.1145/1916461.1916468. En ligne : <https://dx.doi.org/10.1145/1916461.1916468>

S. Le Digabel et S. Wild, “A Taxonomy of Constraints in Simulation-Based

Optimization”, Les cahiers du GERAD, Rapp. tech. G-2015-57, 2015. En ligne : [http://www.optimization-online.org/DB\\_HTML/2015/05/4931.html](http://www.optimization-online.org/DB_HTML/2015/05/4931.html)

M. Lemyre Garneau, “Modelling of a solar thermal power plant for benchmarking blackbox optimization solvers”, Mémoire de maîtrise, Polytechnique Montréal, 2015. En ligne : <https://publications.polymtl.ca/1996/>

J. Müller, “MISO : mixed-integer surrogate optimization framework”, *Optimization and Engineering*, vol. 17, no. 1, pp. 177–203, 2016. DOI : 10.1007/s11081-015-9281-2. En ligne : <https://dx.doi.org/10.1007/s11081-015-9281-2>

K. Murphy, C. Carter, et L. Wolfe, “How long should I simulate, and for how many trials? A practical guide to reliability simulations”, dans *Annual Reliability and Maintainability Symposium. 2001 Proceedings. International Symposium on Product Quality and Integrity (Cat. No. 01CH37179)*. IEEE, 2001, pp. 207–212.

K. Murphy, A. Malerich, et C. Carter, “What is truth? A practical guide to comparing reliability equation answers to simulation results”, dans *Annual Reliability and Maintainability Symposium, 2005. Proceedings*. IEEE, 2005, pp. 439–444.

J. Nelder et R. Mead, “A simplex method for function minimization”, *The Computer Journal*, vol. 7, no. 4, pp. 308–313, 1965. DOI : 10.1093/comjnl/7.4.308. En ligne : <http://comjnl.oxfordjournals.org/content/7/4/308.abstract>

Y. Ozaki, S. Watanabe, et M. Onishi, “Accelerating the Nelder-Mead method with predictive parallel evaluation”, dans *6th ICML Workshop on Automated Machine Learning*, vol. 185, 2019, p. 186.

T. Papalexopoulos, C. Tjandraatmadja, R. Anderson, J. Vielma, et D. Belanger, “Constrained Discrete Black-Box Optimization using Mixed-Integer Programming”, dans *Proceedings of the 39th International Conference on Machine Learning*, série Proceedings of Machine Learning Research, K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, et S. Sabato, édés., vol. 162. PMLR, 17–23 Jul 2022, pp. 17295–17322. En ligne : <https://proceedings.mlr.press/v162/papalexopoulos22a.html>

N. Ploskas et N. Sahinidis, “Review and comparison of algorithms and software for mixed-integer derivative-free optimization”, *Journal of Global Optimization*, vol. 82, no. 3, pp. 433–462, 2022. DOI : 10.1007/s10898-021-01085-0. En ligne : <https://dx.doi.org/10.1007/s10898-021-01085-0>

S. Razavi, B. Tolson, L. Matott, N. Thomson, A. MacLean, et F. Seglenieks, “Reducing the computational cost of automatic calibration through model preemption”, *Water Resources Research*, vol. 46, no. 11, 2010.

L. Rios et N. Sahinidis, “Derivative-free optimization : a review of algorithms and comparison of software implementations”, *Journal of Global Optimization*, vol. 56, no. 3, pp. 1247–1293, 2013. DOI : 10.1007/s10898-012-9951-y. En ligne : <https://dx.doi.org/10.1007/s10898-012-9951-y>

R. Sen, K. Kandasamy, et S. Shakkottai, “Multi-fidelity black-box optimization with hierarchical partitions”, dans *International conference on machine learning*. PMLR, 2018, pp. 4538–4547.

V. Torczon, “PDS : Direct search methods for unconstrained optimization on either sequential or parallel machines”, Rice University, Department of Computational and Applied Mathematics, Mail Stop 134, 6100 Main Street, Houston, Texas 77005-1892, Rapp. tech. 92–09, 1992.

J. Wang, D. Basu, et I. Trummer, “Procrastinated Tree Search : Black-Box Optimization with Delayed, Noisy, and Multi-Fidelity Feedback”, *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 9, pp. 10 381–10 390, Jun. 2022. DOI : 10.1609/aaai.v36i9.21280. En ligne : <https://ojs.aaai.org/index.php/AAAI/article/view/21280>

M. Wetter et E. Polak, “Building design optimization using a convergent pattern search algorithm with adaptive precision simulations”, *Energy and Buildings*, vol. 37, no. 6, pp. 603–612, 2005.

J. Wu, S. Toscano-Palmerin, P. Frazier, et A. Wilson, “Practical Multi-fidelity Bayesian Optimization for Hyperparameter Tuning”, dans *Proceedings of The 35th Uncertainty in Artificial Intelligence Conference*, série Proceedings of Machine Learning Research, R. P. Adams et V. Gogate, édés., vol. 115. PMLR, 22–25 Jul 2020, pp. 788–798. En ligne : <https://proceedings.mlr.press/v115/wu20a.html>

## ANNEXE A DÉMO

Texte de l'annexe A. Remarquez que la phrase précédente se termine par une lettre majuscule suivie d'un point. On indique explicitement cette situation à  $\text{\LaTeX}$  afin que ce dernier ajuste correctement l'espacement entre le point final de la phrase et le début de la phrase suivante.

## ANNEXE B ENCORE UNE ANNEXE

Texte de l'annexe B en mode «landscape».



**ANNEXE C    UNE DERNIÈRE ANNEXE**

Texte de l'annexe C.