

Parte I

THÈSE EN FRANÇAIS - RÉSUMÉ

Préface au Résumé Français

Ce résumé a suivi la structure du rapport de la thèse. Les sections de chaque chapitre ont été décrites dans ce résumé en utilisant les expressions mathématiques et les tables plus importantes du texte principal du rapport, lorsque les figures restent encore dans le text en portugais. Dans les pages préliminaires de ce rapport, sur le titre "Contribuições"(page ix), ils ont été listés les contributions du travail de recherche comme les articles publiés, les formulations proposées et les algorithmes développés, par exemple. Aussi avant les chapitres, sur le titre "Notações"(page xi), ils ont été décrits différentes notations formelles utilisées pour dénoter les variables, les ensembles mathématiques, les opérations, les structures de données, d'entre autres sort de paramètres qui ont mérité spécifications standard pour mieux les distinguer dans le texte.

Le premier chapitre a présenté comment les sources des incertitudes peut s'associer à un système d'optimisation en général. Comment les incertitudes changent la formulation multi-objectif originale en établissent la formulation multi-objectif robuste, qui représente le problème principal à être résolu par cette thèse. L'importance du sujet de la thèse, des objectifs, des méthodologies de recherche ont été aussi discutés.

Le chapitre deux a décrit un état de l'art en recherche qui a été la base par la modélisation des algorithmes développés. Ils ont présenté des langages issus des communautés d'Optimisation Multi-Objectif, d'Algorithmes Évolutionnaires et d'Analyse par Intervalles qui simplifient la compréhension et la construction des algorithmes proposés pour l'optimisation robuste.

Le troisième chapitre a présenté d'abord tous les concepts de base qui permet comprendre les algorithmes développés. Ensuite, les algorithmes ont été introduits avec plusieurs détails. Aussi, une technique de niche par intervalles a été proposée et des métriques pour évaluer les résultats trouvés par les méthodes.

Le chapitre quatre a été caractérisé par les tests qui ont validé les algorithmes. Cinq fonctions tests ont été adaptées de l'optimisation multi-objectif traditionnelle pour l'optimisation robuste. Les résultats ont été discutés.

Le chapitre cinq a présenté une application expérimentale pour exemplifier comment peut être fait un traitement d'un problème complet par les méthodes proposées. Ainsi, pas à pas, une étude de cas d'un problème réel a été réalisée dès la conception à la résolution. À travers ce cas, ils ont été montrés des avantages des algorithmes et misent en évidence quelques difficultés techniques rencontrées durant la réalisation du processus d'optimisation.

Finalement, dans le chapitre six ont été discutées les conclusions et perspectives.

Chapitre 1 - Introduction

En cette section, il a été présenté un problème d'optimisation multi-objectif à travers d'un exemple d'une fonction pour démarrer les discussions diverses sur la présence des incertitudes en systèmes d'optimisation multi-objectif. L'exemple a été écrit comme :

$$\left\{ \begin{array}{l} \min_{\mathbf{x} \in \mathbb{R}^2} \quad f_1(\mathbf{x}) = c_1 x_1^2 + c_2 x_2^2 ; \\ \quad \quad \quad f_2(\mathbf{x}) = c_3 x_1 + c_4 x_2. \\ \\ s.c. : \\ \quad \quad \quad g(\mathbf{x}) = x_1 + x_2 \leq 0. \end{array} \right. \quad (1.1)$$

où x_1 et x_2 sont les variables, c_1 à c_4 sont les coefficients, $g(x)$ est la fonction de contrainte à satisfaire et $f_1(x_1, x_2)$ et $f_2(x_1, x_2)$ sont les objectifs à optimiser. Une méthode pour résoudre un problème comme (1.1) a le but d'obtenir un ensemble de solutions, dit solutions Pareto, que sont non dominées entre se et avec une distribution uniforme dans le Front de Pareto. Les caractéristiques de non dominance et de distribution sont mesurées dans l'espace des objectifs.

En analysant (1.1), c'est clair que ce système d'optimisation n'a pas considéré la présence des incertitudes, que sûrement sont présents n'importe quel est le problème d'optimisation. À la suite, il a été imaginé d'autres scénarios où les incertitudes existent autant pour mesurer et pour construire x_1 et x_2 que des imprecisions pour quantifier les coefficients c_i , $i = 1 \dots 4$. Maintenant, dans ce nouveau scénario le problème formulé (1.1) est devenu inconsistant, et peut-être les solutions qu'ont été trouvés n'étaient pas valides. Ainsi, il y a besoin d'une expression, appelée *formulation robuste*, pour caractériser le scénario avec incertitudes et pour offrir des conditions aux méthodes pour trouver les solutions optimales.

1.1 - Importance

Les problèmes réels d'optimisation nécessitent souvent de la prise en compte de plusieurs critères. Formellement, un système d'optimisation multi-objectif traditionnel sans contraintes est souvent défini comme :

$$\min_{\mathbf{x} \in \mathbb{R}} \mathbf{f}(\mathbf{x}) \quad (1.2)$$

où $\mathbf{x} \in \mathbb{R}^{n_n}$ est le vecteur des variables du problème, et $\mathbf{f}(\mathbf{x}) : \mathbb{R}^{n_n} \rightarrow \mathbb{R}^{n_f}$ est le vecteur des objectifs à optimiser.

Ensuite, considère que ce système a été affecté par différents types d'incertitudes. Par exemple, les incertitudes peuvent exister dû aux imprécisions pour mesurer les paramètres du modèle et aux caractéristiques physiques du milieu que le système a été situé (pression, température, viscosité, etc). Par conséquent, il y a en général plusieurs possibilités de modéliser un système d'optimisation qui prennent en compte des incertitudes. Cette thèse a utilisé un paramètre \mathbf{p} , dans l'expression du vecteur de fonctions objectif, pour représenter tous les types d'incertitudes. Ainsi, en considérant trois genres de incertitudes, la formulation robuste de $\mathbf{f}(\mathbf{x})$ est définie par :

$$\mathbf{f}(\mathbf{x}, \mathbf{p}) = \mathbf{f}(\mathbf{x}, \epsilon, \varepsilon, \xi), \quad \mathbf{p} \in \mathbb{R}^{n_p}. \quad (1.6)$$

où ϵ et ε représentent les incertitudes d'imprécision de mesurer et de quantifier \mathbf{x} et \mathbf{f} respectivement, et ξ c'est le facteur d'ambiant.

En considérant les discussions antérieures, clairement $\mathbf{f}(\mathbf{x}) \neq \mathbf{f}(\mathbf{x}, \mathbf{p})$. Comme conséquence, les résultats d'optimisation de $\mathbf{f}(\mathbf{x})$ et $\mathbf{f}(\mathbf{x}, \mathbf{p})$ peuvent être aussi différents. En principe, il n'est pas possible de déduire l'influence de \mathbf{p} sur $\mathbf{f}(\mathbf{x}, \mathbf{p})$, sauf dans les cas triviaux. Donc, si les incertitudes sont présentes dans le problème d'optimisation, alors les solutions trouvées utilisant $\mathbf{f}(\mathbf{x})$ peuvent perdre l'attribut optimal.

Il y a plusieurs études dans la littérature sur les méthodes pour résoudre les problèmes multi-objectif traditionnelles comme en (1.2). Mais, il y n'existe pas beaucoup de travaux qui ont traité l'ambien d'optimisation avec des incertitudes. Donc, il a été jugé important de considérer les problèmes multi-critères avec incertitudes et développer des méthodes pour chercher cette sorte de solutions, dit solutions robustes.

1.2 - Objectifs

En considérant le paramètre de projet $\mathbf{x} \in \mathbf{X} \subseteq \mathbb{R}^{n_x}$ et le paramètre d'incertitude $\mathbf{p} \in \mathbf{P} \subseteq \mathbb{R}^{n_p}$. Le but principal de cette thèse a été l'étude et la résolution des problèmes d'optimisation multi-objectif robuste définis comme :

$$\begin{aligned} \min_{\mathbf{x} \in \mathbf{X}} \quad & \max_{\mathbf{p} \in \mathbf{P}} \mathbf{f}(\mathbf{x}, \mathbf{p}) \\ \text{s.c. :} \quad & \mathbf{g}(\mathbf{x}, \mathbf{p}) \leq \mathbf{0}, \quad \mathbf{p} \in \mathbf{P}, \end{aligned} \quad (1.7)$$

où $\mathbf{f}(\mathbf{x}, \mathbf{p}) : \mathbb{R}^{n_x} \times \mathbb{R}^{n_p} \mapsto \mathbb{R}^{n_f}$ est le vecteur des fonctions objectif, et $\mathbf{g}(\mathbf{x}, \mathbf{p}) : \mathbb{R}^{n_x} \times \mathbb{R}^{n_p} \mapsto \mathbb{R}^{n_g}$ est le vecteur des fonctions de contraintes.

À partir de ces études ils ont été proposés trois algorithmes multi-critères pour résoudre le problème (1.7). Ces algorithmes ont été développés en utilisant des techniques basées en méthodes intervalles et évolutionnaires.

Les buts secondaires ont été :

- réaliser une étude d'état de l'art sur la théorie de l'Optimisation Multi-Objectif et l'Analyse par Intervalles mettant en évidence les principales caractéristiques des algorithmes de chaque domaine ;
- présenter d'autres philosophies sur l'optimisation robuste et situer la formulation adoptée dans ce document ;
- développer quelques techniques par intervalles pour l'aider les procédés internes aux algorithmes proposés, et à algorithmes pour l'optimisation multi-objectif en général ;
- proposer cinq problèmes test pour l'optimisation robuste multi-objectif ;
- valider les algorithmes avec les problèmes test ;
- développer un problème multi-objectif réel et appliquer les algorithmes développés pour le résoudre.

1.3 - Limitations

Il y a plusieurs façons de traiter et d'interpréter l'optimisation robuste, et aussi de développer les algorithmes pour résoudre cette sorte de problème. Donc, il a eu besoin d'établir quelques limitations ce travail. La thématique générale de cette thèse a été limitée à :

- employer optimisation seulement au format de minimisation - la maximisation peut être obtenu par analogie ;
- utiliser seulement intervalles réels - les intervalles complexes, circulaires et rectangulaires n'ont pas été traités ;
- définir intervalle comment un numéro incertain qui existe dans les limites inférieures et supérieures d'un intervalle - numéro intervalle n'a pas été considéré ;
- présenter seulement les concepts fondamentaux d'Optimisation Multi-Objectif, Algorithmes Évolutionnaires Multi-Objectif et Analyse par Intervalles concernant aux items importants pour le développement des algorithmes proposés - des études plus profondes sur les sujets n'ont pas été réalisées ;

- développer des problèmes test en utilisant opérations et fonctions élémentaires comment $+$, $-$, \times , \div , \exp , \sin , \cos , etc - la bibliothèque mathématique des intervalles codifiés pendant la thèse ne contient pas opérateurs et fonctions plus complexes ;
- développer algorithmes que n'utilise pas l'information de dérivatives dans le processus de recherche - les algorithmes que utilisent ce genre d'information sont abondants dans la littérature ;
- utiliser la structure des algorithmes génétiques pour élaborer la structure d'algorithme évolutionnaire hybride proposé - d'autres sortes d'algorithmes évolutionnaires ne seront pas utilisées ;
- utiliser les méthodes traditionnelles pour la sélection, mutation et croisement - la diversité de ces méthodes est grand et donc impossible d'analyser ;
- utiliser la codification binaire pour représenter les paramètres de l'algorithme évolutionnaire proposé.

1.4 - Méthodologie

Dans cette section, il a été expliqué, comment cette étude a été développée. Les pas ont été :

- l'étude d'état de l'art sur les sujets de la recherche ;
- développement d'une librairie des codes sources pour effectuer les opérations avec intervalles ;
- construction des standards, définitions et du modèle du Problème d'Optimisation Robuste ;
- développement des algorithmes ;
- élaboration des fonctions test ;
- application des algorithmes développés dans un problème réel.

1.5 - Planification

Le Chapitre 2 a présenté les fondements théoriques sur l'Optimisation Multi-Objectif, l'Algorithmes Évolutionnaires et l'Analyse par Intervalles, indispensables au développement des algorithmes présentés sur le Chapitre 3. Au-delà de les méthodes pour l'optimisation robuste, le Chapitre 3 a introduit aussi quelques autres techniques par intervalles comment une technique de niche et une métrique pour mesurer la distribution de points dans de la Frontière de Pareto. Dans le Chapitre 4, des problèmes test ont été proposés afin de valider les algorithmes proposés. Plusieurs tests ont été faits. La formulation et la résolution d'un problème multi-objectif pour la synthèse des contrôleurs PID ont été faites sur le Chapitre 5. Finalement, sur le Chapitre 6, les conclusions ont été présentées.

1.6 - Considérations

Cette section a présenté le résumé du chapitre.

Chapitre 2 - Révision Bibliographique

2.1 - Optimisation Multi-Objectif

Dans cette section plusieurs questions d'optimisation multi-objectif traditionnelle ont été présentées comment de la notion dominance, les notations diverses pour les espaces de recherche et de solutions réalisables d'un problème, et des classifications pour les algorithmes multi-objectif. Le contenu principal de cette section est présenté à la suite.

Les problèmes mono-objectif sont caractérisés par l'existence d'un optimum global. Contrairement, l'optimisation multi-objectif cherche un ensemble de solutions optimales qui reflètent plusieurs compromis entre tous les critères. Ainsi, l'idée de trouver un optimum unique a été remplacée par la recherche d'un ensemble de éléments qualifiés de solutions qui appartenant à frontière de Pareto, ou simplement, de solutions Pareto optimales. Pour mieux comprendre les concepts d'optimisation multi-objectif et de solution Pareto, considère d'abord la notation $\mathbf{X} \subseteq \mathbb{R}^{n_x}$ pour représenter l'espace des variables, \mathbf{x}_1 et \mathbf{x}_2 pour noter deux vecteurs de variables, et un vecteur de

fonctions objectif $\mathbf{f}(\mathbf{x}) : \mathbb{R}^{n_x} \rightarrow \mathbb{R}^{n_f}$ pour évaluer les deux vecteurs. Si le décideur préfère \mathbf{x}_1 à \mathbf{x}_2 , il dit que “ \mathbf{x}_1 domine \mathbf{x}_2 ”. Formellement, l’opérateur de dominance \preceq est défini dans l’espace d’objectif comment :

$$\mathbf{f}(\mathbf{x}_1) \preceq \mathbf{f}(\mathbf{x}_2), \text{ si } \mathbf{f}(\mathbf{x}_1) \leq \mathbf{f}(\mathbf{x}_2), \mathbf{f}(\mathbf{x}_1) \neq \mathbf{f}(\mathbf{x}_2). \quad (2.3)$$

Par analogie, il a été défini aussi l’opérateur \succeq , et les cas stricts \prec et \succ . Maintenant, une fois que l’opérateur de dominance a été défini, le prochain pas est de décrire le problème multi-objectif avec contrainte, donc :

$$\begin{aligned} \min_{\mathbf{x} \in \mathbf{X}} \quad & \mathbf{f}(\mathbf{x}) \\ \text{s.c. : } \quad & \mathbf{g}(\mathbf{x}) \leq \mathbf{0}. \end{aligned} \quad (2.4)$$

L’ensemble solution de ce problème peut être défini comment :

$$\mathbf{X}^* = \{\mathbf{x}^* \in \mathbf{X}' \mid \nexists \mathbf{x} \in \mathbf{X}', \mathbf{f}(\mathbf{x}) \preceq \mathbf{f}(\mathbf{x}^*)\}, \quad (2.6)$$

où $\mathbf{X}' = \{\mathbf{x} \in \mathbf{X} \mid \mathbf{g}(\mathbf{x}) \leq \mathbf{0}\}$ est l’espace réalisable et \mathbf{X}^* est l’ensemble de solutions acceptables du problème.

Le choix d’une solution particulière parmi l’ensemble \mathbf{X}^* dépend du décideur et de la méthode par lui employée. À critère du décideur, il peut utiliser trois sortes de méthodes. Le premier réalise le choix de la solution *à priori*, c’est à dire, ce genre d’algorithme s’agit de mettre des pondérations sur chaque objectif en transformant un problème multi-objectif en un problème mono-objectif. Il y a des inconvénients en problèmes où la frontière de Pareto est convexe. Les algorithmes qui relèvent du deuxième type de méthode, font le choix *à posteriori*, c’est-à-dire, la décision est faite directement sur le \mathbf{X}^* . Finalement, il y a des algorithmes où le choix est fait de façon *progressif*. Dans ce cas, les préférences sont faites pendant le procédé d’optimisation.

Références : Marler et Arora [80], Andersson [4], Sawaragi et al. [100], Miettinen [84], Yu [125], Osyczka [92], Koski et Silvennoinen [74], Zionts [128], Yu [124], Zeleny [126], Waltz [122], Takahashi et al. [117], Benayoun et al. [9], Steuer [114], Messac et al. [83], Messac et al. [82], Das et Dennis [28], Nash Arbitrário, Davis [30], Rao et Freiheit [95], Messac [81].

2.2 - Algorithmes Évolutionnaires

Dans cette section, ils ont été discutés la diversité des Algorithmes Évolutionnaires, leur structure algorithmique, les techniques pour maintenir la diversité dans la frontière de Pareto, et des métriques pour mesurer ces performances.

Les Algorithmes Évolutionnaires sont inspirés sur des analogies avec les phénomènes biologiques, par exemple, du processus de sélection naturelle (Algorithme Génétique), du comportement des fourmis (Algorithmes de Colonies de Fourmis) et du déplacement d'un groupe d'oiseaux (Optimisation par Essaims Particulaires). En cette thèse, la métaheuristique d'optimisation choisie pour le développement d'algorithme évolutionnaire hybride a été l'Algorithme Génétique. Ainsi, le vocabulaire utilisé pour expliquer cet algorithme a été calqué sur la théorie de l'évolution et de la génétique. Par exemple, le terme *chromosome* est associé à représentation discrète d'une variable ; les *individus* d'une *population* représente les solutions potentielles. Aussi, la sélection, la reproduction, le croisement et la mutation, parmi d'autres analogies, sont employés comme techniques pour traiter les solutions en guidance d'algorithme hybride à la recherche de l'ensemble optimal. La Tab.2.3 a décrit un exemple d'algorithme génétique multi-objectif.

Les informations principales des dernières subsection de cette section sont présentées ensuite. La Sous-section 2.2.3 a cité et décrit plusieurs méthodes évolutionnaires multi-objectif. La Sous-section 2.2.4 a présenté et a discuté les paramètres communément trouvés en algorithmes génétiques multi-objectif. La Sous-section 2.2.5 a mis en question l'importance d'utiliser la stratégie élitiste dans le contexte multi-objectif.

La Sous-section 2.2.6 a discuté cinq métriques par mesurer les performances et la qualité d'ensemble solution des algorithmes multi-objectif en général. Des métriques présentées ont été comparées avec la métrique développé dans cette thèse à la Section 3.7. En spécial, une métrique trouvée dans la littérature est utilisée aussi dans la phase de validation. Elle s'appelle TE (Taxe d'Erreur) et est basée sur la connaissance préalable d'un ensemble de comparaison, par exemple \mathbf{Y}^* . La métrique TE est défini par :

$$TE = \frac{\sum_{i=1}^{|\mathbf{A}|} e_i}{|\mathbf{A}|}, \quad (2.14)$$

soient $|\mathbf{A}|$ le numéro de solutions d'ensemble \mathbf{A} , trouvé par la méthode que se désire évaluer, et e_i est tel que :

$$\begin{cases} e_i = 1, & \text{si } e_i \in \mathbf{A} \setminus \mathbf{Y}^* ; \\ e_i = 0, & \text{autrement.} \end{cases} \quad (2.15)$$

La conclusion d'un élément de \mathbf{A} appartient aussi de \mathbf{Y}^* est déterminé par une distance euclidienne, que représente la précision ε_{TE} de la métrique TE.

La Sous-section 2.2.7 disserte sur la formation de niche, qui sont tech-

niques que permettent aux algorithmes génétiques de rechercher plusieurs portions d'espace en simultanéité et donc d'éviter que l'algorithme converge à un optimum local.

Références : Dorigo et Stützle [37], Kennedy et Eberhart [69], Laumanns et al. [76], Deb [32], Knowles [72], Holland [57], Goldberg [45], Soares [107], Gao et al. [44], Schaffer [101], Fosenca et Fleming [42], Murata et Ishibuchi [89], Horn et al. [58], Goldberg et Richardson [46], Srinivas et Deb [113], Deb et al. [34], Coello et Sierra [21], Knowles et Corne [73], Coello et Pulido [20], Knowles [72], Soares et al. [112], Kenneth et Eberhart [69], zitzler [129], Zitzler et al. [130], Veldhuizen [119], Soares [111], Grefenstette [47], Rump [99], Rudolph [97], Czyzak [24], Hansen et Jaszkievicz [51], Schott [102], et Zitzler et Thiele [131].

2.3 - Analyse par Intervalles

Tout d'abord, cette section a présenté des avantages de faire le traitement de données utilisant intervalles. Des avantages ont été basés sous la garantie de propagation correcte d'un numéro incertain contenu dans un intervalle. Aussi, des inconvénients, comme la lenteur des méthodes par intervalles, ont été listés et discutés. Ensuite, un bref historique d'Analyse par Intervalles a été présenté.

Les fondements d'Analyse par Intervalles sont basés sur la théorie des ensembles. Donc, la Sous-section 2.3.2 a défini les opérations fondamentales pour traiter les ensembles, et comme conséquence, les intervalles. Les opérations ont été présentées par les expressions (2.22-2.28), et des définitions et des propriétés des ensembles ont été montrés dans les expressions (2.29-2.32). Les Sous-sections 2.3.3 et 2.3.4 ont présenté les définitions pour les intervalles réels et pour l'arithmétique pour opérer cette sorte de numéro. Pour des raisons pratiques et par simplicité, des définitions ont été réalisées en \mathbb{R} . Par analogie, des expressions peuvent être étendus pour \mathbb{IR}^{n_x} , où n_x représente la dimension du vecteur \mathbf{x} . Un intervalle réelle $[x]$ a été défini comment :

$$[\underline{x}, \bar{x}] = \{x \in \mathbb{R} \mid \underline{x} \leq x \leq \bar{x}\}, \quad \underline{x} \in \mathbb{R} \text{ et } \bar{x} \in \mathbb{R}, \quad (2.33)$$

et les opérations binaires $+$, $-$, $*$ et $/$ comment :

$$[x] \diamond [y] = \{x \diamond y \mid x \in [x], y \in [y]\}. \quad (2.34)$$

Les quatre opérations ont été explicitées dans les expressions (2.35-2.40) ; quelques exemples ont été présentés ensuite. Dans la Sous-section 2.3.5, les

fonctions qui évaluent intervalles ont été définis. Formellement, la fonction intervalle $f([x])$ a été définie comment :

$$f([x]) = [\{f(x) \mid x \in [x]\}]. \quad (2.41)$$

Si une variable apparaît plus qu'une fois dans une fonction intervalle, alors le résultat de cette fonction est un intervalle plus conservateur, car les incertitudes des intervalles sont propagées. Ce problème est appelé de *dépendance* et il n'est pas traité pour l'arithmétique des intervalles. La Sous-section 2.3.6 a expliqué et a exemplifié la dépendance en détaille.

Le *vecteur intervalle* ou une *boîte* est le type donné plus commun dans cette thèse. Il a été décrit dans la Sous-section 2.3.7. Formellement, une boîte $[\mathbf{x}] \in \mathbb{IR}^{n_x}$ a été définie comment :

$$[\underline{\mathbf{x}}, \bar{\mathbf{x}}] = [x_1] \times \dots \times [x_n], \quad (2.44)$$

Normalement, la précision des méthodes intervalles sont mesurées par *longueur* des intervalles. La longueur $w([\mathbf{x}]) : \mathbb{IR}^{n_x} \rightarrow \mathbb{R}$ a été définie comment :

$$w([\mathbf{x}]) = \max_{i=\{1, \dots, n_x\}} w([x_i]). \quad (2.45)$$

Aussi, à la fin d'un procédé d'optimisation, le décideur a besoin de prendre une solution réelle dans un ensemble de boîtes. Une estimation très utilisée a été le point central de la boîte. Ainsi, le *centre* $\mathbf{c}([\mathbf{x}]) : \mathbb{IR}^{n_x} \rightarrow \mathbb{R}^{n_x}$ a été défini par :

$$c_i([\mathbf{x}]) = (\bar{x}_i + \underline{x}_i)/2, \quad i = \{1, \dots, n_x\}. \quad (2.47)$$

Le *volume* de boîtes est une importante mesure dans la technique de niche à être proposée dans le prochaine chapitre. Donc, le volume $v([\mathbf{x}]) : \mathbb{IR}^{n_x} \rightarrow \mathbb{R}$ a été décrit comment :

$$v([\mathbf{x}]) = \prod_{i=\{1, \dots, n_x\}} w([x_i]). \quad (2.47)$$

Plusieurs méthodes intervalles pour l'optimisation commence le processus de recherche par une boîte qui contient tout l'espace de projet, et itération par itération, ces méthodes ont besoin d'examiner portions plus petites d'espace pour améliorer ses précisions d'estimatif, d'entre autres choses. Donc, bissecter d'intervalle se fait nécessaire. La bissection d'une boîte $[\mathbf{x}]$ en deux autres $[\mathbf{x}]_1$ et $[\mathbf{x}]_2$ a été défini comment :

$$[\mathbf{x}]_1 = [x_1] \times \dots \times [x_j, \frac{x_j + \bar{x}_j}{2}] \times \dots \times [x_n], \quad (2.48)$$

$$[\mathbf{x}]_2 = [x_1] \times \dots \times [\frac{x_j + \bar{x}_j}{2}, \bar{x}_j] \times \dots \times [x_n], \quad (2.49)$$

où $j = \min\{i \mid w([x_i]) = w([\mathbf{x}])\}$.

La Sous-section 2.3.8 a décrit les fonctions, appelées *fonctions d'inclusion*, qui est utilisé par les méthodes intervalles en générale. Formellement, considère la fonction $\mathbf{f} : \mathbb{R}^{n_x} \rightarrow \mathbb{R}^{n_f}$. La fonction $[\mathbf{f}] : \mathbb{IR}^{n_x} \rightarrow \mathbb{IR}^{n_f}$ est une fonction d'inclusion pour \mathbf{f} si

$$\forall [\mathbf{x}] \in \mathbb{IR}^{n_x}, \mathbf{f}([\mathbf{x}]) \subseteq [\mathbf{f}]([\mathbf{x}]) \quad (2.51)$$

La Fig.2.4 présente deux exemples de fonctions d'inclusion. Pour une même \mathbf{f} , il y a plusieurs $[\mathbf{f}]([\mathbf{x}])$. Parmi ces fonctions $[\mathbf{f}]([\mathbf{x}])$, il est intéressant d'obtenir des fonctions d'inclusion *fine, convergentes, monotones et minimales*. La fonction $[\mathbf{f}]([\mathbf{x}])$ est définie fine quand :

$$\forall \mathbf{x} \in [\mathbf{x}], \mathbf{f}(\mathbf{x}) = [\mathbf{f}](\mathbf{x}). \quad (2.52)$$

La fonction $[\mathbf{f}]([\mathbf{x}])$ est définie convergente quand :

$$\lim_{k \rightarrow \infty} w([\mathbf{x}]_k) = 0 \Rightarrow \lim_{k \rightarrow \infty} w([\mathbf{f}]([\mathbf{x}]_k)) = 0. \quad (2.53)$$

La fonction $[\mathbf{f}]([\mathbf{x}])$ est définie monotone quand :

$$[\mathbf{x}]_k \subset [\mathbf{x}]_{k-1} \Rightarrow [\mathbf{f}]([\mathbf{x}]_k) \subset [\mathbf{f}]([\mathbf{x}]_{k-1}). \quad (2.54)$$

Finalement, la fonction $[\mathbf{f}]([\mathbf{x}])$ est définie minimale quand elle est la mineuse boîte que contient $\mathbf{f}(\mathbf{x})$, $\forall \mathbf{x} \in [\mathbf{x}]$. Les Fig. 2.5 et Fig. 2.6 ont démontré ces caractéristiques.

Dans les méthodes intervalles, une fonction d'inclusion que trop utilisée est la *fonction d'inclusion naturelle* qui est obtenue en remplaçant la variable \mathbf{x} par l'intervalle $[\mathbf{x}]$, et en remplaçant chaque opérateur d'arithmétique conventionnel ou fonction par son homologue pour les intervalles.

La Sous-section 2.3.9 ont décrit la Logique booléenne pour les intervalles. Dans cette logique, il y a quatre possibilités pour l'ensemble booléen intervalle \mathbb{IB} , sont-elles :

$$\mathbb{IB} = \{\emptyset, 0, 1, [0, 1]\}. \quad (2.55)$$

où \emptyset est la absence de connexion entre les ensembles en question, et $[0, 1]$ signifie le résultat booléen peut être vrai ou faux. Pour mieux comprendre la

Logique Booléenne Intervalle, observez les exemples dans les expressions (2.56 - 2.61).

La Sous-section 2.3.10 a présenté comme sont faits les tests booléens. Ces genres de tests sont importants pour guider les algorithmes dans ses procédés de recherche. Par exemple, les tests booléens peuvent être utilisés pour vérifier si une variable en particulière est réalisable ou non via évaluation de fonctions de contraintes, écrites sur la forme de tests d'inclusion. Un exemple de test a été présenté dans la formule (2.63).

Un *sous-pavage* de \mathbb{R}^n est un ensemble de pavés \mathbb{R}^n ne se recouvrant pas. La Fig. 3.5 a montré un exemple des sous-pavages que ne se recouvrant pas pour représenter une région selon les contraintes définies en (3.31). La Sous-section 2.3.11 a présenté comment les sous-pavages sont organisés en structures de données du type piles et files.

La Sous-section 2.3.12 a cité SIVIA qu'une méthode d'inversion ensembliste que s'intéresse à la caractérisation d'espace $\mathbf{X} \in \mathbb{R}^{n_x}$ donné son image $\mathbf{Y} \in \mathbb{R}^{n_f}$. SIVIA utilise plusieurs concepts d'Analyse par Intervalles ont décrit dans les sous-sections précédents comme bissecter, test d'inclusion et sous-pavages. De nouveau, donné les fonctions de contraintes en (3.31), SIVIA a trouvé l'ensemble des solutions réalisables, que sont montrées en gris dans la figure.

La Sous-section 2.3.13 a présenté les inéquations intervalles. La notion d'inéquations intervalles est important pour la compréhension du traitement de las fonctions de contraintes.

Références : Hayes [53], Daumas [29], Moore [85], Jaulin et al. [63], Kearfott et Kreinovich [68], Hansen et Walster [50], Carreras et Walker [15], Morales et Son [88] em robótica, Soares et al. [109], Siouris et al. [104], Healt [54], Alefeld et Mayer [2], Sunaga [115], Hansen [48], Neumaier [90], Kearfott [67], Hanson [52], Kahan [66], Walster et al. [121], Jaulin et al. [63], Hickey et al. [56], Vaccaro [118], Moore [87], Walster [50], Benhamou et Older [10], et Chabert et Jaulin [17].

2.4 - Optimisation Multi-Objectif Intervalle

Cette section a montré l'évolution des méthodes qui utilisent l'Analyse par Intervalles dans le contexte d'optimisation, et spécifiquement dans le domaine d'optimisation multi-objectif. Ils ont été cités plusieurs travaux de recherche en cette direction. Après la part historique, cette section a discuté des points de vue *optimiste* et *pessimiste* pour la comparaison entre intervalles. La Fig. 2.8 a présenté la comparaison de dominance des intervalles ;

le graphique centrale a présenté le résultat en utilisant un point de vue optimiste et le graphique à droite un point de vue pessimiste. Dans les algorithmes proposés, le pessimisme a été préféré comme stratégie à être utilisé durant les comparaisons de dominance, car il maintient des boîtes qui peuvent contenir des solutions pour le problème en question.

Références : Moore [85], Skelboe [105], Moore [86], Alefeld [1], Hansen [49], Skelboe [106], Ichida et Fujii [60], Hansen et Walster [50], Ichida et Fujii [61], Mahato et Bhunia [78], Casado et al. CasadoGarcíaMartínezSergeyev2003, Csallener et al.[23], Sengupta et Pal [103], Ruetsch [98], Barichard et Hao [6], Benhamou et Older [10], et Jaulin [62].

2.5 - Considérations

Cette section a présentée le résumé du chapitre.

Chapitre 3 - Algorithmes Intervalles Déterministes et Évolutionnaire pour l'Optimisation Robuste Multi-Objectif

Au début du chapitre, pour introduire les algorithmes intervalles, quelques définitions de robustesse d'autres domaines de recherche ont été présentées. Par exemple, la notion de robustesse en Ingénierie du Logiciel a été associé à tolérance aux erreurs, lorsqu'en biologie, la robustesse c'est une caractéristique d'un individu de maintenir ses fonctionnalités même quand l'ambient change. Dans le contexte d'optimisation, la robustesse d'une solution peut être interprétée comme la capacité de maintenir sa réponse au système, même s'il y a des perturbations. Cependant, on peut aussi définir la robustesse d'une solution en regardant seulement sa performance du *scénario du pire cas*. Dans la littérature, l'optimisation robuste a été décrite de deux façons : *stochastique* et *déterministe*. Les méthodes basées sur probabilités utilisant techniques d'échantillonnage pour mesurer la robustesse d'une solution. Donc, la qualité du résultat dépend de la qualité d'échantillonnage. D'autre part, les méthodes déterministes calculent la robustesse d'une solution considérant toutes les interférences des incertitudes. Dans plusieurs études, les intervalles ont été utilisés pour réaliser la tâche de traitement des incertitudes. C'est donc l'optimisation robuste envisagée par ce travail : traiter

le problème robuste multi-objectif au moyen des intervalles pour calculer le scénario du pire cas.

Références : Kitano [71], De Vale et Koopman [35], Du et Chen [38], Parkinson et al. [93], Kouvelis et Yu [75], Roy [96], Dias et Clímaco [36], Beyer et Sendhoff [12], Jin et Branke [65], Deb et Gupta [33], et Soares et al. [108].

3.1 - Bref Historique de l' Optimisation Robuste

Cette section a exposé brièvement quelques travaux généralement cités comme importants pour la démarche de l'optimisation robuste. Les premières études qu'ont pris en compte des incertitudes apparaissaient dans les années 50 en travaux de statistique par améliorer la production agricole, par réaliser la programmation linéaire sous incertitudes et avec contraintes et par améliorer qualité en ingénierie dans le cadre du développement de produits dès la phase de conception. Dès ce début, l'optimisation robuste a évolué en plusieurs directions, par exemple, pour traiter les problèmes convexes, les systèmes discrets et les systèmes dynamiques.

Références : Fisher [41], Dantzig [26], Charnes et Cooper [18], Taguchi [116], Ben-Tal et Nemirovski [7], Ben-Tal et Nemirovski [8], El Ghaoui et al. [40], El Ghaoui et Calafiori [39], Kouvelis et Yu [75], Bertsimas et Sim [11], Mareschal [79], et Dias et Clímaco [36].

3.2 - Concepts et Notation

Les relations de dominance sont utilisées par les méthodologies multi-objectif pour réaliser les tâches de classification, sélection et élimination de solutions dans ses procédés de recherche. Donc, il est important définir des opérateurs pour établir la supériorité ou l'infériorité d'une solution ou d'autre. Une fois que les algorithmes de cette thèse utilisent des données réelles et intervalles simultanément, il est nécessaire redéfinir l'opérateur \preceq comme montré ensuite.

Définition 16 Considère $\mathbf{u} \in \mathbb{R}^{n_n}$, $\mathbf{v} \in \mathbb{R}^{n_n}$, $[\mathbf{t}] \in \mathbb{IR}^{n_n}$ et $[\mathbf{z}] \in \mathbb{IR}^{n_n}$. L'opérateur de comparaison \preceq a été redéfini comment :

$$\mathbf{u} \preceq \mathbf{v} \Leftrightarrow \mathbf{u} \leq \mathbf{v}, \mathbf{u} \neq \mathbf{v}; \quad (3.1)$$

$$\mathbf{u} \preceq [\mathbf{t}] \Leftrightarrow \mathbf{u} \preceq \underline{\mathbf{t}}; \quad (3.2)$$

$$[\mathbf{z}] \preceq [\mathbf{t}] \Leftrightarrow \bar{\mathbf{z}} \preceq \underline{\mathbf{t}}. \quad (3.3)$$

Utilisant la notion de dominance, il est possible de distinguer, dans un ensemble de vecteurs n'importe quel, deux groupes et un terme importants pour l'optimisation robuste : un groupe composé pour les vecteurs plus petits, autre composé pour les vecteurs plus grands et un vecteur qui représente le point idéal de maximisation. La définition de chacun est décrite ensuite.

Définition 17 Considère l'ensemble de vecteurs $\mathbf{U} = \{\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_m\}$, $\mathbf{U} \subseteq \mathbb{R}^{n_n}$. Les plus petits vecteurs de \mathbf{U} définissent l'ensemble Υ_{\preceq} décrit comment :

$$\Upsilon_{\preceq}(\mathbf{U}) = \{\mathbf{u} \in \mathbf{U} \mid \nexists \mathbf{u}' \in \mathbf{U}, \mathbf{u}' \preceq \mathbf{u}\}. \quad (3.5)$$

Par analogie, les plus grands vecteurs de \mathbf{U} composent l'ensemble Υ_{\succeq} décrit comme :

$$\Upsilon_{\succeq}(\mathbf{U}) = \{\mathbf{u} \in \mathbf{U} \mid \nexists \mathbf{u}' \in \mathbf{U}, \mathbf{u} \preceq \mathbf{u}'\}. \quad (3.6)$$

Définition 18 Considère l'ensemble de vecteurs $\mathbf{U} = \{\mathbf{x}, \mathbf{y}, \dots, \mathbf{z}\} \in \mathbb{R}^{n_n}$. Le point idéal de maximisation $\mathbf{u}^{\max} \in \mathbb{R}^{n_n}$ de l'ensemble \mathbf{U} est :

$$u_i^{\max}(\mathbf{U}) = \max_i \{x_i, y_i, \dots, z_i\}, \quad i = 1, \dots, n_n. \quad (3.7)$$

Pour faire suite à cette analyse, un théorème a été formulé pour présenter la relation entre \mathbf{u}^{\max} et Υ_{\succeq} . Ce théorème amène des conséquences importantes pour l'optimisation robuste décrite sur les prochaines sections.

Théorème 1 Considère $\mathbf{x} \in \mathbf{X} \subseteq \mathbb{R}^{n_x}$, $\mathbf{f}(\mathbf{x}) : \mathbb{R}^{n_x} \mapsto \mathbb{R}^{n_f}$ et $\mathbf{U} = \{\mathbf{u} \in \mathbb{R}^{n_f} \mid \forall \mathbf{x} \in \mathbf{X}, \mathbf{u} = \mathbf{f}(\mathbf{x})\}$, $\mathbf{U} \neq \emptyset$. Donc,

$$\Upsilon_{\succeq}(\mathbf{U}) \neq \{\mathbf{u}^{\max}(\mathbf{U})\} \Rightarrow \mathbf{u}^{\max}(\mathbf{U}) \notin \mathbf{U}. \quad \blacksquare \quad (3.8)$$

Preuve : Assume $\mathbf{u}^{\max}(\mathbf{U}) \in \Upsilon_{\succeq}(\mathbf{U})$. Utilisant (3.6) et (3.7) on a conclu que $\Upsilon_{\succeq}(\mathbf{U})$ est l'ensemble unitaire $\Upsilon_{\succeq}(\mathbf{U}) = \{\mathbf{u}^{\max}(\mathbf{U})\}$. Par définition, en utilisant seulement (3.7) on a conclu que $\forall \mathbf{u} \in \mathbf{U}$, $\mathbf{u}^{\max}(\mathbf{U}) \succeq \mathbf{u}$. Donc, si $\mathbf{u}^{\max}(\mathbf{U}) \notin \Upsilon_{\succeq}(\mathbf{U})$, alors $\mathbf{u}^{\max}(\mathbf{U}) \notin \mathbf{U}$.

3.3 - Le Problème Multi-Objectif Robuste

Comme il a été précisé dans l'introduction de ce chapitre, la notion de robustesse enveloppe différents concepts. Pour ce travail, il a été choisi la philosophie de la robustesse qu'il prenne en compte le pire cas de chaque solution évalué. Donc, le problème d'optimisation robuste considéré peut être défini comment présenté ensuite.

Considère le vecteur de variables de projet $\mathbf{x} \in \mathbf{X} \subseteq \mathbb{R}^{n_x}$ et le vecteur de paramètres d'incertitude $\mathbf{p} \in \mathbf{P} \subseteq \mathbb{R}^{n_p}$. En définissant le vecteur de fonctions objectif par $\mathbf{f}(\mathbf{x}, \mathbf{p}) : \mathbb{R}^{n_x} \times \mathbb{R}^{n_p} \mapsto \mathbb{R}^{n_f}$ et le vecteur de contrainte par $\mathbf{g}(\mathbf{x}, \mathbf{p}) : \mathbb{R}^{n_x} \times \mathbb{R}^{n_p} \mapsto \mathbb{R}^{n_g}$, le problème de minimisation robuste multi-objectif peut être écrit comment :

$$\begin{aligned} \min_{\mathbf{x} \in \mathbf{X}} \quad & \max_{\mathbf{p} \in \mathbf{P}} \mathbf{f}(\mathbf{x}, \mathbf{p}) \\ \text{s.c.} \quad & \mathbf{g}(\mathbf{x}, \mathbf{p}) \leq \mathbf{0}. \end{aligned} \quad (3.9)$$

En définissant l'espace réalisable par :

$$\mathbf{X}' = \{\mathbf{x} \in \mathbf{X} \mid \forall \mathbf{p} \in \mathbf{P}, \mathbf{g}(\mathbf{x}, \mathbf{p}) \leq \mathbf{0}\}, \quad (3.10)$$

résoudre (3.9) consiste en trouver l'ensemble de minimiseurs robustes $\mathbf{X}^* \subseteq \mathbf{X}'$ pour le pire cas d'influence des incertitudes. Formellement,

$$\mathbf{X}^* = \{\mathbf{x}^* \in \mathbf{X}' \mid \nexists \mathbf{x} \in \mathbf{X}', \max_{\mathbf{p} \in \mathbf{P}} \mathbf{f}(\mathbf{x}, \mathbf{p}) \preceq \max_{\mathbf{p}' \in \mathbf{P}} \mathbf{f}(\mathbf{x}^*, \mathbf{p}')\}. \quad (3.11)$$

L'espace des objectifs est aussi objet de cette étude. Ainsi, considère les images des fonctions $\mathbf{y} = \mathbf{f}(\mathbf{x}, \mathbf{p})$ et $\mathbf{Y}' = \mathbf{f}(\mathbf{X}', \mathbf{P})$, où :

$$\mathbf{f}(\mathbf{X}', \mathbf{P}) = \bigcup_{\mathbf{x} \in \mathbf{X}', \mathbf{p} \in \mathbf{P}} \mathbf{f}(\mathbf{x}, \mathbf{p}). \quad (3.12)$$

Par conséquent du Théorème 1, si pour quelque solution \mathbf{x}^* le processus de maximisation en (3.9) produire $\mathbf{u}^{\max}(\mathbf{f}(\mathbf{x}^*, \mathbf{p})) \notin \Upsilon_{\succeq}(\mathbf{f}(\mathbf{x}^*, \mathbf{p}))$, donc, est possible que $\mathbf{u}^{\max}(\mathbf{f}(\mathbf{X}^*, \mathbf{p})) \notin \mathbf{Y}'$. Ainsi, pour représenter l'image robuste de toutes les solutions \mathbf{X}^* du problème (3.9) est nécessaire définir un espace que contienne $\mathbf{u}^{\max}(\mathbf{f}(\mathbf{x}^*, \mathbf{p}))$, $\forall \mathbf{x}^* \in \mathbf{X}^*$ et $\forall \mathbf{p} \in \mathbf{P}$. Ceci est fait en replacent l'image $\mathbf{f}(\mathbf{X}', \mathbf{P})$ par la boîte $[\mathbf{Y}]'$ comment :

$$\overline{[\mathbf{Y}]'} = \max_{\mathbf{p} \in \mathbf{P}} f_i(\mathbf{x}, \mathbf{p}), \quad \forall \mathbf{x} \in \mathbf{X}', i = \{1, \dots, n_f\}; \quad (3.13)$$

$$\underline{[\mathbf{Y}]'} = \min_{\mathbf{p} \in \mathbf{P}} f_i(\mathbf{x}, \mathbf{p}), \quad \forall \mathbf{x} \in \mathbf{X}', i = \{1, \dots, n_f\}. \quad (3.14)$$

Maintenant, laissez \mathbf{Y}^* représenter la *frontière de Pareto robuste* ; \mathbf{Y}^+ et \mathbf{Y}^- les *portions au-dessus* et *au-dessous de la frontière robuste*, respectivement. Les images \mathbf{Y}^* , \mathbf{Y}^+ et \mathbf{Y}^- sont liées par la relation :

$$[\mathbf{Y}]' = \mathbf{Y}^- \cup \mathbf{Y}^+, \quad \mathbf{Y}^* \subseteq \mathbf{Y}^+, \quad (3.15)$$

étant :

$$\mathbf{Y}^+ = \{\mathbf{y}^+ \in [\mathbf{Y}]' \mid \exists \mathbf{x} \in \mathbf{X}, \forall \mathbf{p} \in \mathbf{P}, \mathbf{f}(\mathbf{x}, \mathbf{p}) \preceq \mathbf{y}^+\}; \quad (3.16)$$

$$\mathbf{Y}^- = \{\mathbf{y}^- \in [\mathbf{Y}]' \mid \mathbf{y} \in [\mathbf{Y}]', \mathbf{y} \notin \mathbf{Y}^+\}; \quad (3.17)$$

$$\mathbf{Y}^* = \{\mathbf{y}^* \in [\mathbf{Y}]' \mid \nexists \mathbf{y} \in \mathbf{Y}^+, \mathbf{y} \preceq \mathbf{y}^*\}. \quad (3.18)$$

La Fig. 3.1 a présenté un problème robuste de façon générale: le côté à droite illustre les régions \mathbf{Y}^* , \mathbf{Y}^+ et \mathbf{Y}^- ; le côté à gauche montre l'espace de recherche et l'espace des incertitudes. Pour illustrer le problème robuste, cette section a fini avec deux exemples simples.

3.4 - Algorithme Multi-Objectif Robuste par Intervalles I

Le premier algorithme pour traiter le problème (3.9) était appelé de [I]RMOA I (Algorithme Multi-Objectif Robuste par Intervalles I). En fait, cet algorithme ne cherche pas trouver \mathbf{Y}^* , mais il construit une enveloppe discrète sécurisée sur \mathbf{Y}^* . Ainsi, le [I]RMOA I a la finalité de valider d'autres méthodes ; une donnée méthode est considéré valide si l'image de son ensemble trouvé $\mathbf{f}(\mathbf{X}^*)$ appartient à l'enveloppe sur \mathbf{Y}^* , calculé par [I]RMOA I. À la suite, il est présenté comme l'enveloppe était obtenue.

Définition 19 Considère $\mathbf{U} = \{\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_n\}$, $\mathbf{U} \subset \mathbf{Y}^-$. Donc, la représentation discrète de \mathbf{Y}^- lié à \mathbf{U} est :

$$\mathbf{K}^- = \{\mathbf{u}' \in [\mathbf{Y}]' \mid \exists \mathbf{u} \in \Upsilon_{\succeq}(\mathbf{U}), \mathbf{u}' \preceq \mathbf{u}\}. \quad (3.21)$$

Par analogie, si $\mathbf{V} = \{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_m\}$, $\mathbf{V} \subset \mathbf{Y}^+$, donc la représentation discrète de \mathbf{Y}^+ lié à \mathbf{V} est :

$$\mathbf{K}^+ = \{\mathbf{v}' \in [\mathbf{Y}]' \mid \exists \mathbf{v} \in \Upsilon_{\preceq}(\mathbf{V}), \mathbf{v} \preceq \mathbf{v}'\}. \quad (3.22)$$

Définition 20 La frontière supérieure à \mathbf{K}^- et la frontière inférieure à \mathbf{K}^+ sont définies comme :

$$\partial\mathbf{K}^- = \{\mathbf{u} \in \mathbf{K}^- \mid \nexists \mathbf{u}' \in \mathbf{K}^-, \mathbf{u} \preceq \mathbf{u}'\}; \quad (3.23)$$

$$\partial\mathbf{K}^+ = \{\mathbf{v} \in \mathbf{K}^+ \mid \nexists \mathbf{v}' \in \mathbf{K}^+, \mathbf{v}' \preceq \mathbf{v}\}. \quad (3.24)$$

Définition 21 Symbolisant \setminus comment l'opérateur complément, les portions discrètes non obtenues directement de \mathbf{Y}^- et de \mathbf{Y}^+ sont définies par :

$$\Delta\mathbf{K}^- = (\mathbf{Y}^- \setminus \mathbf{K}^-) \cup \partial\mathbf{K}^-; \quad (3.25)$$

$$\Delta\mathbf{K}^+ = (\mathbf{Y}^+ \setminus \mathbf{K}^+) \cup \partial\mathbf{K}^+. \quad (3.26)$$

Théorème 2 Considérant la précision para calculer \mathbf{K}^- et \mathbf{K}^+ , il est vérité l'inclusion suivante :

$$\mathbf{Y}^* \subset \Delta\mathbf{K}^- \cup \Delta\mathbf{K}^+. \quad \blacksquare$$

Preuve: De (3.15), $[\mathbf{Y}]' = \mathbf{Y}^- \cup \mathbf{Y}^+$. De (3.25) et (3.26), $\mathbf{Y}^- = \mathbf{K}^- \cup \Delta\mathbf{K}^-$, $\mathbf{K}^- \cap \Delta\mathbf{K}^- = \partial\mathbf{K}^-$ et $\mathbf{Y}^+ = \mathbf{K}^+ \cup \Delta\mathbf{K}^+$, $\mathbf{K}^+ \cap \Delta\mathbf{K}^+ = \partial\mathbf{K}^+$. En remplaçant \mathbf{Y}^- et \mathbf{Y}^+ en (3.15), résulte en :

$$[\mathbf{Y}]' = (\mathbf{K}^- \cup \Delta\mathbf{K}^-) \cup (\mathbf{K}^+ \cup \Delta\mathbf{K}^+).$$

Mais, $\mathbf{K}^- \cap \Delta\mathbf{K}^- = \partial\mathbf{K}^-$ et $\mathbf{K}^+ \cap \Delta\mathbf{K}^+ = \partial\mathbf{K}^+$ et ainsi, il existe occurrences extras de $\partial\mathbf{K}^-$ et $\partial\mathbf{K}^+$ que peuvent être éliminé. Donc,

$$[\mathbf{Y}]' = ((\mathbf{K}^- \setminus \partial\mathbf{K}^-) \cup \Delta\mathbf{K}^-) \cup ((\mathbf{K}^+ \setminus \partial\mathbf{K}^+) \cup \Delta\mathbf{K}^+);$$

$$[\mathbf{Y}]' = ((\mathbf{K}^- \setminus \partial\mathbf{K}^-) \cup (\mathbf{K}^+ \setminus \partial\mathbf{K}^+)) \cup (\Delta\mathbf{K}^- \cup \Delta\mathbf{K}^+).$$

Les expressions (3.18), (3.21) et (3.22) assurent que $\mathbf{Y}^* \not\subset (\mathbf{K}^- \setminus \partial\mathbf{K}^-)$ et $\mathbf{Y}^* \not\subset (\mathbf{K}^+ \setminus \partial\mathbf{K}^+)$. Donc,

$$\mathbf{Y}^* \subset \Delta\mathbf{K}, \text{ soient } \Delta\mathbf{K} = \Delta\mathbf{K}^- \cup \Delta\mathbf{K}^+. \quad \blacksquare$$

En fait, la Eq. (3.18) garante que $\mathbf{Y}^* \subset \Delta\mathbf{K}^+$. Mais $\Delta\mathbf{K}^+$ seul ne peut pas être directement enveloppé. La Fig. 3.3 illustre la représentation des approximations robustes.

Maintenant, il est présenté la méthode [I]RMOA I. L'algorithme commence à partir d'ensemble d'échantillons (possibles solutions) uniformément distribué dans l'espace de recherche et de façon aléatoire dans l'espace des incertitudes. À la suite, ces échantillons sont évalués par le vecteur des fonctions objectif $\mathbf{f}(\mathbf{x}, \mathbf{p})$, et les résultats sont gardés dans la file $\mathbf{Q}_{\mathbf{fxp}}$. En fait, l'échantillonnage peut se donner directement dans l'espace des objectifs, mais il y a aussi des inconvénients. Donc, chaque élément de $\mathbf{Q}_{\mathbf{fxp}}$ est testé s'il appartient à \mathbf{K}^- ou à \mathbf{K}^+ . Après la classification de tous les éléments, [I]RMOA I a d'information suffisante pour trouver $\Delta\mathbf{K}$. Par exemple, $\Delta\mathbf{K} = [\mathbf{Y}]' \setminus (\mathbf{K}^+ \cup \mathbf{K}^-)$, et comme conséquence $\Delta\mathbf{K} \supset \mathbf{Y}^*$.

L'algorithme [I]RMOA I est présenté dans la Tab.3.1, et ces paramètres sont décrits dans la Tab.3.2. La description schématique des algorithmes FPS et CSC est donnée seulement dans le texte principal du mémoire à la Tab.3.3 et à la Tab.3.4 respectivement. Le plus important c'est savoir que les deux algorithmes FPS et CSC retournent une seule conclusion parmi des options: a) il existe au moins un minimiseur \mathbf{x}^* tel que $\mathbf{f}(\mathbf{x}^*, \mathbf{p}) \preceq \mathbf{y}$; b) il n'existe pas un minimiseur \mathbf{x}^* tel que $\mathbf{f}(\mathbf{x}^*, \mathbf{p}) \preceq \mathbf{y}$. Si FPS et CSC retournent (a), alors \mathbf{y} appartient à $\mathbf{Q}_{\mathbf{K}^+}$; autrement, \mathbf{y} appartient à $\mathbf{Q}_{\mathbf{K}^-}$.

Références: Soares et al. [110] et Jaulin et al. [64].

3.5 - Algorithme Multi-Objectif Robuste par Intervalles II

Différemment du [I]RMOA I, la méthode [I]RMOA II (Algorithme Multi-Objectif Robuste par Intervalles II) cherche envelopper \mathbf{X}^* , lorsque l'autre a l'objectif d'envelopper \mathbf{Y}^* . Spécifiquement, le but principal de l'algorithme [I]RMOA II est trouver tous les sous-pavages $[\mathbf{x}]$ d'espace de recherche initial $[\mathbf{X}]$, selon une précision $\varepsilon_{[\mathbf{x}]}$, tels que le pire cas de ses images est non dominé entre se. Les images sont obtenues en prenant en compte l'action des incertitudes sur chaque $[\mathbf{x}]$. Aussi, il est possible estimer les valeurs réelles des éléments de \mathbf{X}^* en prenant le centre de chaque boîte solution, notée par $\mathbf{c}([\mathbf{x}]^*)$. Avant de décrire le processus de recherche du [I]RMOA II, il est nécessaire présenter quelques notions fondamentales au développement des algorithmes, comment: a) la représentation de l'espace d'incertitude $[\mathbf{P}]$ par sous-pavages; b) le traitement de contraintes; c) la procédure de calculer la fonction d'inclusion et le pire cas; et d) la élimination de région non robuste.

L'espace des incertitudes $[\mathbf{P}]$ est remplacé par sous-pavages $[\mathbf{p}]$ qui ne se recouvrent pas généré d'un numéro finit de bisections de $[\mathbf{P}]$. Formellement,

$$[\mathbf{P}] = \bigcup_{i=1, \dots, n_{sp}} [\mathbf{p}]_i. \quad (3.27)$$

Table 3.1: Algorithme [I]RMOA I.

entrée	sortie
$([\mathbf{X}], [\mathbf{P}], \varepsilon_{\mathbf{x}\#}, \varepsilon_{[\mathbf{x}]})$	$(\mathbf{K}^-, \mathbf{K}^+)$
<ol style="list-style-type: none"> 1 Videz les files $\mathbf{Q}_{\mathbf{fxp}}$, $\mathbf{Q}_{\mathbf{K}^+}$ et $\mathbf{Q}_{\mathbf{K}^-}$. 2 Appelez la fonction $\text{grid}([\mathbf{X}], [\mathbf{P}], \varepsilon_{\mathbf{x}\#})$ pour générer un ensemble de vecteurs \mathbf{x}, uniformément distribué dans $[\mathbf{X}]$. Pour chaque \mathbf{x}, un vecteur \mathbf{p} est choisi par événement aléatoire dans $[\mathbf{P}]$, et évaluez le pair (\mathbf{x}, \mathbf{p}) avec $\mathbf{f}(\mathbf{x}, \mathbf{p})$, enfilez le résultat à $\mathbf{Q}_{\mathbf{fxp}}$. 3 Si $\mathbf{Q}_{\mathbf{fxp}}$ est vide, alors allez au <i>Pas 7</i>, sinon exécute: <ol style="list-style-type: none"> 3.a défilez \mathbf{y} de $\mathbf{Q}_{\mathbf{fxp}}$; 3.b vérifiez si il existe un vecteur $\mathbf{v} \in \mathbf{Q}_{\mathbf{K}^+}$, tel que $\mathbf{v} \preceq \mathbf{y}$ et si il existe un vecteur $\mathbf{u} \in \mathbf{Q}_{\mathbf{K}^-}$, tel que $\mathbf{y} \preceq \mathbf{u}$. Ssi ces conditions seraient fausses, allez au <i>Pas 4</i>. Autrement, allez au <i>Pas 3</i>. 4 Appelez les fonctions $\text{FPS}([\mathbf{P}], [\mathbf{X}], \mathbf{y}, \varepsilon_{[\mathbf{x}]})$ et $\text{CSC}([\mathbf{x}], \mathbf{y}, [\mathbf{P}], \varepsilon_{[\mathbf{x}]})$ par verifier si \mathbf{y} appartient à \mathbf{K}^+ ou si \mathbf{y} appartient à \mathbf{K}^-. 5 Si \mathbf{y} appartient à \mathbf{K}^+, alors supprimez en $\mathbf{Q}_{\mathbf{K}^+}$ les éléments \mathbf{v} tels que $\mathbf{y} \preceq \mathbf{v}$ et enfilez \mathbf{y} à $\mathbf{Q}_{\mathbf{K}^+}$. Autrement, supprimez dans $\mathbf{Q}_{\mathbf{K}^-}$ les éléments \mathbf{u} tels que $\mathbf{u} \preceq \mathbf{y}$ et enfilez \mathbf{y} à $\mathbf{Q}_{\mathbf{K}^-}$. 6 Allez au <i>Pas 3</i>. 7 Calculez \mathbf{K}^-, utilisant (3.21), par chaque $\mathbf{u} \in \mathbf{Q}_{\mathbf{K}^-}$. 8 Calculez \mathbf{K}^+, utilisant (3.22), par chaque $\mathbf{v} \in \mathbf{Q}_{\mathbf{K}^+}$. 9 Retournez \mathbf{K}^- et \mathbf{K}^+. 	

Table 3.2: Symboles du [I]RMOA I.

Symboles	Description	Type
$[\mathbf{X}]$	espace de recherche initial	\mathbb{IR}^{n_x}
$[\mathbf{P}]$	espace d'incertitude initial	\mathbb{IR}^{n_p}
$\mathbf{f}(\mathbf{x}, \mathbf{p})$	fonction objectif robuste	\mathbb{R}^{n_f}
\mathbf{y}	résultat de $\mathbf{f}(\mathbf{x}, \mathbf{p})$	\mathbb{R}^{n_f}
$\varepsilon_{\mathbf{x}\#}$	précision requise pour le grid	\mathbb{R}
$\varepsilon_{[\mathbf{x}]}$	précision requise pour le $[\mathbf{x}]$	\mathbb{R}
$\mathbf{Q}_{\mathbf{fxp}}$	échantillons évalués par $\mathbf{f}(\mathbf{x}, \mathbf{p})$	File: \mathbb{R}^{n_f}
$\mathbf{Q}_{[\mathbf{x}]}$	sous-pavages de $[\mathbf{X}]$	File: \mathbb{IR}^{n_x}
$\mathbf{Q}_{[\mathbf{p}]}$	sous-pavages de $[\mathbf{P}]$	File: \mathbb{IR}^{n_p}
$\mathbf{Q}_{\mathbf{K}^+}$	vecteurs que définissent \mathbf{K}^+	File: \mathbb{R}^{n_f}
$\mathbf{Q}_{\mathbf{K}^-}$	vecteurs que définissent \mathbf{K}^-	File: \mathbb{R}^{n_f}

soient n_{sp} le numéro de sous-pavages généré, en considérant une précision $\varepsilon_{[p]}$. Ces sous-pavages sont enregistrés dans une file $\mathbf{Q}_{[p]}$. La Tab.3.5 présente comme les sous-pavages ont été calculés.

Table 3.5: Décomposition en sous-pavages.

	entrée ($[\mathbf{P}], \varepsilon_{[p]}$)	sortie($\mathbf{Q}_{[p]}$)
1	initialisez $\mathbf{Q}_{[p]}$ avec $[\mathbf{P}]$	
2	si $w(\mathbf{Q}_{[p]}\{1\}) \leq \varepsilon_{[p]}$, alors allez au <i>Pas 5</i>	
3	défilez $[p]$ de $\mathbf{Q}_{[p]}$	
4	bissectez $[p]$, enfiler les résultats à $\mathbf{Q}_{[p]}$ et allez au <i>Pas 2</i>	
5	retournez $\mathbf{Q}_{[p]}$	

Le traitement de contraintes a été fait en utilisant des tests d'inclusion. En considérant que les fonctions de contraintes écrites comme en (3.9), les tests d'inclusion pour ces cas peuvent être formulés comment :

$$[t]([\mathbf{x}]) = \begin{cases} 1, & \text{si } [\mathbf{g}]([\mathbf{x}], [\mathbf{p}]) \subseteq [-\infty, 0], \\ 0, & \text{si } [\mathbf{g}]([\mathbf{x}], [\mathbf{p}]) \subset [0, \infty], \text{ et} \\ [0, 1], & \text{autrement.} \end{cases} \quad (3.30)$$

L'algorithme pour traiter les contraintes est présenté dans la Tab.3.6.

Table 3.6: Algorithme pour traiter les contraintes.

	entrée ($[\mathbf{X}], [\mathbf{P}], \varepsilon_{[x]}, \varepsilon_{[p]}$)	sortie($\mathbf{Q}'_{[x]}, \mathbf{Q}_{[p]}$)
1	initialisez $\mathbf{Q}'_{[x]}$ avec \emptyset , $\mathbf{Q}_{[x]}$ avec $[\mathbf{X}]$ et $\mathbf{Q}_{[p]}$ avec $[\mathbf{P}]$	
2	gérez subpavimentos(entrée: $[\mathbf{P}], \varepsilon_{[p]}$; sortie: $\mathbf{Q}_{[p]}$) (Tab.3.5)	
3	si $\mathbf{Q}_{[x]}$ est vide, alors allez au <i>Pas 8</i>	
4	défilez $[x]$ de $\mathbf{Q}_{[x]}$	
5	si $[\mathbf{g}]([\mathbf{x}], \mathbf{Q}_{[p]}) \subset [0, \infty]$ alors allez au <i>Pas 3</i>	
6	si $[\mathbf{g}]([\mathbf{x}], \mathbf{Q}_{[p]}) \subseteq [-\infty, 0]$ alors enfiler $[x]$ à $\mathbf{Q}'_{[x]}$, allez au <i>Pas 3</i>	
7	si $w([\mathbf{x}]) \geq \varepsilon_{[x]}$, bissectez $[x]$, enfiler les résultats à $\mathbf{Q}_{[x]}$ et allez au <i>Pas 3</i>	
8	retournez $\mathbf{Q}'_{[x]}$ et $\mathbf{Q}_{[p]}$	

L'Exemple Exem.13 a traité un cas avec cinq contraintes. Son résultat est présenté dans la Fig.3.5. En prenant compte cette figure, la région réalisable

($[t]([\mathbf{x}]) = 1$) est en gris clair, la région non réalisable ($[t]([\mathbf{x}]) = 0$) est en blanc, et le gris foncé exhibe une région ambiguë ($[t]([\mathbf{x}]) = [0, 1]$).

Dans cette thèse toutes les fonctions d'inclusion utilisées sont du type naturel. Les calculs des fonctions objectif robustes d'inclusion ont été faites pour une boîte $[\mathbf{x}]$ qui représente une région, et pour un vecteur \mathbf{x} qui représente une possible solution dedans $[\mathbf{x}]$. Donc, en considérant \mathbf{x} la fonction d'inclusion est notée par $[\mathbf{f}](\mathbf{x}, [\mathbf{p}])$, $[\mathbf{p}] \in \mathbf{Q}_{[\mathbf{p}]}$. Ainsi, le pire cas de $[\mathbf{f}](\mathbf{x}, [\mathbf{p}])$, est le point utopie de maximisation $\overline{[\mathbf{f}]}_{\mathbf{x}}^*$ calculé par :

$$\overline{[f_i]}_{\mathbf{x}}^* = \max_{[\mathbf{p}] \in \mathbf{Q}_{[\mathbf{p}]}} \overline{[f_i]}(\mathbf{x}, [\mathbf{p}]), \quad i = \{1, \dots, n_f\}; \quad (3.28)$$

$$[\underline{f_i}]_{\mathbf{x}}^* = \min_{[\mathbf{p}] \in \mathbf{Q}_{[\mathbf{p}]}} \underline{[f_i]}(\mathbf{x}, [\mathbf{p}]), \quad i = \{1, \dots, n_f\}. \quad (3.29)$$

Dans le [I]RMOA II, la *Fonction d'Inclusion Minimale* $\text{MIF}(\mathbf{x}, \mathbf{Q}_{[\mathbf{p}]}) = [\mathbf{f}]_{\mathbf{x}}^*$. Par analogie, $\text{MIF}([\mathbf{x}], \mathbf{Q}_{[\mathbf{p}]}) = [\mathbf{f}]_{[\mathbf{x}]}^*$. La Fig.3.4 illustre des fonctions robuste d'inclusion.

Le [I]RMOA II élimine toutes les boîtes que sont dominées par les pires cas trouvés. Les mieux pires cas \mathbf{y}^* , trouvés dès la première itération, sont enregistrés dans la file $\mathbf{Q}_{\mathbf{Y}^*}$. Maintenant, considère une boîte $[\mathbf{x}]$ n'importe lequel. Le pire cas de $[\mathbf{x}]$ est obtenu de (3.28) et (3.29). Donc,

$$[\mathbf{x}] \text{ est éliminé} \Leftrightarrow \exists \mathbf{y}^* \in \mathbf{Q}_{\mathbf{Y}^*}, \mathbf{y}^* \preceq [\mathbf{f}]_{[\mathbf{x}]}^*. \quad (3.32)$$

À la fin du procédé de recherche de [I]RMOA II, $\mathbf{Q}_{\mathbf{Y}^*}$ est composé par les mieux pires cas de tous les itérations. Ils composent la frontière de Pareto robuste.

Le [I]RMOA II est décrit dans la Tab.3.7. Les paramètres de l'algorithme sont déjà présentés, excepte la file $\mathbf{Q}_{\mathbf{X}^*}$ qui enregistre les centres des boîtes solutions, et la notation $\mathbf{Q}_{[\mathbf{p}]} \{1\}$, où le numéro 1 signifie la première position de la file $\mathbf{Q}_{[\mathbf{p}]}$.

Références: Dao [27].

3.6 - Algorithme Évolutionnaire Multi-Objectif Robuste

Différemment des [I]RMOA I et [I]RMOA II, le [I]RMOEA est un algorithme stochastique que cherche trouver directement l'ensemble optimal \mathbf{X}^* . La structure algorithmique du [I]RMOEA est similaire à la structure

Table 3.7: Algorithme [I]RMOA II.

entrée ($[\mathbf{X}], [\mathbf{P}], \varepsilon_{[\mathbf{x}]}, \varepsilon_{[\mathbf{p}]}$)	sortie ($\mathbf{Q}_{[\mathbf{x}]}, \mathbf{Q}_{\mathbf{X}^*}, \mathbf{Q}_{\mathbf{Y}^*}$)
1	Initialisez $\mathbf{Q}_{\mathbf{X}^*}$, $\mathbf{Q}_{\mathbf{Y}^*}$ et $\mathbf{Q}_{[\mathbf{y}]}$ avec \emptyset .
2	Traitez contraintes(entrée: $[\mathbf{X}], [\mathbf{P}], \varepsilon_{[\mathbf{x}]}, \varepsilon_{[\mathbf{p}]}$; sortie: $\mathbf{Q}_{[\mathbf{x}]}, \mathbf{Q}_{[\mathbf{p}]}$)-Tab.3.6.
3	Si $\mathbf{Q}_{[\mathbf{x}]}$ est vide, alors allez au <i>Pas 13</i> , sinon défilez $[\mathbf{x}]$ de $\mathbf{Q}_{[\mathbf{x}]}$.
4	Appelez MIF($[\mathbf{x}], \mathbf{Q}_{[\mathbf{p}]}$), mettez le résultat en $[\mathbf{f}]_{[\mathbf{x}]}^*$.
5	Si existe un \mathbf{y} en $\mathbf{Q}_{\mathbf{Y}^*}$ tel que $\mathbf{y} \preceq [\mathbf{f}]_{[\mathbf{x}]}^*$ alors allez au <i>Pas 3</i> .
6	Appelez MIF($\mathbf{c}([\mathbf{x}]), \mathbf{Q}_{[\mathbf{p}]}$), mettez le résultat en $[\mathbf{f}]_{\mathbf{x}}^*$.
7	Si existe un \mathbf{y} en $\mathbf{Q}_{\mathbf{Y}^*}$ tel que $\mathbf{y} \preceq [\mathbf{f}]_{\mathbf{x}}^*$,
8	si $w([\mathbf{x}]) > \varepsilon_{[\mathbf{x}]}$, alors bissectez $[\mathbf{x}]$, enfiler les résultats à $\mathbf{Q}_{[\mathbf{x}]}$ et allez au <i>Pas 3</i> ; sinon enfiler $[\mathbf{f}]_{[\mathbf{x}]}^*$ à $\mathbf{Q}_{[\mathbf{y}]}$ et $[\mathbf{x}]$ en $\mathbf{Q}_{[\mathbf{x}]}$.
9	Supprimez tout le \mathbf{y} en $\mathbf{Q}_{\mathbf{Y}^*}$ tel que $[\mathbf{f}]_{\mathbf{x}}^* \preceq \mathbf{y}$ et les respectifs éléments en $\mathbf{Q}_{\mathbf{X}^*}$.
10	Enfiler $[\mathbf{f}]_{\mathbf{x}}^*$ à $\mathbf{Q}_{\mathbf{Y}^*}$ et $\mathbf{c}([\mathbf{x}])$ à $\mathbf{Q}_{\mathbf{X}^*}$.
11	Supprimez tout le $[\mathbf{y}]$ en $\mathbf{Q}_{[\mathbf{y}]}$ tel que $[\mathbf{f}]_{\mathbf{x}}^* \preceq [\mathbf{y}]$ et les respectifs boîtes en $\mathbf{Q}_{[\mathbf{x}]}$.
12	Allez au <i>Pas 3</i> .
13	Retournez $\mathbf{Q}_{[\mathbf{x}]}, \mathbf{Q}_{\mathbf{X}^*}$ et $\mathbf{Q}_{\mathbf{Y}^*}$.

d'un Algorithme Génétique en général, mais internement, les variables sont écrites en utilisant le format réel et intervalles. En conséquence, les processus qui manipulaient ces variables sont aussi adaptés pour traiter les deux sortes de données. Avant de décrire le [I]RMOEA, il est expliqué quelques notions importantes pour mieux lui comprendre, comment la formation de la population, la méthode de sélection, d'élitisme et de niche utilisés.

Considère la notation \mathbf{x}^i pour un individu i , qui dans l'optimisation est le paramètre de recherche. Ainsi, en génération t , la population \mathbf{pop}_t est structuré comme :

$$\mathbf{pop}_t = \{\mathbf{x}^1, \dots, \mathbf{x}^i, \dots, \mathbf{x}^{n_{pop}}\}, \quad (3.34)$$

soient $\mathbf{pop}_t(i)$ l'accès au i -ième individu. La performance de la population \mathbf{des}_t est :

$$\mathbf{des}_t = \{[\mathbf{f}]_{\mathbf{x}^1}^*, \dots, [\mathbf{f}]_{\mathbf{x}^i}^*, \dots, [\mathbf{f}]_{\mathbf{x}^{n_{pop}}}^*\}. \quad (3.35)$$

Par analogie, $\mathbf{des}_t(i)$ représente la performance du i -ième individu. La Tab.3.8 exhibe les détails pour trouver \mathbf{des}_t .

Le traitement de contraintes est fait de la même façon que cela présenté pour le [I]RMOA II en (3.30). La différence c'est que [I]RMOA II a besoin aussi d'évaluer le vecteur \mathbf{x} dans les tests d'inclusion. Ainsi,

$$[t](\mathbf{x}) = \begin{cases} 1, & \text{si } [\mathbf{g}](\mathbf{x}, [\mathbf{p}]) \subseteq [-\infty, 0], \\ 0, & \text{si } [\mathbf{g}](\mathbf{x}, [\mathbf{p}]) \subset [0, \infty], \\ [0, 1], & \text{autrement.} \end{cases} \quad (3.36)$$

La procédure utilisée pour choisir les individus dans la population est la méthode de sélection du type *Roulette*. Dans les problèmes mono-objectif, cette méthode attribue à chaque individu une valeur de pourcentage, selon l'évaluation obtenue de la fonction objectif, qui représente la probabilité d'individu d'être choisi. Dans les méthodes multi-objectif, la méthode Roleta est faible parce qu'il a, normalement, des objectifs qui sont contradictoires. Donc, dans cette thèse, il a été développé une agrégation des objectifs que ne favorisât pas directement un objectif spécifique. Ainsi, la probabilité reste dépendant de la valeur obtenu pour l'agrégation. La *performance utile* $\mathbf{des}'_t(i)$ de chaque individu i est mesurer comment :

$$\mathbf{des}'_t(i) = \frac{1}{f_{niche}^i} + \frac{1}{f_{front}^i}. \quad (3.37)$$

soient f_{niche}^i le valeur de performance selon la quantité d'individus que participent de la même niche que l'individu i , et f_{front}^i selon la frontière de dominance que l'individu i est situé. Par exemple, si l'individu i appartient à la première frontière (de Pareto) alors $f_{front}^i=1$; si il appartient à la deuxième frontière alors $f_{front}^i=2$, etc. La méthode Roleta utilise la valeur $\mathbf{des}'_t(i)/\sum_{j=1}^{n_{pop}} \mathbf{des}'_t(j)$ de probabilité pour chaque individu i .

La codification de l'individu, la méthode de croisement et la méthode de mutation employés dans [I]RMOEA sont classifiés comment classiques. La codification est binaire, le croisement est réalisé avec un point de coupe et la mutation est faite avec le changement de bits, selon une probabilité p_m . La technique de élitisme consiste en ajouter les populations des génération t et $t + 1$, et choisir les premières n_{pop} individus des premières frontières.

Les méthodes de niche ont la finalité d'obtenir et maintenir un niveau de diversité dans des populations d'algorithmes génétiques à travers de l'encourageant la formation de sous-populations de solutions stables. Ainsi, les niches permettent aux algorithmes génétiques de rechercher des régions en parallèle et donc d'éviter la convergence anticipée de l'algorithme à une région optimale local. Plusieurs méthodes de niche ont été reportées dans

la littérature. Dans ce travail, il est introduit la technique de niche NB[I] (Niches par Bisection d'Intervalles). NB[I] sépare les niches naturellement sans avoir besoin d'exécuter calculs de distance entre les solutions, qui est un inconvénient normalement cité dans les travaux scientifiques. NB[I] commence par encadrer toutes les solutions, noté pour la performance individuelle $\overline{[f]}_{\mathbf{x}^i}^*$, dans une boîte $[\mathbf{F}]_0$ sur l'espace des objectifs. Ainsi, pour chaque individu, il est toujours vérité l'inclusion suivante :

$$\overline{[f]}_{\mathbf{x}^i}^* \subseteq [\mathbf{F}]_0, \quad [\mathbf{F}]_0 = [\underline{F}_1, \overline{F}_1] \times \dots \times [\underline{F}_j, \overline{F}_j] \times \dots \times [\underline{F}_{n_f}, \overline{F}_{n_f}], \quad (3.38)$$

soient :

$$\underline{F}_j = \min_{i=1, \dots, n_{pop}} \overline{[f_j]}_{\mathbf{x}^i}^* \quad \text{et} \quad \overline{F}_j = \max_{i=1, \dots, n_{pop}} \overline{[f_j]}_{\mathbf{x}^i}^*. \quad (3.39)$$

Le boîte $[\mathbf{F}]_0$ est le niche initiale. Dans la prochaine iteration, $[\mathbf{F}]_0$ est bisecté en produisant les deux boîtes :

$$[\mathbf{F}]_1 = [F_1] \times \dots \times [\underline{F}_j, \frac{F_j + \overline{F}_j}{2}] \times \dots \times [F_{n_f}], \quad (3.40)$$

$$[\mathbf{F}]_2 = [F_1] \times \dots \times [\frac{F_j + \overline{F}_j}{2}, \overline{F}_j] \times \dots \times [F_{n_f}]. \quad (3.41)$$

Ainsi, la classification de \mathbf{x}^i est réglé par :

$$\begin{cases} \overline{[f]}_{\mathbf{x}^i}^* \in [\mathbf{F}]_1, & \text{si } \overline{[f_j]}_{\mathbf{x}^i}^* \leq \frac{F_j + \overline{F}_j}{2}, \\ \overline{[f]}_{\mathbf{x}^i}^* \in [\mathbf{F}]_2, & \text{autrement.} \end{cases} \quad (3.42)$$

Après la classification de toutes les solutions, s'il y a une niche vide, alors il est éliminé. Le procédé de bisection continue jusqu'à un *numéro maximale de cycles de bisection* r_{bis}^{max} , soient un cycle de bisection la exécution d'un seule bisection en chaque boîte qui étaient disponibles dès le début da l'itération. Sur le texte principal du mémoire, le Fig.3.6 et le Fig.3.7 présentent quelques itérations réalisées sur un exemple générique. Et aussi, la Tab.3.9 détaille le processus de la méthode NB[I].

Après cette phase de définitions, il est présenté l'algorithme [I]RMOEA dans la Tab.3.11. Les paramètres utilisés par [I]RMOEA sont listés dans la Tab.3.12.

Références: Deb [32], et Deb et al. [34].

Table 3.11: Algorithme [I]RMOEA.

entrée($n_{bits}, n_f, n_{pop}, n_{gen}, p_c, p_m, [\mathbf{X}], [\mathbf{P}], \varepsilon_{[p]}$) sortie($\mathbf{pop}_t, \mathbf{des}_t$)	
1	Initialisez t avec 0.
2	Traitez contraintes(entrée: $[\mathbf{X}], [\mathbf{P}], \varepsilon_{[x]}, \varepsilon_{[p]}$; sortie: $\mathbf{Q}_{[x]}, \mathbf{Q}_{[p]}$)-Tab.3.6.
3	Initialisez \mathbf{pop}_t avec n_{bits}, n_f, n_{pop} et $\mathbf{Q}_{[x]}$.
5	Calculez $\mathbf{des}_t = \overline{\mathbf{f}}_{\mathbf{x}}^*$, $\mathbf{x} \in \mathbf{pop}_{aux}$ - Eq. (3.28) et Eq. (3.29).
6	Classifiez les frontières et calculez f_{front}^i .
7	Si $t \geq n_{gen}$ allez au <i>Pas 15</i> .
8	Ajoutez 1 à t .
9	Calculez f_{niche}^i - NBI.
10	Calculez \mathbf{des}'_t - Eq.(3.37).
10	Réalisez <i>sélection</i> - probabilité $\mathbf{des}'_t(i) / \sum_{j=1}^{n_{pop}} \mathbf{des}'_t(j)$.
11	Réalisez <i>croisement</i> et <i>mutation</i> selon les probabilités p_c et p_m . Composez la population \mathbf{pop}_{aux} .
12	Remplacez $\forall \mathbf{x} \in \mathbf{pop}_{aux}$ tel que $[\mathbf{g}](\mathbf{x}, \mathbf{Q}_{[p]}) \subset [0, \infty]$ par autre généré par événement aléatoire en $\mathbf{Q}_{[x]}$.
13	Calculez $\mathbf{des}_{aux} = \overline{\mathbf{f}}_{\mathbf{x}}^*$, $\mathbf{x} \in \mathbf{pop}_{aux}$ - Eq. (3.28) et Eq. (3.29).
14	Réalisez élitisme en considérant $\mathbf{des}_{t-1} \cup \mathbf{des}_{aux}$, composez la population \mathbf{pop}_t et allez au <i>Pas 7</i> .
15	Retournez \mathbf{pop}_t et \mathbf{des}_t .

Table 3.12: Paramètres du [I]RMOEA.

Symboles	Description	Type
n_{bits}	numéro de bits de la codification	\mathbb{Z}
n_{gen}	numéro de générations	\mathbb{Z}
n_{pop}	numéro d'individus	\mathbb{Z}
n_f	numéro de fonctions objectif	\mathbb{Z}
f_{front}^i	performance d'individu i selon sa frontière	\mathbb{Z}
f_{niche}^i	performance d'individu i selon sa niche	\mathbb{Z}
\mathbf{pop}_t	population en $n_{gen} = t$	Matrice: \mathbb{R}^{n_f}
\mathbf{des}_t	performance de \mathbf{pop}_t	Matrice: \mathbb{R}^{n_f}
\mathbf{pop}_{aux}	population auxiliaire	Matrice: \mathbb{R}^{n_f}
\mathbf{des}_{aux}	performance de \mathbf{pop}_{aux}	Matrice: \mathbb{R}^{n_f}

3.7 - Métriques de Mesure des Performances Révisées

Dans cette section il est présenté une technique, appelée MB[I] (Métrique

par Bissection d'Intervalles), pour mesurer l'homogénéité des images des solutions sur la frontière de Pareto. La structure algorithmique est similaire à celle décrite pour la technique NB[I]. La métrique MB[I] commence par encadrer toutes les solutions, selon ses valeurs de performance \mathbf{des}_t :

$$\mathbf{des}_t = \{\mathbf{f}_{\mathbf{x}^1}, \dots, \mathbf{f}_{\mathbf{x}^i}, \dots, \mathbf{f}_{\mathbf{x}^{n_{pop}}}\}, \quad i = \{1, \dots, n_{pop}\} \quad (3.43)$$

soient $\mathbf{f}_{\mathbf{x}^i} \in \mathbb{R}^{n_f}$ et l'accès à j -ième fonction objectif d'individu \mathbf{x}^i est fait par $f_{j_{\mathbf{x}^i}}$. Ainsi, pour chaque individu, il est toujours vérité l'inclusion suivante :

$$\mathbf{f}_{\mathbf{x}^i} \subseteq [\mathbf{F}]'_0, \quad [\mathbf{F}]'_0 = [\underline{F}'_1, \overline{F}'_1] \times \dots \times [\underline{F}'_j, \overline{F}'_j] \times \dots \times [\underline{F}'_{n_f}, \overline{F}'_{n_f}], \quad (3.44)$$

soient :

$$\underline{F}'_j = \min_{i=1, \dots, n_{pop}} f_{j_{\mathbf{x}^i}} \quad \text{e} \quad \overline{F}'_j = \max_{i=1, \dots, n_{pop}} f_{j_{\mathbf{x}^i}}. \quad (3.45)$$

Un exemple de la construction de la boîte $[\mathbf{F}]'_0$ est montré dans la Fig.3.8, itération 0. Le prochain pas est bissecter $[\mathbf{F}]'_0$. La coupe de bissection est fait sur un individu k interne à $[\mathbf{F}]'_0$ et une fonction objectif j choisit événement aléatoire, en résultant :

$$[\mathbf{F}]'_1 = [F_1] \times \dots \times [\underline{F}_j, f_{j_{\mathbf{x}^k}}] \times \dots \times [F_{n_f}], \quad (3.46)$$

$$[\mathbf{F}]'_2 = [F_1] \times \dots \times [f_{j_{\mathbf{x}^k}}, \overline{F}_j] \times \dots \times [F_{n_f}]. \quad (3.47)$$

Pour mieux comprendre la première bissection, observez la Fig.3.8, iteration 1. Les individus i dans la boîte $[\mathbf{F}]'_0$ sont séparés en considérant l'objectif j , selon la règle :

$$\begin{cases} \mathbf{f}_{\mathbf{x}^i} \in [\mathbf{F}]_1, & \text{si } f_{j_{\mathbf{x}^i}} \leq f_{j_{\mathbf{x}^k}}, \\ \mathbf{f}_{\mathbf{x}^i} \in [\mathbf{F}]_2, & \text{autrement.} \end{cases} \quad (3.48)$$

À la suite, les bisections continuent jusqu'au pas où toutes les boîtes qui lient les points sont vides, conformément montré dans la Fig.3.8, itération 11. À la fin du procédé de bissection, toutes les boîtes sont mémorisées dans la file $\mathbf{Q}_{[\mathbf{F}]}$, et en calculent le volume $v([\mathbf{F}])$ de chacune, la métrique MB[I] est définie comment l'écart type $\sigma_{v([\mathbf{F}])}$, ou formellement,

$$\text{MB}[\text{I}] = \sigma_{v([\mathbf{F}])}. \quad (3.49)$$

Autant $\sigma_{v(\mathbf{F})}$ est petit, autant mieux sera l'homogénéité de la frontière de Pareto. Alternativement, il peut être utilisé une valeur normalisée $MB[I]_{norm}$:

$$MB[I]_{norm} = \frac{MB[I] - MB[I]_{front}}{MB[I]_{max} - MB[I]_{front}}, \quad MB[I]_{max} \neq MB[I]_{front}, \quad (3.50)$$

soient $MB[I]_{front}$ la valeur de $MB[I]$ calculé d'un ensemble solution de Pareto connu, et $MB[I]_{max}$ la valeur maximale trouvée pour la métrique après l'évaluation d'autres ensembles solution.

Cette section a réalisé aussi un comparatif entre les autres métriques présentés sur le chapitre dernier. Les résultats situent $MB[I]$ comme une méthode efficace pour évaluer la qualité de la distribution de point sur la frontière de Pareto.

3.8 - Considérations

Cette section a présentée le résumé du chapitre.

Chapitre 4 - Validation Numérique des Algorithmes pour l'Optimisation Robuste

Dans ce chapitre, ils sont présentés des tests de validation pour les algorithmes [I]RMOA I, [I]RMOA II et [I]RMOEA. La validation a été réalisée par moyen de fonctions test. Les expressions mathématiques des fonctions tests ont été formulées selon la philosophie du scénario du pire cas. C'est important à observer que les méthodes proposées dans ce travail sont lentes et un peu précises si comparées avec des méthodes spécifiques pour un problème particulier. Par exemple, si le problème robuste en question est continu et convexe, alors il y a des *méthodes de points intérieurs*, basés en dérivatives, que résoudre ce genre de problème rapidement, précisément et en considérant plusieurs centaines de variables. Par contre, les algorithmes proposés ont été développés pour traiter des problèmes d'optimisation multi-objectif génériques, c'est à dire, qu'ils soient: a) de recherche globale ; b) garantis pour envelopper la frontière robuste ([I]RMOA I et [I]RMOA II) ; b) indépendants de dérivatives ; c) indépendants de probabilités pour calculer les incertitudes ; et d) indépendants de convexité de la Frontière de Pareto.

Références: Kouvelis et Yu [75], et Boyd et Vandenberghe [13]

4.1 - Méthodologie

En partant de l'idée de valider les méthodes d'optimisation proposées, ils ont été formulés cinq problèmes tests qui présentent quelques caractéristiques différentes comme la non convexité et non continuité de la frontière de Pareto robuste. Les problèmes choisis sont versions robustes adaptées de problèmes tests multi-objectif traditionnelles, et ils prennent en compte deux objectifs pour faciliter l'interprétation graphique des résultats. Donc, la validation consiste des analyses de tests expérimentaux.

4.2 - Fonctions Test

La définition du problème robuste est dépendent d'interprétation d'action des incertitudes sur le système d'optimisation, comment a été discuté antérieurement. Dans ce travail, l'incertitude a été considérée un paramètre borné et différent à la variable de projet. Les fonctions test robustes utilisées sont présentées à la suite. Dans le texte principal du rapport se trouvent aussi les expressions originales de chaque problème et un graphique confrontant les images résultantes des deux cas.

La fonction SCH2 (Fonction Schaffer 2 - Eq. 4.2, Fig.4.1) se caractérise par être sous contraintes, utiliser une seule variable, par avoir la frontière de Pareto non continue et par avoir simples occurrences de x et \mathbf{p} . Ce dernier aspect est important pour la qualité des résultats des fonctions d'inclusion. Quand il y a plusieurs occurrences, la fonction d'inclusion produit une image plus pessimiste, qui est indésirable. Dans ce cas, l'incertitude peut être classifié comment quelque type d'imprécision de x . L'ensemble solution robuste est $\mathbf{X}^* \in \{[1, 2] \cup [4, 5]\}$. Le pire cas se trouve en $p_1 = 0.1$ par f_1 et en $\mathbf{p} = \{-0.1, 0.1\}$ por f_2 . Comme conséquence $\mathbf{Y}^* \not\subset \mathbf{f}(x, \mathbf{p})$.

$$\min_{x \in [-5, 10]} \max_{\mathbf{p} \in [-0.1, 0.1]} f_1(x, \mathbf{p}) = \begin{cases} -(x + p_1), & \text{se } x \leq 1 \\ (x + p_1) - 2, & \text{se } 1 < x \leq 3 \\ 4 - (x + p_1), & \text{se } 3 < x \leq 4 \\ (x + p_1) - 4, & \text{se } x > 4 ; \end{cases} \quad (4.2)$$

$$f_2(x, \mathbf{p}) = ((x + p_1) - 5)^2 + p_2.$$

La fonction FON (Fonseca et Fleming - Eq. 4.4, Fig.4.2) se caractérise par être sous contraintes, par utiliser n variables et deux objectifs, par avoir la frontière de Pareto continue, non convexe. Aussi dans ce cas, l'incertitude peut être interprété comment une sorte d'imprécision de paramètre. Le pire cas se trouve en $p = 1.3$, quand $x_i = 1/\sqrt{n}$ et $x_i = -1/\sqrt{n}$, par f_1 et f_2 , respectivement.

$$\begin{aligned} \min_{\mathbf{x} \in [-4,4]} \max_{p \in [1.1, 1.3]} f_1(\mathbf{x}, p) &= 1 - e^{-\sum_{i=1}^n (x_i - \frac{p}{\sqrt{n}})^2} ; \\ f_2(\mathbf{x}, p) &= 1 - e^{-\sum_{i=1}^n (x_i + \frac{p}{\sqrt{n}})^2}. \end{aligned} \quad (4.4)$$

La fonction ZDT1 (Fonction Zitzler, Deb et Thiele 1 - Eq. 4.7, Fig.4.3) se caractérise par être sous contraintes, par utiliser n variables, par avoir la

$$\begin{aligned} \min_{\mathbf{x} \in [0,1]} \max_{\mathbf{p} \in [0,0.05]} f_1(\mathbf{x}, \mathbf{p}) &= x_1 + p_1 ; \\ f_2(\mathbf{x}, \mathbf{p}) &= h'(\mathbf{x}) \times h(f_1, h', \mathbf{p}). \end{aligned} \quad (4.7)$$

$$\text{soient :} \quad h'(\mathbf{x}) = 1 + \frac{9}{n-1} \sum_{i=2}^n x_i,$$

$$h(f_1, h', \mathbf{p}) = 1 - \sqrt{f_1/h' + p_2}.$$

frontière de Pareto continue et convexe, et par avoir multiples occurrences de x et \mathbf{p} . Dans ce cas, le paramètre d'incertitude représente un type d'imprécision de x_1 en f_1 , et l'idée que f_2 est une approximation de sa valeur correcte. Le domaine de la région robuste est $x_1 \in [0, 1]$ et $x_i = 0$, $i = 2, \dots, n$.

La fonction ZDT2 (Fonction Zitzler, Deb et Thiele 2 - Eq. 4.9, Fig.4.4) se caractérise par être contraintes, par utiliser n variables, par avoir la frontière de Pareto non continue et non convexe, et par avoir multiples occurrences de x et \mathbf{p} . Aussi dans ce cas, la fonction f_2 est une approximation de sa valeur correcte. Le domaine de la région robuste est $x_1 \in [0, 1]$ et $x_i = 0$, $i = 2, \dots, n$.

$$\begin{aligned} \min_{\mathbf{x} \in [0,1]} \max_{\mathbf{p} \in [-0.05, 0.05]} f_1(\mathbf{x}, p) &= x_1 ; \\ f_2(\mathbf{x}, p) &= h'(\mathbf{x}, p) \times h(f_1, h', p). \\ \text{s.c.} \quad g(\mathbf{x}) &= \sqrt{(x_1 - 0.7)^2 + x_2^2 + \dots + x_n^2} > 0.1 \quad (4.9) \end{aligned}$$

$$\begin{aligned} \text{soient :} \quad h'(\mathbf{x}, p) &= 1 + \frac{9}{n-1} (\sum_{i=2}^n x_i) + p, \\ h(f_1, h', p) &= 1 - (f_1/h')^2 + p. \end{aligned}$$

La fonction ZDT3 (Fonction Zitzler, Deb et Thiele 3 - Eq. 4.11, Fig.4.5) se caractérise par être sous contraintes, par utiliser n variables, par avoir la frontière de Pareto non continue, par avoir l'espace des objectifs multimodal, et par avoir multiples occurrences de x et \mathbf{p} . Dans ce cas, le paramètre d'incertitude représente l'idée que la fonction f_2 est une approximation de sa

valeur correcte. Le domaine de la région robuste est $x_1 \in [0, 1]$ et $x_i = 0$, $i = 2, \dots, n$.

$$\begin{aligned} \min_{\mathbf{x} \in [0,1]} \max_{\mathbf{p} \in [-0.1,0.1]} \quad & f_1(\mathbf{x}, \mathbf{p}) = x_1 ; \\ & f_2(\mathbf{x}) = h'(\mathbf{x}, p) \times h(f_1, h', p). \end{aligned} \quad (4.11)$$

$$\begin{aligned} \text{soient :} \quad & h'(\mathbf{x}, p) = 1 + \frac{9}{n-1} ((\sum_{i=2}^n x_i) + p), \\ & h(f_1, h', p) = 1 - \sqrt{f_1/h'} - (f_1/h') \text{sen}(10\pi f_1). \end{aligned}$$

Références: Barichard et Hao [6], Coello et Sierra [21], Deb et al. [34], Deb [32], Schaffer [101], Fonseca et Fleming [43], Deb [31], et Zitzler et al. [130]

4.3 - Tests et Résultats

Dans les tests, il a été observé comment les algorithmes se comportent quand ses paramètres changent de valeurs. Les critères d'évaluation ont été basés en approximations de l'ensemble solution robuste \mathbf{X}^* et son image \mathbf{Y}^* . Donc, pour aider ces comparaisons, ils ont été définis un ensemble solution discrète $\mathbf{X}_d^* \subset \mathbf{X}^*$ et sa image $\mathbf{Y}_d^* \subset \mathbf{Y}^*$ comme les ensembles corrects à être obtenus pour chaque fonction test (\mathbf{X}_d^* a été trouvé en testant toutes les solutions d'un domaine discret), et les ensembles \mathbf{X}_s^* et \mathbf{Y}_s^* représentent les résultats obtenus des algorithmes proposés.

Les algorithmes sont différents, comme conséquence les paramètres à être testés sont aussi un peu divers. Ainsi, les résultats sont présentés en considérant les exécutions réalisées pour chaque algorithme de façon séparée. Avant les résultats, un sommaire sur la Tab.4.1 présent quelques valeurs fixées des problèmes tests.

Table 4.1: Sommaire du problème tests pour l'optimisation robuste.

Fonction	$\mathbf{f}(\mathbf{x}, \mathbf{p})$, $\mathbf{g}(\mathbf{x}, \mathbf{p})$	$[\mathbf{X}] \in \mathbb{IR}^{n_x}$	$[\mathbf{P}] \in \mathbb{IR}^{n_p}$	n_x	n_p
SCH2	(4.2)	$[-5, 10]$	$[-\mathbf{0.1}, \mathbf{0.1}]$	1	2
FON	(4.4)	$[-\mathbf{4}, \mathbf{4}]$	$[1.1, 1.3]$	2	1
ZDT1	(4.5) et (4.7)	$[\mathbf{0}, \mathbf{1}]$	$[\mathbf{0}, \mathbf{0.05}]$	2	2
ZDT2	(4.5) et (4.9)	$[\mathbf{0}, \mathbf{1}]$	$[-0.05, 0.05]$	2	1
ZDT3	(4.5) et (4.11)	$[\mathbf{0}, \mathbf{1}]$	$[-0.1, 0.1]$	2	1

4.3.1 - [I]RMOA I

La performance et l'efficace du [I]RMOA I dépendent de ses paramètres de précision $\varepsilon_{\mathbf{x}\#}$ et $\varepsilon_{[\mathbf{x}]}$. Donc, les tests consistent en observer comment l'algorithme se comporte quand $\varepsilon_{\mathbf{x}\#}$ et $\varepsilon_{[\mathbf{x}]}$ changent de valeur. Pour cet algorithme trois résultats de tests ont été présentés : a) la quantité d'éléments de \mathbf{K}^- et \mathbf{K}^+ , que définissent l'enveloppe sur la frontière robuste ; b) l'effort computationnel par évaluer les échantillons obtenus de la fonction *grid* et pour compléter toute la simulation (ces efforts ont été mesurés en numéro de calculs de fonctions objectif) ; et c) la qualité des enveloppes sur \mathbf{Y}^* .

Pour réaliser les tests, ils ont été utilisés plusieurs valeurs de $\varepsilon_{\mathbf{x}\#}$ et $\varepsilon_{[\mathbf{x}]}$. En considérant que la notation $\varepsilon_{\mathbf{x}\#} = [0.02 : 0.02 : 0.1]$ pour représenter la structure de données $\varepsilon_{\mathbf{x}\#} = \{0.02, 0.04, \dots, 0.1\}$, la table ci-dessous décrite de valeur de paramètres de précision testés.

Fonction	$\varepsilon_{\mathbf{x}\#}$	$\varepsilon_{[\mathbf{x}]}$
SCH2	[0.02 : 0.02 : 0.1]	[0.02 : 0.02 : 0.1]
FON	[0.05, 0.1, 0.2 : 0.2 : 1]	[0.2 : 0.2 : 1]
ZDT1	[0.02 : 0.01 : 0.1]	[0.02 : 0.01 : 0.1]
ZDT2	[0.02 : 0.02 : 0.1]	[0.01, 0.02 : 0.02 : 0.1]
ZDT3	[0.02 : 0.02 : 0.1]	[0.02 : 0.01 : 0.1]

L'enveloppe sur la frontière robuste est composée pour les éléments des ensembles discrets \mathbf{K}^- et \mathbf{K}^+ . Ainsi, la quantité de ces éléments reflètent en un augmente de la précision des design d'enveloppe. Les figures Fig.4.6, Fig.4.9, Fig.4.12, Fig.4.15 et Fig.4.18 confrontent \mathbf{K}^- et \mathbf{K}^+ , en considérant 3 valeurs de $\varepsilon_{[\mathbf{x}]}$. En analysant les figures, il est constaté que l'échantillonnage, défini par $\varepsilon_{\mathbf{x}\#}$, est le facteur fondamental pour grandir les ensembles \mathbf{K}^- et \mathbf{K}^+ .

Le [I]RMOA I se présente comme un algorithme "coûteux" en considérant le numéro de calculs de fonctions objectif dépensées pour résoudre tous les cas discutés. La question du coût computationnel est montré dans les figures Fig.4.7, Fig.4.10, Fig.4.13, Fig.4.16 et Fig.4.19. L'effort principal se concentre sur les deux fonctions FPS (Tab.3.3) et CSC (Tab.3.4) quand elles essayent, en vain, trouver un minimiseur pour une candidat à \mathbf{K}^- .

Quelques résultats des enveloppes obtenus ont été exhibés dans les figures Fig.4.8, Fig.4.11, Fig.4.14, Fig.4.17 et Fig.4.20. Parfois, les figures ont été présentées des simulations qui ne réussissaient pas envelopper toute la frontière robuste comment en Fig.4.11(b) Fig.4.20(a). Les raisons pour lesquelles ça est arrivé sont dues aux fonctions d'inclusions utilisées, que ne sont pas

finies, car de la multiplicité d'occurrences des paramètres dans la formulation, par exemple. Pour équilibrer l'effet de la multiplicité, le paramètre $\varepsilon_{[x]}$ doit être suffisant plus petit que $\varepsilon_{x\#}$, au moins $\varepsilon_{[x]} < \varepsilon_{x\#}/2$, pour assurer que la précision de recherche soit mieux que la précision de l'échantillonnage.

En considérant les résultats, le [I]RMOA I a réussi envelopper les frontières robustes dès que les facteurs de précision sont réglés et que le vecteur de fonctions objectif d'inclusion soit fin. Le processus de recherche dans le [I]RMOA I est simple et l'algorithme utilise seulement deux paramètres. De l'autre côté, des inconvénients sont liés au grand effort dépensé pour finir la recherche par solutions, et au fait qu'il trouve une enveloppe sur \mathbf{Y}^* et non sur le \mathbf{X}^* . En fait, il y a des méthodes qui font l'inversion d'ensembles et trouvent \mathbf{X}^* . Mais, l'enveloppe sur \mathbf{Y}^* est aussi un avantage, car le [I]RMOA I peut devenir une méthode de validation à autres méthodes. Ce sujet est traité dans la Section 4.4.

4.3.2 - [I]RMOA II

La performance et l'efficace du [I]RMOA II dépendent de ses paramètres de précision $\varepsilon_{[x]}$ et $\varepsilon_{[p]}$. Pour chaque fonction test, $\varepsilon_{[p]}$ a été fixé, et ainsi, les tests consistent en observer comment l'algorithme se comporte quand $\varepsilon_{[x]}$ change de valeur. Pour cet algorithme trois résultats de tests ont été présentés: a) le numéro de solutions trouvées ; b) l'effort computationnel ; c) la comparaison entre \mathbf{X}_s^* et \mathbf{X}_d^* ; et d) la comparaison entre \mathbf{Y}_s^* et \mathbf{Y}_d^* . Les valeurs utilisées pour $\varepsilon_{[x]}$ et $\varepsilon_{[p]}$ sont montrés dans la table à la suite.

Fonction	$\varepsilon_{[x]}$	$\varepsilon_{[p]}$
SCH2	[0.1 : 0.1 : 1]	0.2
FON	[0.1 : 0.1 : 1]	0.025
ZDT1	[0.01 : 0.01 : 0.1]	0.02
ZDT2	[0.01 : 0.01 : 0.1]	0.02
ZDT3	[0.01 : 0.01 : 0.1]	0.02

Selon la Tab.3.7, les produits de [I]RMOA II sont une file $\mathbf{Q}_{[x]}$ qui contient toutes les boîtes qu'enveloppent les solutions robustes, et deux autres files $\mathbf{Q}_{\mathbf{X}^*}$ et $\mathbf{Q}_{\mathbf{Y}^*}$, composées pour les points centraux de chaque élément de $\mathbf{Q}_{[x]}$ et de ses images robustes, respectivement. Les figures Fig.4.21, Fig.4.24, Fig.4.27, Fig.4.30 et Fig.4.33 présentent comment le numéro de solutions augmente avec la précision de $\varepsilon_{[x]}$. En fait, à chaque itération ils sont réalisés des bisections sur toutes les boîtes enregistrées en $\mathbf{Q}_{[x]}$, et après les bisections, l'algorithme [I]RMOA II met à jour $\mathbf{Q}_{[x]}$, $\mathbf{Q}_{\mathbf{X}^*}$ et $\mathbf{Q}_{\mathbf{Y}^*}$. Par conséquent,

à chaque itération la précision augmente et aussi la quantité de solutions. Aussi, les figures citées dernièrement ont montré l'effort computationnel par valeur de précision. Le [I]RMOA II se présente aussi comme un algorithme coûteux, car dans toutes les itérations des évaluations sont doubles : sur les boîtes de $\mathbf{Q}_{[x]}$ et sur ses centres $\mathbf{Q}_{\mathbf{X}^*}$.

Les figures Fig.4.22, Fig.4.25, Fig.4.28, Fig.4.31 et Fig.4.34 exhibent le contraste entre les solutions discrètes \mathbf{X}_d^* et les solutions trouvées \mathbf{X}_s^* , en considérant deux valeurs de $\varepsilon_{[x]}$. Dans les cas, observez que tous les éléments de \mathbf{X}_s^* ont atteint la précision déterminée par $\varepsilon_{[x]}$, et avec une évidente uniformité de la distribution. Cependant, [I]RMOA II est dit garanti. Pour obtenir plus de solutions un nouveau cycle de bisections est requis.

Les figures Fig.4.23, Fig.4.26, Fig.4.29, Fig.4.32 et Fig.4.35 comparent les frontières robustes \mathbf{Y}_d^* et \mathbf{Y}_s^* , en considérant deux valeurs de $\varepsilon_{[x]}$. Dans les cas, observez que la frontière trouvée \mathbf{Y}_s^* est uniforme et s'approche de \mathbf{Y}_d^* quand la valeur de $\varepsilon_{[x]}$ se devient plus petite.

En considérant les résultats, il peut être vérifié que le [I]RMOA II est une méthode multi-objectif globale, car il converge à toutes les régions optimales. La proximité de \mathbf{X}_s^* à l'ensemble solution du problème est dépendante du paramètre de précision $\varepsilon_{[x]}$. Les tests ont montré que le [I]RMOA II marche indépendamment de caractéristiques, qui normalement sont considérées source de difficultés, comme fonctions objectif non continues, non convexes et contraintes. Il y a aussi des inconvénients. Le nombre de variables et d'incertitudes peut laisser le procédé de bisection irréalisable, au moins qu'une stratégie optimiste soit utilisée (voir Fig.2.8). Le [I]RMOA II, comme toutes les méthodes d'intervalles, est dépendant de la qualité de la fonction d'inclusion. Donc, il est tant et plus préférable que les fonctions objectif d'inclusion soient fines et minimales, et fondamentales qu'elles soient convergentes et monotones.

4.3.3 - [I]RMOEA

La performance et l'efficacité du [I]RMOEA dépendent de plusieurs paramètres comme aux autres algorithmes évolutionnaires en général. Pour simplifier les expériences, quelques paramètres ont été fixés, sont-ils : a) probabilité de croisement $p_c = 1$; b) probabilité de mutation $p_m = 0.001$; et c) nombre maximal de cycles de bisection $r_{bis}^{max} = 8$. Les paramètres $\varepsilon_{[p]}$ et ε_{TE} ont été définis pour chaque problème. Les tests ont évalué comment le [I]RMOEA répond quand la taille de la population n_{pop} et le nombre de génération n_{gen} changent de valeur. Pour cet algorithme cinq résultats de tests ont été présentés : a) la convergence de l'algorithme, mesurée en nombre de frontières ; b) la comparaison entre \mathbf{X}_s^* et \mathbf{X}_d^* , via métrique TE ; c) l'homogénéité de l'ensemble Pareto, via métrique MB[I] ; d) la comparaison entre \mathbf{Y}_s^* et \mathbf{Y}_d^* , via métrique TE ; et e) l'effort computationnel, mesuré

en numéro de calculs de fonctions. Les valeurs utilisées pour n_{pop} , n_{gen} , $\varepsilon_{[p]}$ et ε_{TE} sont listés dans la table présentée suivant.

Fonction	n_{pop}	n_{gen}	$\varepsilon_{[p]}$	ε_{TE}
SCH2	[10 : 10 : 100]	[10 : 10 : 50]	0.2	0.10
FON	[10 : 10 : 100]	[10 : 10 : 50]	0.02	0.05
ZDT1	[10 : 10 : 100]	[10 : 10 : 100]	0.01	0.01
ZDT2	[10 : 10 : 100]	[10 : 10 : 100]	0.01	0.01
ZDT3	[10 : 10 : 100]	[10 : 10 : 100]	0.01	0.01

Dans les algorithmes évolutionnaires, l'effort est normalement constant par itération principalement parce que la taille de la population de solutions ne change pas. C'est le cas du [I]RMOEA. La Fig.4.36 résume les valeurs d'effort par génération en considérant n_{pop} sur le abscisse, pour tous les problèmes tests. Le coût computationnel du [I]RMOEA est le plus petit d'entre aux autres algorithmes proposés.

L'évolution du numéro de frontières en fonction des générations est un indicateur la convergence du [I]RMOEA au l'ensemble \mathbf{Y}_s^* . Les figures Fig.4.37, Fig.4.41, Fig.4.45, Fig.4.49 et Fig.4.53 ont montré cette convergence. En fait, la convergence à la première frontière est importante, mais il est aussi important d'avoir diversité et uniformité dans l'ensemble optimal.

L'application de la métrique TE dans cette phase de tests a une double finalité: a) mesurer combien des solutions dans l'ensemble \mathbf{X}_s^* appartient \mathbf{X}_d^* ; et b) mesurer combien de solutions sont dupliquées. Le but en (b) a été qualifier la diversité dans les ensembles produits. En utilisant la métrique TE, les figures Fig.4.38, Fig.4.42, Fig.4.46, Fig.4.50 et Fig.4.54 présentent que les mieux résultats ont été obtenus en pour les populations plus grandes. Mais, le numéro de duplicates augmente aussi avec la taille de la population. Exécutions avec numéro de générations plus grandes ont obtenu les mieux résultats.

La métrique MB[I] a été développée pour mesurer l'uniformité des ensembles Pareto. La MB[I] est aussi un indicateur indirect de la qualité de la technique de niche NB[I], qui est le mécanisme responsable pour maintenir la diversité avant du procédé de sélection. Les figures Fig.4.39, Fig.4.43, Fig.4.47, Fig.4.51 et Fig.4.55 ont montré que les populations plus grandes ont été plus succès en obtenir ensembles plus uniformes.

Les résultats dans l'espace des objectifs ont été montrés sur les figures Fig.4.40, Fig.4.44, Fig.4.48, Fig.4.52 et Fig.4.56. L'uniformité de distribution des ensembles \mathbf{Y}_s^* sur la frontière robuste a été clairement mieux quand ils ont a été utilisées des populations de tailles plus grandes. Mais, il faut justifier le

pourquoi. Dans les expériences, ils ont été fixés le numéro maximal de cycles de bisection ($r_{bis}^{max} = 8$) pour exécuter la technique NB[I]. Si le r_{bis}^{max} définit le volume des niches et s'il a plusieurs niches, alors populations de tailles plus petites n'ont pas beaucoup nombre de représentants pour appartenir à toutes les niches. Les niches vides sont éliminées dans les procédés du NB[I] amenant l'apparition de "trous" sur la frontière robuste.

Selon les résultats présentés pour le [I]RMOEA, il a été considéré que le méthode a obtenu succès en trouver les frontières robustes en chaque cas, même quand [I]RMOEA a confronté les frontières robustes non continues et non convexes. Le traitement de contraintes par intervalles a été considéré efficient, car la classification a priori éliminée toutes les régions irréalisables. Le coût d'exécuter l'algorithme a été bas quand se compare avec les deux autres méthodes proposées. La convergence à la première frontière est arrivée avant la vingtième génération en tout les cas étudiés. Ainsi, le critère pour arrêter le [I]RMOEA avait pu considéré ce renseignement et finir l'exécution avant le numéro de génération fixée à priori.

Références: Dao [27], et Soares [107]

4.4 - Tests Complémentaires

Cette section a présenté des expériences spécifiques pour chaque méthode proposée. Pour le [I]RMOA I, il a été démontré comme l'algorithme peut être utilisé pour valider d'autres méthodes développées pour l'optimisation multi-objectif robuste. Pour les [I]RMOA II et [I]RMOEA, les expériences réalisés ont testés la dépendance des algorithmes en relation à dimension du problème, c'est à dire lesquels sont les limitations pour résoudre les cas avec plusieurs variables.

4.4.1 - Le [I]RMOA I comment Outil de Validation

Le [I]RMOA I a été développé pour être une méthode de validation d'autres techniques construits pour traiter des problèmes robustes, formulés utilisant la philosophie du scénario du pire cas. Pour être apte à validation, le première pas est trouver, pour un problème connu, les paramètres de précision $\varepsilon_{[x]}$ et $\varepsilon_{x\#}$ que produisent une enveloppe correcte sur la frontière robuste. Dans cette phase, c'est le temps aussi de tester quelques fonctions objectif d'inclusion et choisir laquelle qui a le mineur pessimisme. Donc, assume qu'une frontière robuste \mathbf{Y}_1^* et que un ensemble solution \mathbf{X}_1^* sont connus

pour une valeur particulière d'incertitude \mathbf{p}_1 . Considère aussi qu'ils ont été construits des fonctions objectif d'inclusion de "bonne" qualité, et qu'ils ont été trouvés un pair $\varepsilon_{[x]}$ et $\varepsilon_{\mathbf{x}\#}$ qu'a résulté une analyse coût-bénéfice jugée satisfaisante. Encore, suppose que l'ambien d'incertitude a changé complètement, c'est à dire, la valeur \mathbf{p}_1 a changé pour \mathbf{p}_2 , et probablement, la frontière robuste \mathbf{Y}_1^* ne sera pas consistant avec ce nouveau scénario. Ainsi, denotez \mathbf{Y}_2^* et \mathbf{X}_2^* les ensembles liés à \mathbf{p}_2 . Dans l'espace de recherche, trois cas sont possibles : a) $\mathbf{X}_1^* = \mathbf{X}_2^*$; b) $\mathbf{X}_1^* \cap \mathbf{X}_2^* \neq \emptyset$, $\mathbf{X}_1^* \neq \mathbf{X}_2^*$; c) $\mathbf{X}_1^* \neq \mathbf{X}_2^*$. Si l'action des incertitudes n'est pas claire dans les expressions des fonctions objectif, alors il peut être difficile, à priori, savoir lequel sera le cas de \mathbf{X}_2^* . Ainsi, en prenant en compte le scénario à \mathbf{p}_2 , comment garantir qu'un ensemble \mathbf{Y}_2^* trouvé par une méthode n'importe quel est correct? Une possible réponse à cette question est l'application du [I]RMOA I pour envelopper \mathbf{Y}_2^* . Parce que, une fois qu'il a été vérifié pour résoudre le problème avec \mathbf{p}_1 , il est garanti crier une autre enveloppe en utilisant \mathbf{p}_2 (voir Théorème 2). À savoir, le [I]RMOA I utilisera les mêmes fonctions d'inclusion, $\varepsilon_{[x]}$ et $\varepsilon_{\mathbf{x}\#}$; et un nouveau \mathbf{p} .

Les expériences ont consisté en confronter le [I]RMOA I et un algorithme stochastique qu'a été adapté pour résoudre problèmes multi-objectif robustes. Le RMOGA (Algorithme Génétique Multi-Objectif Robuste) utilise la configuration suivante : a) la codification binaire ; b) le croisement avec un point de coupe ; c) la mutation par changement de bits ; d) la sélection par Roulette (performance individuelle calculée par Eq. 3.37) ; e) technique niche NB[I] (voir Sous-section 3.6.2) ; et f) l'élitisme a été défini comme le choix des premiers individus non dominés d'entre les deux dernières populations. Les fonctions tests choisies ont été ZDT3 et FON. Le Fig.4.57 a présenté les enveloppes sur les frontières robustes correctes pour les cas ZDT3 et FON. Pour le RMOGA, le problème multi-objectif robuste a été défini comment une approximation stochastique de la philosophie du pire, soient :

$$\min_{\mathbf{x} \in \mathbf{X}} \max_{\mathbf{p}' \in \mathbf{P}'} \mathbf{f}(\mathbf{x}, \mathbf{p}'), \quad (4.12)$$

où \mathbf{P}' est un ensemble discrète d'échantillons sélectionné par événement aléatoire de \mathbf{P} . En observant (4.12), notez que la qualité de la frontière Pareto gérée dépend de la taille de \mathbf{P}' . Alors, le [I]RMOA I a été utilisé pour aider le RMOGA en trouver une taille de population que représente la frontière robuste. Dans les expériences, pour raisons de simplicité, le paramètre \mathbf{p} a été fixé, donc les enveloppes présentés en Fig.4.57 sont encore valides. Les tailles des populations testés ont été $n_{pop} = \{5, 10, \dots, 50\}$, et les autres paramètres ont été $p_c = 1$, $p_m = 0.0005$, $n_{pop} = 50$ et $n_{gen} = 40$. Pour chaque

cas, le RMOGA a été exécuté 10 fois, soient les résultats jugé plus intéressantes exhibés en Fig.4.58 et Fig.4.59. Observez que dans les deux cas, la taille inadéquat a géré des frontières robustes faux, soient le $n_{pop} = 50$ la taille minimal to RMOGA produit frontière de qualité acceptable.

4.4.2 - [I]RMOA II × Dimension d'Espace de Recherche

Le but de cette sous-section a été présenter la limitation du [I]RMOA II en relation à dimension des paramètres d'un problème. Pour cet expérience, seulement la fonction ZDT3 a été utilisée. La précision pour la computation des incertitudes a été $\varepsilon_{[p]} = 0.01$ et l'espace de recherche analysé a été composé pour 3 et 4 variables. La précision $\varepsilon_{[x]} = 0.04$ a été employé pour limiter les bisections dans le processus de recherche. Le [I]RMOA II est une méthode garantie, alors les résultats obtenus ont été similaire auxquels présenté à la Sous-section 4.3.2, et donc ils ont été omis. Les simulations avec plusieurs variables pour les mêmes $\varepsilon_{[x]}$ et $\varepsilon_{[p]}$ cités plus haut ont été déclaré comment "irréalisables" pour l'équipement utilisé pour réaliser les tests (CPU Core Duo 1.73GHz/1.73GHz ; RAM 1024).

4.4.3 - [I]RMOEA × Dimension d'Espace de Recherche

Dans cette sous-section, la batterie des tests vise vérifier le comportement "moyen" du [I]RMOEA en relation à dimension des paramètres d'un problème. Pour réaliser les testes, seulement la version robuste da fonction ZDT3 a été utilisé. Les espaces de recherche adoptée pour les simulations ont été 5, 10, . . . , 30 variables, et la taille de la population a changé irrégulièrement de 15 à 300 individus, dépendant du cas. Le paramètre de précision d'incertitude a été fixé comme $\varepsilon_{[p]} = 0.01$. Ils ont été utilisés deux critères pour arrêter le [I]RMOEA: a) pour le numéro maximal de 200 générations ; et b) pour la convergence de la population dans la première frontière pendant 20 générations consécutives. En fait, les populations plus petites, $n_{pop} \leq 100$, ont la tendance de converger selon le critère d'arrête (b). Trois tests ont été exécutés en chaque cas. Les résultats jugés plus intéressants sont présentés à la suite.

Les Fig.4.60 et Fig.4.61 ont exemplifié les résultats de las frontières robustes pour les cas de 10 et 30 variables respectivement. Observez que la proximité de \mathbf{Y}_d^* dépend de la taille de la population. En analysant le cas plus critique (n=30 variables), les résultats ont été considérés satisfaisant, car le coût pour génération a resté constant et la frontière a été obtenu. La Tab.4.3 résume les résultats de cette section.

Table 4.3: [I]RMOEA - Résumé des tests complémentaires

n	n_{pop}	TE (%)	Dupliquées (%)	MB[I]
5	100	1.00	67.0	0.0040
10	100	3.00	52.0	0.0079
15	100	22.0	60.0	0.0244
20	200	1.50	62.5	0.0044
25	250	2.00	56.0	0.0040
30	300	3.33	47.7	0.0050

Références: Soares et al [110], Deb et al. [34], et Soares et al. [108]

4.5 - Considérations

Cette section a présenté le résumé du chapitre.

Chapitre 5 - Optimisation Robuste Multi-Objectif de Synthèse des Régulateurs PID

Normalement, le développement de fonctions d'inclusion part de l'expression analytique que décrit le problème. Dans le Chapitre 4, les fonctions test d'inclusion, classifiés comme fonctions d'inclusion naturelle, qui ont été obtenu par le remplacement direct des paramètres réels par son respectif intervalle. Cependant, la construction de fonctions d'inclusion est une tâche plus complexe quand les situations problèmes réelles sont examinées. Il y a aussi des particularités pour traiter problèmes par intervalles, comme contraintes de singularités dans opérations de division par exemple. Donc, cette chapitre a été dédicacé à traiter un problème d'optimisation multi-objectif robuste complète, de la conception à la résolution, qui a été composée par les étapes suivantes : a) une petite révision de l'application ; b) la description du problème général ; et c) la formulation, résolution et discussion d'une étude de cas.

En considérant des facteurs comme l'importance de l'application réelle dans l'ingénierie et l'industrie, l'intéresse de la communauté scientifique sur le sujet, et la possibilité de traiter le problème par intervalles, il a été choisi un problème d'optimisation multi-objectif de synthèse des contrôleurs PID.

Donc sur le problème général, un cas spécifique a été proposé et résolu en utilisant les algorithmes [I]RMOA I, [I]RMOA II et [I]RMOEA.

5.1 - Révision sur les Contrôleurs PID

Les contrôleurs PID, soient analogiques et digitaux, sont utilisés en diverses applications industrielles pour contrôler flux de liquides, température des fours et vitesse de moteurs par exemple. La simplicité de fonctionnement et la performance considérée optimale sont souvent citées comme motifs pour le choix de contrôleurs PID dans l'industrie.

En contrôleurs PID, l'ajuste de variables de contrôle est fait de façon automatique selon les mesures obtenues de la sortie du processus $z(t)$ comparé à une *valeur de référence* $r(t)$ dans le temps t . La différence entre les valeurs $r(t)$ et $z(t)$ est le *signal d'erreur* $e(t)$:

$$e(t) = r(t) - z(t), \quad (5.1)$$

que se désire minimiser ou éliminer, en stabilisant $z(t)$ au niveau plus proche de $r(t)$. L'action de contrôle agit dans le processus de façon proportionnel, intégrale et dérivatif conforme l'expression :

$$m(t) = K_p e(t) + K_i \int_0^t e(t) dt + K_d \frac{de(t)}{dt}, \quad (5.2)$$

soient $m(t)$ la sortie du contrôleur, K_p le gain proportionnel, K_i le gain intégrale et K_d le gain dérivatif. Un schéma simple d'un contrôleur PID associé à un processus générique ou une *planta* a été présenté dans la Fig.5.1.

Le vecteur de paramètres $\mathbf{K} = \{K_p, K_i, K_d\}$ du PID est ajusté en fonction du plant à être régler. La variable pour être manipulé assume les valeurs définies par le contrôleur, guidées par l'erreur existant entre $r(t)$ et $z(t)$. La tâche de sélectionner les valeurs appropriées pour les gains \mathbf{K} est appelé de *calibration ou régulation de PID*. Il y a différentes méthodes pour le calibrage de ces contrôleurs, lesquels ont été cités sur le texte principal. La méthode employé ici est basée en optimisation.

Références: Astrom et Hugglund [5], Ziegler-Nichols [127], Cohen-Coon [22], Visioli [120], Wang et Cai [123], et Chidambaram et Sree [19].

5.2 - Description du Problème

L'optimisation de systèmes de contrôle PI, PD ou PID pour un plant générique signifie trouver le mieux ensemble de paramètres K_p , K_i et K_d que minimise quelques métriques/spécifications de performance de la réponse indicielle du système. Les spécifications dans le domaine du temps sont observées dans les systèmes où la réponse est considéré *stable*. La Fig.5.2 a présenté l'identification de ces métriques via un exemple de réponse indicielle d'un système en boucle fermée.

Le *dépassement* M_p est la valeur de la réponse indicielle du système au-dessus de $r(t)$. Le *temps de montée* t_s marque la vitesse du système pour répondre à l'entrée. Ils sont valeurs typiques à t_s : a) de 10% à 90%, b) de 5% à 95% e c) de 0% à 100%. Dans ce travail, l'option (c) a été choisie pour t_s . De même façon, le *temps de sommet* t_p est le temps mesuré dès le début du procédé jusqu'au plus haut dépassement M_p . Le *temps de réponse* t_a marque la stabilisation de la réponse du système dans une région stable (tolérance) proche de la valeur $r(t)$ que se désire obtenir. La tolérance permise est normalement définie en tour de $\pm 2\%$ à $\pm 5\%$. Excepte dans certaines applications, il est désiré que la réponse indicielle soit suffisante rapide et amorti. Ainsi, les métriques des réponses listées antérieurement, il est désiré que le processus d'optimisation minimise M_p , t_s , t_p et t_a . En addition, il a été choisi une métrique basée en erreur accumulée pour évaluer la qualité de la réponse du système. Il a été choisi la *métrique ISE* (Integral Square-Error) donné par l'équation :

$$ISE = \int_0^{\infty} e^2(t)dt. \quad (5.3)$$

Les mesures de performance et de qualité décrites en haut ont été utilisées pour formuler le problème d'optimisation de synthèse de contrôleurs PID. Cependant, avant de décrire le problème, il a besoin établir les conditions de stabilité du système. Normalement, la stabilité est une caractéristique fondamentale en systèmes avec rétroaction, soit comme considérée une pré-condition de projet de systèmes de contrôle linéaires.

Le critère RH (Routh-Hurwitz) a été la méthode choisie pour vérifier la stabilité du système. Il utilise seulement les coefficients de l'équation caractéristique (dénominateur da fonction de Laplace de boucle fermée) pour conclure sur l'existence de stabilité. L'équation caractéristique est donnée par :

$$a_n s^n + a_{n-1} s^{n-1} + \dots + a_1 s + a_0, \quad a_n \neq 0, \quad (5.4)$$

soient s la variable de Laplace et les coefficients a_n, \dots, a_0 sont grandeurs réelles. En partant de (5.4) la Tab.5.1, dit Table RH, a été construit pour déterminer les conditions RH de stabilité.

Table 5.1: Coeficientes RH.

s^n	a_n	a_{n-2}	a_{n-4}	\dots
s^{n-1}	a_{n-1}	a_{n-3}	a_{n-5}	\dots
s^{n-2}	b_1	b_2	b_3	\dots
s^{n-3}	c_1	c_2	c_3	\dots
\vdots	\vdots	\vdots	\vdots	\vdots

Les coefficients b_1, b_2, \dots sont :

$$b_1 = \frac{a_{n-1}a_{n-2} - a_n a_{n-3}}{a_{n-1}}, \quad b_2 = \frac{a_{n-1}a_{n-4} - a_n a_{n-5}}{a_{n-1}}, \quad \dots \quad (5.5)$$

Et les coefficients c_1, c_2, \dots sont :

$$c_1 = \frac{b_1 a_{n-3} - b_2 a_{n-1}}{b_1}, \quad c_2 = \frac{b_1 a_{n-5} - b_3 a_{n-1}}{b_1}, \quad \dots \quad (5.6)$$

Le système est déclaré stable si les termes de la première colonne sont tous positifs ou tous négatifs.

Le critère RH peut être étendu quand les coefficients de l'équation caractéristique sont intervalles. Mais, il y a plusieurs occurrences de mêmes coefficients comme se peut voir en (5.5) et (5.6), en résultant une analyse de stabilité très pessimiste. Donc, Kharitonov a développé une technique pour traiter les coefficients intervalles. Avant, de l'introduction de la technique de Kharitonov, considère l'équation caractéristique intervalle :

$$[a_n]s^n + [a_{n-1}]s^{n-1} + \dots + [a_1]s + [a_0], \quad 0 \notin [a_n], \quad (5.7)$$

et maintenant, les polynômes de Kharitonov sont :

$$p_1 : \bar{a}_n s^n + \underline{a}_{n_1} s^{n-1} + \underline{a}_{n-2} s^{n-2} + \bar{a}_{n-3} s^{n-3} + \bar{a}_{n-4} s^{n-4} + \dots \quad (5.8)$$

$$p_2 : \bar{a}_n s^n + \bar{a}_{n_1} s^{n-1} + \underline{a}_{n-2} s^{n-2} + \underline{a}_{n-3} s^{n-3} + \bar{a}_{n-4} s^{n-4} + \dots \quad (5.9)$$

$$p_3 : \underline{a}_n s^n + \bar{a}_{n_1} s^{n-1} + \bar{a}_{n-2} s^{n-2} + \underline{a}_{n-3} s^{n-3} + \underline{a}_{n-4} s^{n-4} + \dots \quad (5.10)$$

$$p_4 : \underline{a}_n s^n + \underline{a}_{n_1} s^{n-1} + \bar{a}_{n-2} s^{n-2} + \bar{a}_{n-3} s^{n-3} + \underline{a}_{n-4} s^{n-4} + \dots \quad (5.11)$$

Le système est déclaré stable si p_1, p_2, p_3 et p_4 sont stables, selon le critère RH. Pour systèmes d'ordre $n \leq 5$, les conditions de stabilité sont plus simples, comme a été proposé par Anderson au-dessous :

$$n = 2 : \text{ stable si } a_2, a_1, a_0 < 0 \text{ ou } a_2, a_1, a_0 > 0 \quad (5.12)$$

$$n = 3 : \text{ stable si } p_1 \text{ for RH} \quad (5.13)$$

$$n = 4 : \text{ stable si } p_1, p_2 \text{ forem RH} \quad (5.14)$$

$$n = 5 : \text{ stable si } p_1, p_2, p_3 \text{ forem RH} \quad (5.15)$$

Une fois que les conditions de stabilité sont satisfaites, le problème de contrôle PID peut être formulé.

En récente recherche sur le travail de contrôle robuste de plantes intervalles, il a été observé un grandissant intérêt de la communauté scientifique sur le sujet. Il a des études qu'associe le terme "contrôle robuste" à capacité du système à maintenir la stabilité quand la valeur de ses paramètres change. Dans l'optimisation, le but souvent est trouver le mieux ensemble de paramètres que maximise la performance de la réponse indicielle, dans une région garantie robuste. D'entre les méthodes étudiés, un algorithme génétique a été employé pour minimiser l'erreur ISE dans une formulation mono-objectif qui utilisait le pire cas du scénario de la fonction de contrainte. En ce cas, les incertitudes ont été considérées sur les paramètres du plant ; les paramètres des contrôleurs ont été fixés. Dans une autre étude, des plants intervalles ont été utilisé pour trouver les poles du système, et aussi en partant d'équation caractéristique intervalle des paramètres de contrôleurs intervalles ont été obtenus via l'équation Diophantine intervalle. Finalement, le problème PID a été formulé comme un problème multi-objectif, soit le plant à être optimisé dans la forme d'intervalles. Les références sur ces travaux de recherche sont présentées à la fin de la section. La formulation du problème robuste choisie dans cette thèse peut être considéré un "mélange" de ces études, car: a) le traitement de la stabilité est fait par moyen d'intervalles ; b) les contrôleurs sont écrits en format d'intervalles ; et c) la formulation du problème est multi-objectif. Ainsi, soient le vecteur \mathbf{K} les gains du système et \mathbf{p} le paramètre d'incertitude, un problème robuste multi-objectif pour la synthèse de contrôleurs PID d'un plant générique de ordre n peut être écrit comment :

$$\begin{aligned}
& \min_{\mathbf{K} \in [\mathbf{K}]} \max_{\mathbf{p} \in [\mathbf{P}]} f_1, f_2, f_3, f_4 \\
\text{s.c :} & \quad g_1 = \text{critère RH} ; \\
& \quad g_2 = \max(z(\mathbf{K}, \mathbf{p}, t)) < 1.3u(t), \quad t = t_0 ; \\
& \quad g_3 = \text{singularités (Analyse par Intervalles)} ; \\
& \quad g_4 = \text{autres singularités.} \tag{5.16}
\end{aligned}$$

Soient :

$$\begin{aligned}
f_1 &= z(\mathbf{K}, \mathbf{p}, t), \quad t \in [0, \infty] ; \\
f_2 &= \arg \max_{t \in [0, \infty]} (f_1) ; \\
f_3 &= \min_{t \in [0, \infty]} \{t \mid z(\mathbf{K}, \mathbf{p}, t) = r(t)\} ; \\
f_4 &= \text{ISE.}
\end{aligned}$$

Normalement, le dépassement M_p (f_1) et le temps de sommet t_p (f_2) sont spécifications contradictoires, cependant que le temps monté t_s (f_3) et t_p ne sont pas. Par fois, il y a des situations que M_p et t_p n'existe pas comme dans les systèmes sur-amortis. La métrique ISE (f_4) inséré bénéficie les réponses avec mineur erreur accumulée dans période transitoire. En (5.16), la contrainte g_1 est lié à la vérification de la stabilité via Routh-Hurwitz, Kharitonov et Anderson. La contrainte g_2 disqualifie les réponses "mauvaises" en régime stationnaire, définies au temps $t = t_0$. Et, g_3 garantit l'utilisation de méthodes intervalles. Par exemple, considère le intervalle $[a, b]$, $a \in \mathbb{R}$ et $b \in \mathbb{R}$. Si $a < 0$ et $b > 0$, les calculs $\sqrt{[a, b]}$ et $1/[a, b]$ représentent des singularités, car ces opérations ne sont pas définies en Analyse par Intervalles basée en systèmes finis, qui est le cas cette thèse. Finalement, g_4 traite d'autres singularités. Par exemple, g_4 évite que la formulation géré des résultats impossibles tels que $t_s < 0$.

Références: Ogata [91], Kharitonov [70], Anderson et al. [3], Dahleh et al. [25], Pujara et Roy [94], Henrion et Bachelier [55], Hsu et Yu [59], Lordelo et al. [77], Takahashi et al. [117].

5.3 - Étude de Cas

La formulation proposée en (5.16) peut être employé dans plusieurs problèmes de systèmes de contrôle via PID. Donc, il est nécessaire préciser l'étude de cas abordé ici. Le cas considéré est simple. Cependant, la finalité est détailler le processus d'optimisation robuste via les algorithmes proposés. Le lecteur intéressé peut étendre le procédure à problèmes que la fonction de transference soit de ordre $n \leq 4$. La limitation d'ordre se justifie, car il y a besoin de définir la fonction d'inclusion du système de forme analytique.

Ainsi, si la réponse du système dépend du calcul de racines du polynôme caractéristique, alors la solution analytique “fermée” seulement existe pour équations de quatrième ordre.

Pour mieux description de cette étude de cas, cette section a été fragmentée dans les parts suivantes : a) Caractérisation du Plant ; b) Réponse Temporelle du Système ; c) Formulation du Problème Multi-Objectif Robuste ; d) Fonctions Objectif ; d) Analyse de las Fonctions Objectif et de Contrainte ; et e) Expérimente et Résultats.

5.3.1 - Caractérisation du Plant

Dans l'étude de cas, il est considéré un système de contrôle linéaire, invariant au temps, sans retard, sans bruit d'entrée et avec le contrôle analogique proportionnel plus dérivatif pour répondre à une entrée échelon unitaire. Le système a été présenté dans la Fig.5.3. La fonction de transference resultant $G(s)$ pour le cas est donné par :

$$G(s) = \frac{H(s)C(s)}{1 + H(s)C(s)} = \frac{K_p + (K_d)s}{s^2 + (1 + K_d)s + K_p} \quad (5.17)$$

5.3.2 - Réponse Temporelle du Système

En appliquant un échelon unitaire ($r(t) = 0, t = 0; r(t) = 1, t > 0$) et réalisant la transformé inverse Laplace est obtenu la réponse du système $z(K_p, K_d, t)$ suivante :

$$z(K_p, K_d, t) = 1 + e^{-\frac{(1+K_d)}{2}t} \left(-\cosh\left(\frac{\sqrt{\alpha}}{2}t\right) + \frac{(-1 + K_d)}{\sqrt{\alpha}} \sinh\left(\frac{\sqrt{\alpha}}{2}t\right) \right) \quad (5.18)$$

soient :

$$\alpha = (K_d + 1)^2 - 4K_p. \quad (5.19)$$

En considérant l'incertitude p dans la précision du contrôleur analogique, les gains K_p et K_d doivent être récrits comment $K_p \pm p$ et $K_d \pm p$. L'action des contrôleurs K_p et K_d sur le plant est linéaire, cependant l'effet de l'incertitude p sur K_p et K_d n'est pas triviale. Ainsi, en considérant p , les formulations en (5.18) et (5.19) peuvent être récrits comment :

$$\begin{aligned}
z(K_p, K_d, p, t) &= 1 + e^{-\frac{(1+K_d+p)}{2}t} \left(-\cosh\left(\frac{\sqrt{\alpha'}}{2}t\right) \right) + \dots \\
&\dots + e^{-\frac{(1+K_d+p)}{2}t} \left(\frac{(-1 + K_d + p)}{\sqrt{\alpha'}} \sinh\left(\frac{\sqrt{\alpha'}}{2}t\right) \right)
\end{aligned} \tag{5.20}$$

soient :

$$\alpha' = (K_d + p + 1)^2 - 4(K_p + p). \tag{5.21}$$

5.3.3 - Formulation du Problème Multi-objectif Robuste

En considérant la réponse temporelle du système en (5.20) et (5.21), le problème multi-objectif robuste pour l'étude de cas é défini comment :

$$\begin{aligned}
&\min_{K_p, K_d \in [0, 15]} \quad \max_{p \in [-0.1, 0.1]} \quad f_1, f_2, f_3, f_4 \\
\text{s.c.} : & \quad g_1 = s^2 + (1 + K_d + p)s + (K_p + p) \text{ soit RH ;} \\
& \quad g_2 = \max(z(K_p, K_d, p, t)) < 1.3, \quad t = 3s ; \\
& \quad g_3 = \alpha' > 1 ; \\
& \quad g_4 = \frac{-\sqrt{\alpha'+1}-(Kd+p)}{\sqrt{\alpha'+1}-(Kd+p)} > 1, \quad 0 \notin \sqrt{\alpha'} + 1 - (Kd + p). \tag{5.22}
\end{aligned}$$

$$\begin{aligned}
\text{Soient :} \quad & f_1 = z(K_p, K_d, p, t), \quad t \in [0, \infty] ; \\
& f_2 = \arg \max_{t \in [0, \infty]} (f_1) ; \\
& f_3 = \min_{t \in [0, \infty]} \{t \mid z(K_p, K_d, p, t) = r(t)\}, \quad r(t) = 1 ; \\
& f_4 = \text{ISE.}
\end{aligned}$$

5.3.4 - Fonctions Objectif

Les fonctions objectif f_1 et f_2 traitent du dépassement M_p et du temps de sommet t_p de $z(K_p, K_d, p, t)$ (5.20). Pour fonctions continues, le dérivatif a valeur nulle sur les points maximums et minimums. Donc, M_p et t_p sont calculés quand :

$$\frac{dz(K_p, K_d, p, t)}{dt} = 0, \tag{5.23}$$

en résultant en :

$$t_p = \frac{1}{\sqrt{\alpha'}} \ln \left(\frac{1 - (\sqrt{\alpha'} + (K_d + p))^2}{1 - (\sqrt{\alpha'} - (K_d + p))^2} \right), \quad (5.24)$$

et

$$M_p = z(K_p, K_d, p, t_s) - 1. \quad (5.25)$$

La déduction de les équations (5.24) et (5.25) ont été fait dans l'Appendice A en considérant un modèle sans le paramètre d'incertitude p et pour valeurs fixes de K_p et K_d .

La fonction objectif f_3 , que traite du temps de monté t_s , est aussi obtenue de $z(K_p, K_d, p, t)$. En considérant t_s comme le période pour le système aller atteindre la référence ($r(t) = 1$), t_s est formulé comment :

$$t_s = \frac{1}{\sqrt{\alpha'}} \ln \left(\frac{-\sqrt{\alpha'} + 1 - (K_d + p)}{\sqrt{\alpha'} + 1 - (K_d + p)} \right) \quad (5.26)$$

Le développement de l'équation t_s a été réalisé dans l'Appendice B, que sans perdre de généralité, ont été fixés K_p et K_d , et le paramètre p n'a pas été considéré.

La fonction f_4 , réfèrent à métrique ISE, a été calculé de l'équation intégrale (5.3). La résolution de l'intégral, a été fait dans l'Appendice C, a produit comme résultat :

$$\begin{aligned} \text{ISE} &= -\frac{\left(1 + \frac{1-(K_d+p)}{\sqrt{\alpha'}}\right)^2}{\sqrt{\alpha'} - 1 - (K_d + p)} \dots \\ &\dots -\frac{2\left(1 - \left(\frac{1-(K_d+p)}{\sqrt{\alpha'}}\right)^2\right)}{-1 - (K_d + p)} - \frac{\left(1 - \frac{1-(K_d+p)}{\sqrt{\alpha'}}\right)^2}{-\sqrt{\alpha'} - 1 - (K_d + p)}. \end{aligned} \quad (5.27)$$

Les fonctions objectif d'inclusion pour cette application sont fonctions d'inclusion naturelles. Les fonctions objectif formulés en (5.24-5.27) contient seulement des opérateurs mathématiques élémentaires, donc les fonctions d'inclusion sont convergentes et monotones. Cependant, les caractéristiques

d'être minimal et fine dépendent du numéro d'occurrences des variables et du type d'opération qu'existe entre eux. Ce fait justifie les simplifications réalisées dans les appendices.

5.3.5 - Analyse de las Fonctions Objectif et de Contrainte

Le traitement de contraintes formulé en (5.22) est similar au procedimento présenté à la Sous-section 3.5.1. Dans cette étude de cas, les tests d'inclusion classifient tout l'espace de recherche en "réalisable" ou "de frontière", selon la précision donnée. Aucun espace n'a été classifié comment "non réalisable" car les fonctions de contraintes n'ont pas été construites pour garantir l'instabilité. La Fig.5.4 présente l'espace réalisable de l'étude de cas. La région en gris foncé représente l'espace réalisable ; en gris clair, les intervalles de frontières. Détaillées pour obtention de la figure: a) algorithme SIVIA (2.3.12) ; b) paramètre de précision $\varepsilon = 0.1$; et c) 590 boîtes réalisables.

Antérieurement, il a été dit que la fonction d'inclusion de l'étude de cas est convergente et monotone. Le Fig.5.5, le Fig.5.6 et la Tab.5.3 confirment, à travers d'un exemple, les propriétés listées en haut importantes pour l'utilisation de méthodes intervalles. Dans l'exemple, notez la propriété $[\mathbf{K}]_2 \subset [\mathbf{K}]_1 \subset [\mathbf{K}]_0 \Rightarrow \mathbf{f}([\mathbf{K}]_2) \subset \mathbf{f}([\mathbf{K}]_1) \subset \mathbf{f}([\mathbf{K}]_0)$. Si les fonctions ne sont pas convergentes et monotones, alors elles ne seront pas utiles, car c'est impossible d'éliminer régions avec sécurité et garantie la globalité des méthodes par intervalles, par exemple.

5.3.6 - Expérimente et Résultats

L'expérimente a consisté d'utiliser les méthodes [I]RMOA I, [I]RMOA II et [I]RMOEA, validés dans le Chapitre 4, pour résoudre le problème multi-objectif robuste décrit en (5.22). Le problème a quatre objectifs, donc la visualisation graphique a été limitée pour présenter trois objectifs simultanément. Les résultats d'optimisation de (5.22) ont été organisé dans quatre blocs: a) les enveloppes sur la frontière robuste ; b) les ensembles solutions ; c) les frontières robustes ; et d) l'analyse des solutions choisies dans la réponse du système de contrôle.

Les paramètres de l'exécution sont présentés ensuite. Conforme la formulation (5.22), ils ont été employés $K_p = [0, 15]$, $K_d = [0, 15]$ et $P = [-0.1, 0.1]$. Aussi, les paramètres spécifiques de chaque algorithme sont: a) $\varepsilon_{x\#} = 0.25$ et $\varepsilon_{[x]} = 0.05$ pour le [I]RMOA I ; b) $\varepsilon_{[x]} = 0.05$ et $\varepsilon_{[p]} = 0.025$ pour le [I]RMOA II ; et c) $p_c = 1$, $p_m = 0.0005$, $n_{pop} = 100$, $n_{gen} = 60$, $n_{bit} = 15$, $\varepsilon_{[p]} = 0.025$ et $r_{bis}^{max} = 8$ pour le [I]RMOEA. Dans l'expérimente qu'analyse

les enveloppes, ils ont été aussi utilisés $\varepsilon_{[\mathbf{p}]} = 0.005$ et $\varepsilon_{[\mathbf{p}]} = 0.001$ pour les méthodes [I]RMOA II et [I]RMOEA.

Dans les premiers tests, le [I]RMOA I a confronté le [I]RMOA II et le [I]RMOEA. Les résultats ont été montrés en Fig.5.7 et Fig.5.8. Dans ces figures, les enveloppes 3D pour \mathbf{K}^- et \mathbf{K}^+ ont été remplacées par une visualisation plus simplifiée, qu'utilise seulement des points sur le graphique que définissent \mathbf{K}^- et \mathbf{K}^+ . Ainsi, la comparaison entre les algorithmes est devenu plus claire. L'enveloppe produite par [I]RMOA I a été comparée avec trois cas des autres algorithmes. Les cas de simulation consistent en vérifier les valeurs de précision du paramètre d'incertitude $\varepsilon_{[\mathbf{p}]}$ que produisent des frontières robustes correctes. Les résultats des les simulations (voir Fig.5.7 et Fig.5.8) ont présenté la dépendance de la frontière robuste de $\varepsilon_{[\mathbf{p}]}$. Cette dépendance est conséquence que les fonctions objectif d'inclusion n'est sont pas minimales. Aussi, à travers de ces tests, le [I]RMOA I se confirme comme outil de validation.

Les ensembles solutions ont été analysés dans la deuxième batterie de tests. Les résultats du [I]RMOA II et du [I]RMOEA ont été présenté en Fig.5.9 et Fig.5.10 respectivement. En observant les résultats, il est possible conclure: a) l'ensemble solution trouvée se situe en $K_p = 15$ et $K_d \in [9.5, 13.5]$; b) l'ensemble solution n'est pas dépendent de $\varepsilon_{[\mathbf{p}]}$, mais l'image de l'ensemble l'est; et c) l'uniformité de distribution des solutions est satisfaisante.

Les frontières ont été aussi analysées en considérant les mêmes valeurs de $\varepsilon_{[\mathbf{p}]}$ pour contraster les algorithmes [I]RMOA II et [I]RMOEA. Les résultats (voir Fig.5.11) ont montré cohérence entre les deux méthodes. Une fois que les deux algorithmes sont différents, cette information est suffisante pour valider les deux et aussi les ensembles solutions trouvées par eux.

Le problème d'optimisation multi-objectif proposé en (5.22) ne peut pas être considéré conclu avec seulement l'ensemble solution et son image robuste. Pour compléter la tâche d'optimisation multi-objectif, il est nécessaire prendre la décision sur une solution spécifique choisir, c'est à dire, c'est important déterminer la *solution robuste de projet* $\mathbf{K}^{**} = \{K_p^{**}, K_d^{**}\}$ qu'optimise l'action du contrôleur. Deux \mathbf{K}^{**} ont été choisi: un ensemble du [I]RMOA II, et l'autre du [I]RMOEA. Le critère adopté pour la prise de décision a été basé sur une sorte de fonction objectif normalisée. Ainsi, en dénotant l'ensemble solution robuste par \mathbf{K}^* et par $f_i^{\max}(\mathbf{K}^*, \mathbf{p})$ le maximum valeur de performance de la fonction objectif f_i selon \mathbf{K}^* , \mathbf{K}^{**} est formulé comme :

$$\mathbf{K}^{**} = \arg \min_{\mathbf{k}^* \in \mathbf{K}^*} \sum_{i=1}^{n_f} \frac{f_i^{\max}(\mathbf{K}^*, \mathbf{p}) - f_i(\mathbf{k}^*, \mathbf{p})}{f_i^{\max}(\mathbf{K}^*, \mathbf{p})}, \quad \mathbf{p} \in \mathbf{P}, \quad (5.28)$$

soient $f_i^{\max}(\mathbf{K}^*, \mathbf{p}) > 0$ garanti par (5.24)-(5.27). Ainsi, en considérant la précision $\varepsilon_{[\mathbf{p}]} = 0.001$, la solution robuste choisi par [I]RMOA II a été $K_p^{**} = 14.9854$ et $K_d^{**} = 9.6240$, et pour le [I]RMOEA a été $K_p^{**} = 15.0000$ et $K_d^{**} = 9.6088$. Selon les métriques de performance du système formulé en (5.24)-(5.27), la solution du [I]RMOA II ($K_p^{**} = 14.9854$, $K_d^{**} = 9.6240$) a produit $t_s = 0.3411$, $t_p = 0.5730$, $M_p = 0.0305$ et $ISE = 0.2031$, et le [I]RMOEA ($K_p^{**} = 15.0000$, $K_d^{**} = 9.6088$) en $t_s = 0.3409$, $t_p = 0.5728$, $M_p = 0.0307$ et $ISE = 0.2034$. La réponse du système en considérant la solution du [I]RMOA II a été montré dans la Fig.5.12. Le résultat du [I]RMOEA est similaire, et donc il a été omis. Observez dans la figure que la réponse indicielle a été réellement optimale, car la réponse a été stable, rapide, presque sans dépassement et avec la réponse en régime dedans la tolérance et le plus important, en prisant compte les incertitudes de la formulation.

5.4 - Considérations

Cette section a présenté le résumé du chapitre.

Chapitre 6 - Conclusions et Perspectives

6.1 - Conclusions

Le choix d'étudier et proposer des algorithmes pour traiter la philosophie d'optimisation robuste que considère le pire cas est lié à décision d'utiliser Analyse par Intervalles ; le calcul des incertitudes est fait de façon naturelle dans les méthodes par intervalles. En traitant les incertitudes du système d'optimisation comme un paramètre isolé, le calcul d'incertitude peut être encore plus intuitif. Si le paramètre d'incertitude apparie une seule fois dans l'expression formelle de la fonction objectif robuste, donc avec une seule évaluation de la fonction objectif, il est possible à calculer tout l'incertitude avec le minimum pessimisme (voir Sous-section 3.5.1). S'il y a multiplicité d'occurrences du paramètre d'incertitude, alors il est nécessaire crier les sous-pavages pour calculer plus précisément la performance robuste d'une solution particulière. Il est important registrer qu'il y des avantages et des inconvénients de considérer le pire cas. Par exemple, dans l'optimisation de dispositifs, l'image d'une solution robuste peut, souvent, être irréalizable

dans l'espace des objectifs. Par contre, il y a des situations de risque tel qu'il soit inadmissible une erreur. Donc, le pire cas est la philosophie indiquée pour ce dernier scénario robuste.

Le problème d'optimisation robuste est plus complexe que l'optimisation traditionnelle, car il considère les définitions $\mathbf{f}(\mathbf{x}, \mathbf{p}) : \mathbb{R}^{n_x} \times \mathbb{R}^{n_p} \mapsto \mathbb{R}^{n_f}$, $\mathbf{x} \in \mathbb{R}^{n_x}$ et $\mathbf{p} \in \mathbb{R}^{n_p}$. Par conséquent, l'ordre du système augmente par une facteur $n_x \times n_p$. Si le problème d'optimisation envelopper seulement fonctions convexes, la résolution du problème est simple, rapide et précise à travers d'application de méthodes dirigées par dérivatifs. Cependant, dans la phase de planification de la thèse il a été défini en développer méthodes pour l'optimisation robuste indépendante des caractéristiques particulières de fonctions objectif et de contraintes, par exemple non convexité et discontinuité de la frontière robuste.

Les méthodes proposées [I]RMOA I, [I]RMOA II et [I]RMOEA présentent, en cet ordre, l'évolution temporelle de la création des algorithmes dans cette thèse. Chaque un de ces algorithmes a eu des contributions. Premièrement, l'abstraction de la frontière Pareto robuste a été fondé avec les résultats du [I]RMOA I, à travers de la construction d'enveloppes sur la frontière robuste. Une fois que le [I]RMOA I est calibré pour un spécifique \mathbf{P} , la méthode est prêt à gérer l'autre enveloppe pour un autre scénario robuste différent de \mathbf{P} . De l'autre côté, le [I]RMOA II recherche solutions robustes à partir d'une boîte que contient toute la région d'espace de recherche. Les régions irréalisables sont éliminées pour les fonctions de contraintes, écrites sur la forme d'intervalles. En utilisant la philosophie pessimisme, les régions réalisables seulement sont éliminées lorsque avoir garanti que celle région ne contient pas de solutions robustes. Ainsi, le point fort du [I]RMOA II est la garanti d'enregistrer tous les points de la frontière, selon une précision, dedans les intervalles non éliminés. Tant [I]RMOA I tant [I]RMOA II sont dépendants de las dimensions des variables de projet et le paramètre d'incertitudes, cas les bisections utilisant ces espaces. Par contre, dans le [I]RMOEA la recherche est stochastique et l'influence des dimensions n'augmente pas l'effort computationel. Cependant, le [I]RMOEA ne garante pas trouver les solutions robustes. La contribution des méthodes intervalles est limité au calcul de l'interférence des incertitudes et aussi dans la technique de niche NB[I]. D'entre autres aspects, une caractéristique vérifiée dans les algorithmes [I]RMOA I et [I]RMOA II, et commun dans les méthodes intervalles, est la convergence du procédé de recherche, c'est à dire, les étapes dans les processus de recherche sont tous déterministes, et les algorithmes finissent de façon globale et naturelle selon les valeurs des paramètres de précision définis. Par contre, dans le [I]RMOEA la convergence est déterminée en fixant un numéro maximum d'itérations, ou à travers d'autre critère em-

pirique qu'évalue la convergence. Finalement, les méthodes [I]RMOA I et [I]RMOA II ont seulement deux paramètres à régler, et le [I]RMOEA a plusieurs. Cette information est cohérente avec la nature déterministe des algorithmes [I]RMOA I et [I]RMOA II, et stochastique-déterministe du [I]RMOEA. La Tab.6.1 résume la comparaison des algorithmes proposés. Notez dans la table qu'il y a des items comme "coût" avaient eu une évaluation subjectif. Ces évaluations sont parties d'observations durant la phase de tests.

Table 6.1: Comparaison entre [I]RMOA I, [I]RMOA II et [I]RMOEA.

Attribut	[I]RMOA I	[I]RMOA II	[I]RMOEA
Classification	déterministe	déterministe	hybride
Recherche	globale	globale	globale
But	envelopper \mathbf{Y}^*	envelopper \mathbf{X}^*	trouver \mathbf{X}^*
Qualité \mathbf{Y}^*	non applicable	haut	moyen/haut
Convergence	oui	oui	dépend critère
Garanti	dépend $\{\mathbf{f}, \varepsilon_{\mathbf{x}\#}, \varepsilon_{[\mathbf{x}]}\}$	oui	non
Coût	haut	moyen/haut	bas
Dimension	dépendante	dépendante	indépendante
Paramètres	peu	peu	plusieurs

En analysant les expériences réalisés pour le [I]RMOEA, il a été considéré que l'uniformité dans la frontière de Pareto robuste a été obtenue avec succès, et avec coût bas. Le succès de l'uniformité est dû à la technique de niche NB[I]. Aussi, la métrique MB[I] s'a présentée comme une technique efficace pour mesurer la qualité d'uniformité dans l'ensemble de Pareto (voir les comparaisons à Sous-section 3.7.2).

Les méthodes d'optimisation par intervalles ont besoin de construire des fonctions d'inclusion convergentes, monotones, minimales et fines. Une fois que ces caractéristiques sont satisfaites, les méthodes par intervalles peuvent éliminer avec sécurité les régions où les solutions du système ne sont pas. En fait, les caractéristiques d'être minimales et fines sont importantes principalement pour réduire le pessimisme. D'entre les fonctions test utilisés, la fonction d'inclusion naturelle à ZDT3 a présenté pire cas de pessimisme, car il y a des multiplicités de paramètres. Comme conséquence, le [I]RMOA I que dépende d'exhaustifs calculs de fonctions d'inclusion a eu sa performance préjudicié. Dans l'application réelle de synthèse de contrôleurs, la construction de fonctions d'inclusion a été une tâche plus difficile. Pour systèmes d'ordre supérieur à quatre, la construction de fonctions d'inclusion c'est plus compliqué, car n'existent pas des solutions analytiques par ces cas.

L'application des algorithmes [I]RMOA I, [I]RMOA II et [I]RMOEA à une application réelle a établi une connexion importante entre toutes les étapes du procédé d'optimisation. Le problème choisi a été la synthèse de contrôleurs PID. D'entre autres questionnes, il a été détaillé comment le problème pouvait être modélé et traité par intervalles, en considérant tous les inconvénients. La stabilité est, normalement, une caractéristique désirable dans les systèmes de boucle fermée. Donc, les critères de stabilité ont été considérés comme contraints. En addition, les autres contraintes sont liées à singularités des méthodes intervalles ont été aussi traités comme contraintes. Le résultat du traitement de contraintes (voir Fig.5.4) démontre l'efficacité des méthodes par intervalles en séparer les régions que sont réalisables de celles que non sont. L'étude de cas a été finie avec l'application des algorithmes proposés. Les résultats obtenus ont été comparés, comme les frontières et les ensembles solution. Les résultats ont montré cohérente entre eux, donc prouvant que les trois méthodes produisent solutions correctes.

Références: Boyd et Vandenberghe [13], Jaulin et al.[64], Benhamou et Older [10], et Chabert et Jaulin [17]

6.2 - Perspectives

L'optimisation robuste multi-objectif est un problème de coût haut par définition ($\mathbf{f}(\mathbf{x}, \mathbf{p}) : \mathbb{R}^{n_x} \times \mathbb{R}^{n_p}$). Donc, le plus part des perspectives sont liées améliorer le temps d'exécution des algorithmes intervalles. L'utilisation de dérivatifs en problèmes continus, l'emploi d'architecture parallèle dans quelques procédés, le enregistrement et manipulation de boîtes intervalles en arbres sont exemples de comme des algorithmes pouvant améliorer ses performances. Aussi, il est désiré développer la métrique MB[I] par traiter l'espace multidimensionnel. De plus, la technique de niche NB[I] est facile d'adapter, en principe. Donc, NB[I] sera employé à différents algorithmes évolutionnaires et aussi aux autres algorithmes proposés dans cette thèse.