

# وزارة التعليم العالي والبحث العلمي

BADJI-MOKHAR-ANNABA UNIVERCITY  
UNIVERCITE BADJI-MOKHAR-ANNABA



جامعة باجي مختار - عنابة

Le 28/06/2016

Faculté des Sciences de l'ingéniorat

Département de Génie Mécanique

## THESE

Présentée en vue de l'obtention du diplôme de DOCTORAT

# Conception fiabiliste et automatisation d'un pont roulant

Spécialité: Génie Mécanique

Option: Génie Mécanique

Par :

CHAKRI Asma

DIRECTEUR DE THESE : KHELIF Rabia      Professeur      Université Annaba

### DEVANT LE JURY

PRESIDENT : BEY Kamel      Professeur      Université Annaba

EXAMINATEURS :

- |                      |            |                             |
|----------------------|------------|-----------------------------|
| 1. LAOUAR Lakhdar    | Professeur | Université Annaba           |
| 2. CHAIB Rachid      | Professeur | Université de Constantine I |
| 3. KHOHEMANE Lakhdar | Professeur | Université de Skikda        |

## Remerciements

Je rends ma profonde gratitude à ALLAH, le tout puissant, le miséricordieux, de m'avoir appris ce que j'ignorais, de m'avoir donné la santé et tout ce dont j'ai besoin pour l'accomplissement de ce travail et de la volonté pour atteindre ce but.

Ce travail a été réalisé au Laboratoire de Mécanique Industrielle, Département de génie Mécanique, Université Badji Mokhtar Annaba.

En tout premier lieu, je tiens à remercier vivement Professeur Rabia KHELIF de l'université Badji Mokhtar, qui a accepté de diriger cette thèse. Qu'il trouve ici l'expression de ma plus grande gratitude pour son soutien bienveillant et la confiance qu'il m'a manifestés. Qu'il soit assuré de mes profondes gratitude et de mes reconnaissances.

Toute ma reconnaissance aussi au Professeur Xin-She YANG de l'université de Middlesex, Londres. Je le remercie très vivement pour avoir suivi une partie très significative de mes travaux.

Je tiens à remercier Professeur Kamel BEY de l'université Badji Mokhtar, qui m'a fait l'honneur d'être le président du jury de ma soutenance. Qu'il trouve ici toute ma gratitude et mes reconnaissances pour l'intérêt qu'il a manifesté pour ce travail.

Je remercie Professeur Rachid CHAIB de l'université de Constantine I, qui m'a fait l'honneur d'être rapporteur de thèse. Je tiens particulièrement à lui témoigner ma gratitude pour les précieux conseils et informations qu'il m'a prodigués, et pour son déplacement et le temps qu'il a consacré à l'étude de ce document.

Je remercie Professeur Lakhdar LAOUAR de l'université Badji Mokhtar, qui m'a fait l'honneur d'être rapporteur de thèse. Je lui exprime ici ma reconnaissance pour l'intérêt qu'il a manifesté dans ce travail. Qu'il trouve ici l'expression de ma profonde gratitude.

Je remercie Professeur Lakhdar KHOCHMANE de l'université de Skikda, qui m'a fait l'honneur d'être examinateur de thèse. Qu'il trouve ici l'expression de ma vive gratitude et ma sincère reconnaissance, et pour le déplacement et le temps qu'il a consacré à l'étude de ce document.

## Remerciements

---

J'adresse également mes plus sincères salutations à ma famille et mes amis, pour leur soutien moral qu'ils m'ont apporté pour mener à bien cette thèse.

Enfin, mes sincères remerciements vont à Monsieur Haroun RAGUEB pour son aide précieuse dans l'élaboration de ce travail.

## Résumé

Dans cette thèse, une nouvelle version améliorée de l'algorithme des chauves-souris, appelé algorithme des chauves-souris directionnelles (dBA) est présentée. En se basant sur le phénomène d'écholocation directionnelle, quatre modifications ont été introduites pour améliorer les performances de l'algorithme des chauves-souris. Une série de tests a été réalisée, et les résultats montrent clairement la supériorité de cet algorithme. Par la suite, ce dernier est utilisé pour développer une nouvelle méthode d'estimation de la fiabilité des structures. Plusieurs benchmarks et cas pratiques ont été considérés, et les résultats ont dévoilé la fiabilité et la robustesse de cette méthode. En se basant sur le concept de l'espace de conception fiable, dBA a été adapté pour résoudre les problèmes d'optimisation basée sur la fiabilité. A l'instar des méthodes classiques qui sont destinées à résoudre une certaine classe de problèmes, notre méthode d'optimisation fiabiliste basée sur dBA a un spectre d'application plus général. Afin d'évaluer les capacités des algorithmes et méthodes, proposées dans cette thèse, à résoudre des problèmes industriels, une analyse de fiabilité et une optimisation fiabiliste de la poutre principale d'un pont roulant à bipoutre sont considérées. Enfin, une proposition sur l'automatisation d'un pont roulant est présentée. L'idée de base consiste à utiliser dBA pour rendre le mouvement du pont roulant autonome et sans risque de collision.

**Mots clés :** *Conception, Fiabilité, Pont roulant, dBA, Optimisation, Automatisation.*

## Abstract

In this thesis, a new improved version of the bat algorithm, named directional bat algorithm (dBA), is presented. Based on the directional echolocation behavior of bats, four modifications have been embedded to the standard bat algorithm with the aim to enhance its performance. A set of tests have been conducted, and the results reveal the superiority of the directional bat algorithm. After that, the former algorithm has been used to develop a new reliability assessment method. Several benchmarks and practical cases have been analyzed and the results show the reliability and robustness of this new method. Using the concept of the reliable space design, dBA has been adjusted to perform reliability based-design optimization (RBDO). Unlike the classical methods, which they are dedicated to certain kinds of problems; our RBDO method based on dBA is more suitable for general use. To evaluate the performance of the methods and algorithms, proposed in this thesis, in real world application, reliability assessment and reliability based-design optimization of the main girder of an overhead traveling crane are considered. Finally, a proposal on the automation of a crane is presented. The basic idea is to use dBA to make the movement of the autonomous crane and without collision.

**Keys words:** *Design, Reliability, Overhead traveling Crane, dBA, Optimization, Automating.*

## ملخص

في هذه الأطروحة ، تم تطبيق نسخة جديدة محسنة من خوارزمية الخفافيش ، تدعى خوارزمية الخفافيش الاتجاهية (DBA). وبناء على ظاهرة تحديد الموقع بالصدى الاتجاهي ، تم إجراء أربعة تعديلات لتحسين أداء خوارزمية الخفافيش. وأجريت سلسلة من الاختبارات أظهرت نتائجها بشكل واضح تفوق هذه الخوارزمية. وبناء على ذلك يمكن استخدامها لتطوير طريقة جديدة لتقدير موثوقية الهياكل. وبالأخذ بعين الاعتبار عدة معايير وحالات عملية وكذلك النتائج التي بينت موثوقية و متانة هذا الأسلوب. واستنادا إلى مفهوم فضاء التصميم الموثوق به تم تكييف خوارزمية الخفافيش الاتجاهية لحل مشاكل موثوقية المفاضلة المثلى عكس الطرق التقليدية التي تهدف إلى حل فئة معينة من المشاكل ، أما أسلوب هذه الطريقة فهو الأمثل لأنه يعتمد على موثوقية خوارزمية الخفافيش الاتجاهية والتي تتميز بفاعلية عالية. وتقييم قدرات الخوارزميات والأساليب المقترحة في هذه الأطروحة يهدف إلى حل المشاكل الصناعية. حيث تم اعتماد تحليل موثوقية ومفاضلة مثلى لرافعة بعارضة مزدوجة. وأخيرا، يتم تقديم اقتراحا بشأن الأتمتة رافعة. الفكرة الأساسية هي استخدام DBA لجعل حركة رافعة ذاتية الحكم و دون تصادم.

**كلمات البحث:** تصميم، موثوقية، رافعة، خوارزمية الخفافيش الاتجاهية، مفاضلة، أتمتة

# Sommaire

<b>Remerciements</b>	I
<b>Résumé</b>	III
<b>Sommaire</b>	VI
<b>Liste des figures</b>	X
<b>Liste des tableaux</b>	XII
<b>Introduction générale</b>	1
<b>Chapitre 1 : Etat de l'art sur les méthodes de fiabilité des structures</b>	5
1.1 Introduction	6
1.2 Analyse de la fiabilité des structures	8
1.3 Méthode de fiabilité du premier ordre (FORM)	9
1.3.1 Algorithme HLRF	10
1.3.2 Algorithme iHLRF	12
1.3.3 Algorithme nHLRF	14
1.4 Méthode de fiabilité du second ordre (SORM)	15
1.4.1 Approximation de Breitung	16
1.4.2 Approximation de Tvedt	17
1.5 Méthode de Simulation de Monte Carlo	17
1.6 Epilogue	19
<b>Chapitre 2 : Algorithme des chauves-souris directionnelles</b>	20
2.1 Introduction	21
2.2 Algorithme des chauves-souris standard	23
2.3 Enquête sur les différentes variantes de BA existant dans la littérature	25
2.4 Le nouvel algorithme de chauve-souris directionnelle	29
2.4.1 La 1 <sup>ère</sup> modification	29
2.4.1 La 2 <sup>ème</sup> modification	30
2.4.3 La 3 <sup>ème</sup> modification	31
2.4.4 La 4 <sup>ème</sup> modification	32

2.5 Simulations et discussions	33
2.5.1 La 1 <sup>ère</sup> expérience	34
2.5.2 La 2 <sup>ème</sup> expérience	44
2.5.3 La 3 <sup>ème</sup> expérience	50
2.6 Conclusions	52
<b>Chapitre 3 : Estimation de la fiabilité des structures par l'algorithme des chauves-souris directionnelles</b>	<b>54</b>
3.1 Introduction	55
3.2 Adaptation de dBA pour l'analyse de fiabilité structurelle	56
3.3 Résultats de simulation et validation	59
3.3.1 Validation et analyse de sensibilité	59
3.3.1.1 Exemple 1 : poutre en béton armé	59
3.3.1.2 Exemple 2 : une structure conique	63
3.3.1.3 Exemple 3 : systèmes en parallèle et en série	66
3.3.2 Application à quelques problèmes de fiabilité structurelle	68
3.3.2.1 Exemple4 : une poutre en porte à faux	68
3.3.2.2 Exemple 5 : un joint soudé en acier	70
3.4 Pourquoi cet algorithme ?	71
3.5 Conclusions	72
<b>Chapitre 4 : Optimisation basée sur la fiabilité par l'algorithme des chauves-souris directionnelles</b>	<b>73</b>
4.1 Introduction	74
4.2 Optimisation basée sur la fiabilité	76
4.2.1 Principes de l'RBDO	76
4.2.2 Revue sur l'ensemble des méthodes de résolution des problèmes d'RBDO	77
4.2.3 RBDO et les algorithmes métaheuristiques	79
4.3 RBDO avec l'algorithme des chauves-souris directionnelles	81
4.3.1 Description de l'approche à boucle unique adopté	81
4.3.2 Adaptation de dBA pour la résolution des problèmes d'optimisation sous contraintes	84
4.4 Exemples d'application et validation	86



4.4.1 Exemple 1 : résistance à l'impact latéral d'un véhicule	86
4.4.2 Exemple 2 : problème mathématique	93
4.4.3 Exemple 3 : conception d'un réducteur de vitesse	95
4.4.4 Exemple 4 : poutre soudée	98
5 Conclusions	103
<b>Chapitre 5 : Optimisation basée sur la fiabilité de la poutre principale d'un pont roulant</b>	<b>104</b>
5.1 Introduction	105
5.2 Revue de littérature	106
5.3 Dimensionnement de la poutre principale	109
5.3.1 Critère sur contrainte admissible en traction de l'aile inférieure	111
5.3.2 Critère sous contrainte de fatigue admissible de l'aile inférieure	113
5.3.3 Critère sur le flambement de l'aile supérieure	114
5.3.4 Critère sur le flambement de l'âme principale	115
5.3.5 Critère sur le flambement de l'âme secondaire	117
5.3.6 Critère sur la flèche maximale de la poutre	118
5.4 Optimisation de la poutre principale	118
5.4.1 Formulation du problème d'optimisation	118
5.4.2 Résultats d'optimisation et discussions	120
5.4.2.1 Analyse de sensibilité et paramétrages de dBA	120
5.4.2.2 Optimisation du pont roulant par dBA	123
5.4.3 Conclusion partielle	126
5.5 Analyse de fiabilité de la poutre optimisée	126
5.5.1 Définition du problème d'analyse de fiabilité et configuration	126
5.5.1 Résultats d'analyse de fiabilité et discussion	127
5.5.1 Conclusion partielle	131
5.6 Optimisation basée sur la fiabilité	133
5.6.1 Formulation du problème d'optimisation basé sur la fiabilité	133
5.6.2 Résultats et discussions	134
5.6.3 Conclusion partielle	138

5.7 Conclusions	138
<b>Chapitre 6 : Proposition d'un système de mouvement autonome avec évitement des obstacles pour un pont roulant basé sur dBA</b>	140
6.1 Introduction	141
6.2 Stratégie proposée	143
6.3 Simulation, résultats et discussions	146
6.4 Conclusions	148
<b>Conclusion générale et perspectives</b>	150
<b>Références</b>	153

---

## Liste des figures

<b>Fig. 1.1</b>	Schéma hypothétique de la méthode FORM.	10
<b>Fig. 1.2</b>	Schéma hypothétique de la méthode SORM.	16
<b>Fig. 1.3</b>	Principe de la simulation de Monte Carlo.	18
<b>Fig. 2.1</b>	Schématique de l'écholocation directionnelle.	30
<b>Fig. 2.2</b>	Evolution de la minimisation lors du processus itératif (F01-F10).	39
<b>Fig. 2.3</b>	Evolution de la minimisation lors du processus itératif (F11-F20).	40
<b>Fig. 2.4</b>	Représentation de SF23 à 2 dimensions	44
<b>Fig. 3.1</b>	La section transversale de la poutre en béton armé.	60
<b>Fig. 3.2</b>	Comparaison du minimum obtenu à chaque itération du problème de la poutre en béton armé.	60
<b>Fig. 3.3</b>	Effets de la forme de l'équation d'état limite et le coefficient de pénalité sur le processus de minimisation et de violation de contrainte.	61
<b>Fig. 3.4</b>	La section transversale de la structure conique.	63
<b>Fig. 3.5</b>	Une poutre en porte à faux soumise à une charge uniformément répartie.	69
<b>Fig. 4.1</b>	Les approches de résolution des problèmes d'RBDO	78
<b>Fig. 4.2</b>	Schématique du principe de l'espace fiable	82
<b>Fig. 4.3</b>	Evolution de la moyenne de 25 essais de la fonction objectif (gauche) et le taux de violation des contraintes (droite) des quatre algorithmes, dBA, BA, PSO et DE.	89
<b>Fig. 4.4</b>	Les valeurs de la fonction objectif pour chaque chauve-souris pour $t = 0, 500,$ et 1000 (Exemple 1)	91
<b>Fig. 4.5</b>	Positions des chauves-souris dans l'espace de recherche au cours de l'optimisation à $t = 0, 50$ et 100 (Exemple 2).	94
<b>Fig. 4.6</b>	La structure de la poutre soudée	99
<b>Fig. 5.1</b>	Coupe transversale de la poutre principale.	111
<b>Fig. 5.2</b>	Evolution de la minimisation de la fonction objectif au cours des itérations.	121
<b>Fig. 5.3</b>	Evolution de l'aire optimale de la section transversale de la poutre principale en fonction de la limite d'élasticité de l'acier utilisé.	124

<b>Fig. 5.4</b>	Pourcentage de réduction en masse de la poutre principale en fonction de la limite d'élasticité de l'acier utilisé.	124
<b>Fig. 5.5</b>	Etat des contraintes d'optimisation normalisées à chaque point optimal.	125
<b>Fig. 5.6</b>	Indice de fiabilité pour chaque équation d'état limite.	129
<b>Fig. 5.7</b>	Valeurs des cosinus directeurs au carré des variables aléatoires pour chaque équation d'état limite et différentes configurations du pont roulant.	130
<b>Fig. 5.8</b>	Comparaison de l'aire optimale de la section transversale de la poutre obtenue par l'RBDO , contraintes des quatre algorithmes, dBA, BA, PSO et DE.	135
<b>Fig. 6.1</b>	Organigramme du système de contrôle du mouvement proposé.	145
<b>Fig. 6.2</b>	Schéma hypothétique du mécanisme d'évitement d'obstacle.	145
<b>Fig. 6.3</b>	Trajectoires décrites par le mouvement du pont roulant.	148
<b>Fig. 6.4</b>	Trajectoires présumées sans l'existence des obstacles.	149
<b>Fig. 6.5</b>	Trajectoire du retour dans le même environnement.	149

## Liste des tableaux

<b>Tableau 2.1</b>	Les fonctions de référence classiques (F01-F17).	35
<b>Tableau 2.2</b>	Résultats statistiques de la minimisation des fonctions classiques (F01~F10).	37
<b>Tableau 2.3</b>	Résultats statistiques de la minimisation des fonctions classiques (F11~F20).	38
<b>Tableau 2.4</b>	Résultats de comparaison par paires (1 <sup>ère</sup> expérience).	41
<b>Tableau 2.5</b>	Classement des algorithmes selon les tests de Friedman, Friedman Aligné et Quade (1 <sup>ère</sup> expérience).	42
<b>Tableau 2.6</b>	Valeur de <i>p-value</i> entre dBA et les autres algorithmes selon les différentes procédures Post-hoc considérées avec un niveau de signification $\alpha = 0.05$ (1 <sup>ère</sup> expérience).	43
<b>Tableau 2.7</b>	Résultats de l'estimation de contraste (1 <sup>ère</sup> expérience).	44
<b>Tableau 2.8</b>	Les fonctions de références (benchmark) de la compétition de CEC'2005	45
<b>Tableau 2.9</b>	Valeurs des erreurs des fonctions de référence de la compétition de CEC'2005 pour 25 exécutions de dBA (2 <sup>ème</sup> expérience)	46
<b>Tableau 2.10</b>	Comparaison de la moyenne d'erreur obtenue par dBA avec d'autres algorithmes (2 <sup>ème</sup> expérience).	47
<b>Tableau 2.11</b>	Résultats de comparaisons par paires (2 <sup>ème</sup> expérience)	48
<b>Tableau 2.12</b>	Classements des algorithmes selon les tests de Friedman, Friedman Aligné et Quade (2 <sup>ème</sup> expérience).	48
<b>Tableau 2.13</b>	Les valeurs de <i>p-value</i> entre dBA et les autres algorithmes selon les différentes procédures Post-hoc considérées avec un niveau de signification $\alpha = 0.05$ (2 <sup>ème</sup> expérience).	49
<b>Tableau 2.14</b>	Résultats de l'estimation de contraste (2 <sup>ème</sup> expérience).	50
<b>Tableau 2.15</b>	Comparaisons entre dBA et les différentes variantes de BA amélioré (3 <sup>ème</sup> expérience).	51
<b>Tableau 2.16</b>	Comparaison entre dBA et CBSO (3 <sup>ème</sup> expérience).	52
<b>Tableau 3.1</b>	Paramètre des algorithmes considérés.	59
<b>Tableau 3.2</b>	Les résultats statistiques d'analyse de fiabilité du problème de la poutre en béton armé.	61

<b>Tableau 3.3</b>	La comparaison des solutions obtenues avec différentes configurations de la fonction objectif.	62
<b>Tableau 3.4</b>	Effet de la taille de la population de chauves-souris et le nombre d'itérations sur la précision de l'indice de fiabilité pour le problème de poutre en béton.	63
<b>Tableau 3.5</b>	Les variables aléatoires de l'exemple 2, structure conique.	64
<b>Tableau 3.6</b>	Les résultats statistiques de l'exemple 2 avec différentes tailles de population.	65
<b>Tableau 3.7</b>	Les résultats numériques pour les systèmes parallèles et série dans l'exemple 3.	68
<b>Tableau 3.8</b>	Les résultats numériques de l'exemple 4.	69
<b>Tableau 3.9</b>	Caractéristiques des variables aléatoires de l'exemple 5.	70
<b>Tableau 3.10</b>	Les résultats numériques de l'exemple 5.	71
<b>Tableau 4.1</b>	Caractéristiques des variables d'optimisation et des paramètres aléatoires de l'exemple 1.	87
<b>Tableau 4.2</b>	Comparaison entre dBA, BA, PSO et DE pour minimisation de l'Exemple 1.	88
<b>Tableau 4.3</b>	Effet du nombre des itérations et la taille de la population des chauves-souris sur la qualité des résultats d'optimisation de l'Exemple 1.	90
<b>Tableau 4.4</b>	Comparaison entre la meilleure, la médiane et la pire solution obtenues par dBA avec les méthodes classiques pour $\beta = 1.28$ (Exemple1).	92
<b>Tableau 4.5</b>	Comparaison entre la meilleure, la médiane et la pire solution obtenues par dBA avec les méthodes classiques pour $\beta = 3$ (Exemple1).	92
<b>Tableau 4.6</b>	Comparaison entre la meilleure, la médiane et la pire solution obtenues par dBA avec les méthodes classiques (Exemple2).	94
<b>Tableau 4.7</b>	Résultats d'optimisation de l'Exemple 2 pour différents indices de fiabilité et lois de probabilité.	95
<b>Tableau 4.8</b>	Caractéristiques des variables d'optimisation et paramètres de l'Exemple 3.	97
<b>Tableau 4.9</b>	Résultats d'optimisation de l'Exemple 3.	97
<b>Tableau 4.10</b>	Valeurs des contraintes, indice de fiabilité et la probabilité de défaillance pour chaque contrainte de la meilleure solution obtenue par dBA de l'Exemple 3.	98
<b>Tableau 4.11</b>	Caractéristique des variables et paramètres d'optimisation de l'Exemple 4.	100
<b>Tableau 4.12</b>	Résultats d'optimisation de l'Exemple 4 avec variables continues et paramètres déterministes.	101

<b>Tableau 4.13</b>	Résultats d'optimisation de l'Exemple 4 avec variables discrètes et paramètres aléatoires.	101
<b>Tableau 4.14</b>	Valeurs des contraintes, indice de fiabilité et probabilité de défaillance des contraintes de l'Exemple 4 avec variables discrètes et paramètres aléatoires.	103
<b>Tableau 5.1</b>	Les classes d'utilisation du pont roulant.	110
<b>Tableau 5.2</b>	Etat de chargement du pont roulant.	110
<b>Tableau 5.3</b>	Classification des ponts roulants.	110
<b>Tableau 5.4</b>	Paramètres communs des ponts roulants considérés.	119
<b>Tableau 5.5</b>	Paramètres et classes des ponts roulants considérés.	119
<b>Tableau 5.6</b>	Les résultants statistiques de l'optimisation pour différentes paires $(N, t_{max})$ .	122
<b>Tableau 5.7</b>	Taux de succès pour différentes paires $(N, t_{max})$ .	112
<b>Tableau 5.8</b>	Résultats d'optimisation de la poutre principale pour différentes classes et nuances d'acier.	123
<b>Tableau 5.9</b>	Caractéristiques des variables aléatoires et déterministes.	127
<b>Tableau 5.10</b>	Les résultats statistiques de l'indice de fiabilité pour 25 exécutions de l'algorithme dBA.	128
<b>Tableau 5.11</b>	Résultats d'approximation par FORM de la probabilité de défaillance.	132
<b>Tableau 5.12</b>	Comparaisons entre les probabilités de défaillances obtenues par dBA, SORM et MCS.	133
<b>Tableau 5.13</b>	Résultats d'optimisation basée sur la fiabilité.	135
<b>Tableau 5.14</b>	Comparaison de l'indice de fiabilité pour chaque équation d'état limite.	136
<b>Tableau 5.15</b>	Probabilité de défaillance des conceptions obtenues par l'RBDO.	137
<b>Tableau 5.16</b>	Valeurs déterministes des critères d'optimisation pour deux valeurs du facteur de sécurité.	137
<b>Tableau 6.1</b>	Position des obstacles.	146

# Introduction générale



*L'incertitude n'est pas dans les choses mais dans notre tête : l'incertitude est une méconnaissance. (Jacques Berboulli, 1654-1705)*

Le dimensionnement des structures est traditionnellement réalisé à partir des calculs déterministes. Dans la vie réelle, ces structures sont soumises à des fluctuations de chargement, la variabilité des propriétés mécaniques des matériaux, les incertitudes de mesure, l'erreur humaine et les imperfections des modèles d'analyse, ...etc. Ces dernières contribuent toutes, plus ou moins, à ce qu'une structure ait une résistance imprévisible. La mesure de la contribution de ces fluctuations sur la résistance d'une structure a conduit les chercheurs à définir un terme plus général qu'est la probabilité de défaillance. La « fiabilité » qui peut être définie comme le complémentaire de la probabilité de rupture, est un moyen de mesurer le degré de sécurité de l'ouvrage.

Mesurer la sécurité d'un ouvrage par sa fiabilité fait de la fiabilité un paramètre de décision utile. La fiabilité des systèmes complexes est un défi majeur pour les entreprises industrielles. Satisfaire à des exigences de fiabilité est alors nécessaire pour assurer un degré de sécurité suffisant au niveau de la conception des ouvrages. Ces derniers doivent répondre aux exigences des donneurs d'ordre dont le non-respect entraînerait des pénalités compromettant les marchés futurs. Pour répondre à de telles exigences et sans perdre la compétitivité dans le marché, les entreprises font recours à l'optimisation basée sur la fiabilité afin de minimiser les coûts de réalisation ou de fabrication tout en respectant un certain degré de fiabilité.

L'optimisation dans les applications du monde réel implique des problèmes fortement non linéaires avec de nombreuses variables et des contraintes complexes. L'objectif d'un problème d'optimisation peut généralement être associé à la minimisation des déchets, les coûts, les délais, ou la maximisation des avantages, du rendement et des bénéfices. En raison du fait que les méthodes ou les algorithmes analytiques traditionnels ne peuvent pas résoudre un grand nombre de problèmes pratiques, et surtout quand la fonction objectif est multimodale avec de nombreux optima locaux, les scientifiques se sont tournés vers la nature pour rechercher de nouvelles idées et inspiration pour

trouver des nouvelles méthodes de résolution. En essayant d'imiter les processus naturels, une large classe d'algorithmes appelés bio-inspirés ou algorithmes métaheuristiques a été introduite. Due à leur simplicité et leur robustesse à fournir des solutions, les métaheuristiques ont été largement utilisées dans diverses applications industrielles.

Due à leur importance, il est rare de trouver un hangar industriel sans un pont roulant. Les ponts roulants et les portiques sont des machines de manutention qui permettent, au moyen des mouvements élémentaires, de déplacer des charges sur une surface donnée. Selon les besoins, les ponts roulants sont conçus pour soulever de charges allant de quelques tonnes à des milliers de tonnes (les ponts roulants des chantiers de construction navale). Concevoir des ponts légers et fiables peut réduire le coût de fabrication ainsi que les coûts de fonctionnement à long terme.

Dans cette thèse, cinq grands axes sont abordés. Dans le premier axe, nous développons un nouvel algorithme métaheuristique appelé algorithme des chauves-souris directionnelles (directional Bat Algorithm, dBA). Cet algorithme a été construit en introduisant le phénomène d'écholocation directionnelle à l'algorithme des chauves-souris standard. Dans le second axe, nous proposons une nouvelle méthode d'estimation de la fiabilité basée sur l'algorithme des chauves-souris directionnelles. Dans la troisième partie, nous adaptons dBA de sorte qu'il puisse réaliser l'optimisation basée sur la fiabilité des structures. Dans la quatrième partie, nous considérons la conception basée sur la fiabilité de la poutre principale d'un pont roulant à bipoutres avec une automatisation pour rendre le mouvement du pont roulant autonome. Et enfin, dans la dernière partie, nous proposons une méthode pour l'autonomie du mouvement du pont roulant avec évitement d'obstacle en utilisant l'algorithme dBA. Pour ce faire, cette thèse est architecturée comme suit :

- Le premier chapitre introduit une synthèse sur les méthodes de fiabilité ainsi que les aspects théoriques fondamentaux de l'analyse de fiabilité. Une description des méthodes classiques d'estimation de la probabilité de défaillance est présentée (FORM, SORM et SMC).
- Le deuxième chapitre contient une présentation du nouvel algorithme métaheuristique introduit dans ce travail, à savoir l'algorithme des chauves-souris directionnelles. Ce dernier est conçu à base de l'algorithme des chauves-souris standard en introduisant quatre modifications telles que le phénomène d'écholocation directionnelle, pour améliorer l'efficacité et la robustesse de la résolution.

- Dans le troisième chapitre, nous présentons une nouvelle méthode d'estimation de la fiabilité des structures par l'algorithme des chauves-souris directionnelles. Un nouveau coefficient de pénalité adaptatif destiné à la résolution des problèmes d'analyse de fiabilité est introduit afin d'améliorer la certitude des résultats et l'efficacité de l'algorithme. Nous introduisons cette méthode en raison de sa simplicité de mise en œuvre et sa capacité à résoudre des problèmes de fiabilité structurelle complexe.
- Dans le quatrième chapitre, nous adaptons l'algorithme des chauves-souris directionnelles pour qu'il puisse résoudre des problèmes d'optimisation basée sur la fiabilité, en utilisant le concept de l'espace fiable et le niveau de comparaison  $\epsilon$ . A l'instar des méthodes classiques, cet algorithme est conçu pour répondre à des problèmes d'optimisation fiabiliste plus généraux.
- Dans le cinquième chapitre, l'analyse de fiabilité et l'optimisation de la poutre principale d'un pont roulant à bipoutres par les méthodes discutées dans les chapitres précédents sont développées
- Dans le dernier chapitre, une proposition sur l'automatisation d'un pont roulant est présentée. L'idée de base consiste à utiliser dBA pour rendre le mouvement du pont roulant autonome et sans risque de collision.
- Nous terminons par les conclusions et perspectives.

# **Chapitre 1**

# **Etat de l'art sur la fiabilité des structures**

*La connaissance progresse en intégrant en elle l'incertitude, non en l'exorcisant.  
(Edgar Morin, 1921)*

## 1.1 Introduction

Actuellement, les structures modernes exigent des conceptions plus critiques et complexes, le besoin d'approches précises et efficaces pour évaluer les incertitudes dans les charges, la géométrie, les propriétés des matériaux, les procédés de fabrication et des environnements opérationnels a augmenté de manière significative. L'application des techniques d'évaluation de la fiabilité contribuent à développer une orientation initiale pour la réalisation d'une conception fiable. Elles peuvent également être utilisées pour identifier et prendre en compte les incertitudes que se produisant dans les systèmes et structures [1].

L'estimation de la probabilité de défaillance d'une structure a été le principal sujet de plusieurs ouvrages sur la fiabilité des structures. Dans le cas d'une structure simple, la probabilité de défaillance peut être obtenue sous une forme fermée par intégration de la fonction de densité de probabilité conjointe (Joint Probability Density Function, JPDF) de toutes les variables aléatoires [2]. Dans la pratique, la détermination de la probabilité de défaillance par l'intégration multidimensionnelle analytique ou numérique est très difficile, par conséquent, plusieurs méthodes d'approximation ont été développées. Il existe essentiellement cinq catégories [3, 4], à savoir les méthodes de tirage [5, 6], les méthodes d'expansion [7], les méthodes basées sur la surface de réponse [8], les méthodes d'approximation par intégrale [9, 10], et les méthodes basées sur le point le plus probable (Most Probable Point, MPP) comme la méthode de fiabilité du premier ordre (First Order Reliability Method, FORM) [11, 12] et la méthode de fiabilité du second ordre (Second Order Reliability Method SORM) [13, 14].

L'idée de base des méthodes basées sur le MPP, tels que FORM, SORM ou d'autres comme la méthode univariée [15, 16], est d'employer le MPP comme le point de référence pour déterminer l'indice de fiabilité  $\beta$  et ensuite prédire la probabilité de défaillance. Dans l'espace normal

(Gaussien), l'indice de fiabilité  $\beta$  représente la distance minimale de l'origine à la surface de l'état limite. La évaluation de ce dernier, connu aussi sous le nom de l'indice de Hasofer-Lind, est un problème d'optimisation sous contrainte où la fonction objectif est la distance euclidienne  $\beta = \|U\|$ , sous la contrainte de  $G(U) = 0$  ( $G(\cdot)$  est la fonction d'état limite). La méthode la plus couramment utilisée pour résoudre ce problème d'optimisation est l'algorithme de Hasofer-Lind-Rackwitz-Fiessler (HLRF) [11, 12]. L'algorithme de HLRF a été développé pour être efficace, mais pas robuste, car il ne parvient pas à converger pour un nombre important de problèmes [17].

De nombreux travaux ont été réalisés pour améliorer cet algorithme. Liu et Kiureghian [18] ont présenté une version améliorée de l'algorithme de HLRF appelé M-HLRF pour surmonter les problèmes de convergence de l'algorithme de HLRF. L'idée de base est d'effectuer une recherche linéaire dans la direction de recherche générée par l'algorithme HLRF jusqu'à ce qu'une baisse suffisante de la fonction de mérite soit atteinte. Par la suite, Zhang et Kiureghian [19] ont présenté un algorithme amélioré (iHLRF) basé sur la recherche linéaire avec une nouvelle fonction de mérite. Santosh et *al.* [20] ont suggéré d'appliquer la règle de Armijo pour déterminer le pas. Santos et *al.* [21] ont proposé d'utiliser une nouvelle fonction de mérite combinée avec les conditions de Wolfe pour sélectionner la longueur du pas dans la recherche linéaire. Récemment, Yang[4] a présenté une nouvelle méthode pour remédier les problèmes d'instabilité numérique de l'algorithme HLRF, en se basant sur les méthodes de contrôle du chaos.

La méthode de fiabilité du deuxième ordre (SORM) a été établie pour améliorer la précision de FORM, en utilisant une approximation de deuxième ordre de l'équation d'état limite. La première approche de SORM a été proposée par Fiessler et *al.* [22] où ils ont utilisé la série de Taylor pour approximer la surface d'état limite au point le plus probable et les courbures de cette surface à ce point pour estimer la probabilité de défaillance. Breitung [13] a dérivé un résultat asymptotique exact de la parabole générée par la série de Taylor, puis, Tvedt [14] a proposé une formule à trois points pour approximer la probabilité de défaillance. Généralement, les formules de Breitung et Tvedt sont les plus utilisées dans la pratique. Ils existent d'autres formules de SORM dans la littérature [23-26], mais leur application reste faible.

Parmi les méthodes d'échantillonnage, la simulation de Monte Carlo (Monte Carlo Simulation, MCS) est l'une des plus puissantes et plus coûteuse en temps de calcul [27, 28]. Pour approximer la

probabilité de défaillance, un nombre important de tirage doit être réalisé pour couvrir l'espace de conception. Par conséquent, cette méthode consomme un énorme temps de calcul, spécialement si la probabilité de défaillance est très petite [1, 27-29]. Cependant, plusieurs méthodes basées sur les tirages ont été proposées pour réduire le temps de calcul, comme la méthode de tirage d'importance (Importance Sampling, IS)[30, 31], la méthode de tirage directionnel (Directional Sampling, DS) [32, 33], la méthode de tirage basée sur l'hypercube latin, (Latin Hypercube Sampling, LHS) [34, 35] et la simulation par sous-ensemble (Subset Simulation, SI) [36, 37].

Dans ce chapitre, nous présentons le fondement de la fiabilité des structures ainsi que les principales méthodes d'évaluation de la probabilité de défaillance. Dans la prochaine section, nous exposons l'idée principale de l'analyse de la fiabilité des structures. Dans la Section 1.3, la méthode de fiabilité du premier ordre (FORM) est expliquée ainsi que les principaux algorithmes d'estimation de l'indice de fiabilité. Dans la Section 1.4, nous introduisons la méthode de fiabilité du deuxième ordre (SORM) suivie de la simulation de Monte Carlo dans la section 1.5.

## **1.2 Analyse de la fiabilité des structures**

*L'analyse de fiabilité* consiste à évaluer la probabilité de défaillance d'une structure en déterminant si les fonctions d'états-limites sont dépassées ou non. Lorsqu'une structure (ou une partie de la structure) est soumise à des sollicitations qui dépassent une limite spécifique, la structure (ou une partie de la structure) est incapable d'exercer les besoins exigés. Cette limite spécifique est appelée *équation d'état limite*. La structure sera considérée comme défaillante et non-fiable, si la probabilité de défaillance dépasse la valeur requise. Pour la plupart des structures, l'équation d'état-limite peut être divisée en deux catégories :

*Les états limites ultimes* sont liés à un effondrement d'une partie ou la totalité d'une structure. La corrosion, la fatigue, la détérioration, le feu, la déformation plastique, l'effondrement progressif, la rupture ...etc., sont des exemples d'états limites ultimes. Un tel état-limite doit avoir une très faible probabilité d'occurrence, car il peut risquer la perte de la vie et des pertes financières importantes.

*Les états-limites de service* sont liés à la perturbation de l'utilisation normale des biens. Des exemples d'états-limites de service sont la déviation excessive, les vibrations excessives, la fuite, les dommages locaux, ...etc. Les états-limites de service sont moins dangereux que dans le cas des états limites ultimes, se qu'implique la possibilité de tolérer une probabilité d'occurrence plus grande.

Cependant, les gens ne peuvent pas utiliser les structures qui produisent trop de déflexion ou de vibrations, ...etc.

Généralement, l'état-limite indique la marge de sécurité entre la résistance d'une structure et les sollicitations appliquées. La fonction d'état-limite,  $g(\cdot)$ , et la probabilité de défaillance,  $P_f$ , peuvent être définies comme :

$$g(X) = R(X) - S(X) \quad (1-1)$$

$$P_f = P[g(X) < 0] \quad (1-2)$$

tel que  $R$  est la résistance et  $S$  représente les sollicitations et les charges appliquées au système. Les deux  $R(\cdot)$  et  $S(\cdot)$  sont des fonctions de la variable aléatoire  $X = (X_1, X_2, \dots, X_n)^T$ . La notation  $g(\cdot) < 0$  désigne *la région de défaillance*. De même,  $g(\cdot) = 0$  et  $g(\cdot) > 0$  indiquent *la surface de rupture* et de *la région de sécurité*, respectivement. Cependant, la probabilité de défaillance est évaluée comme suit :

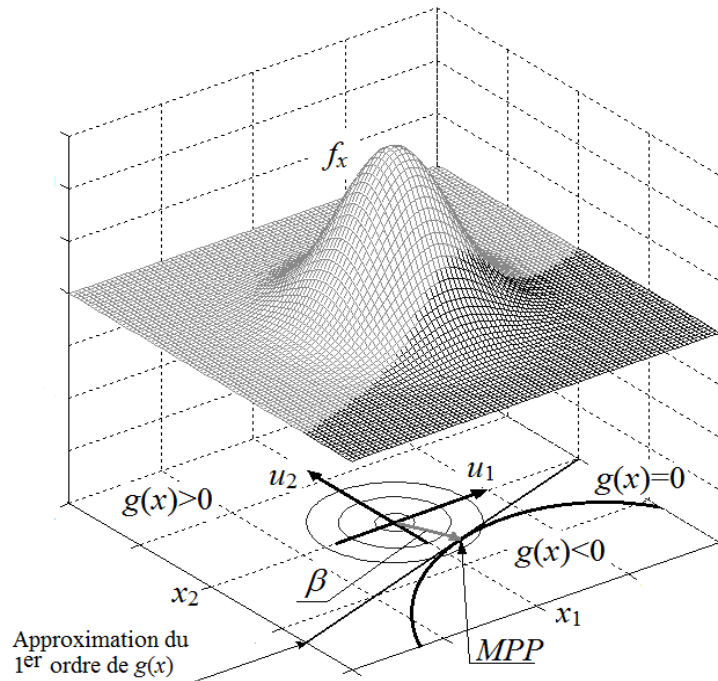
$$P[g(X) \leq 0] = \int_{g(X) \leq 0} f_x(x) dx \quad (1-3)$$

où  $f_x$  est la fonction de densité de probabilité conjointe (Joint Probability Density Function, JPDF). L'évaluation de l'intégrale de l'Eq. (1-3), que ce soit analytiquement ou numériquement, est très difficile à réaliser. Pour résoudre ce problème, les chercheurs ont proposé plusieurs méthodes d'approximation de la probabilité de défaillance (FORM, SORM, MCS...).

### 1.3 Méthode de fiabilité du premier ordre (FORM)

L'idée principale de FORM consiste à utiliser la série de Taylor pour créer une approximation linéaire de première ordre (d'où le nom premier ordre) de l'équation d'état limite au point le plus probable (Most Probable Point, MPP), qui présente le point de la surface d'état limite ( $g(x) = 0$ ) dont la probabilité d'occurrence est la plus haute des autres points de la surface.





**Fig. 1.1** Schéma hypothétique de la méthode FORM.

Considérons la Fig. 1.1, elle représente un schéma hypothétique du principe de FORM. Comme il est indiqué dans la figure, la méthode de premier ordre linéarise l'équation d'état limite au MPP, en utilisant la série de Taylor. En effet :

$$\tilde{g}(X) \approx g(\mu^*) + \nabla g(\mu^*)^T (X - \mu^*) \quad (1-4)$$

où  $\mu^*$  est le vecteur des moyennes du MPP.

Dans la partie haute de la Fig. 1.1, la surface  $f_x$  représente la fonction de densité de probabilité conjointe. Selon le principe de FORM, la probabilité de défaillance est approximativement égale au volume de la partie en couleur foncée, tandis que la fiabilité est égale au volume de la partie claire.

Pour estimer ce volume mathématiquement (la probabilité de défaillance), dans un cas général avec des variables indépendantes et qui obéissent à la loi Gaussienne dans un espace à  $n$ -dimensions, FORM transforme les variables à leurs formes normalisées selon cette transformation :

$$u_i = \frac{x_i - \mu_i}{\sigma_i} \quad (1-5)$$

où  $\mu_i$  et  $\sigma_i$  représentent la moyenne et l'écart type de la variables  $x_i$ , respectivement.  $u_i$  est appelée variable normale dont la moyenne est nulle avec un écart-type égal à l'unité. En suite, un algorithme d'optimisation est utilisé pour trouver la distance minimale entre l'origine du repère et la surface d'état limite, en résolvant ce problème :

$$\begin{cases} \text{Minimizer } \beta = \sqrt{\sum_i^n u_i^2} \\ \text{Telque: } G(u) = 0 \end{cases} \quad (1-6)$$

où

$$G(u) = g(\sigma u + \mu) = g(x) \quad (1-7)$$

La solution de l'Eq. (1-6) est appelée le point le plus probable  $u^*$ , et  $\beta$  est l'indice de fiabilité.

La probabilité de défaillance est approximée comme suit :

$$P_f = \Phi(-\beta) \quad (1-8)$$

La recherche du MPP sur la surface d'état-limite est une étape clé dans la méthode FORM. Il existe de nombreux algorithmes disponibles qui peuvent résoudre ce problème, tels que l'algorithme SQP (Sequential Quadratic Programming) ou d'autres algorithmes itératifs. Dans[38], plusieurs méthodes d'optimisation sous contraintes ont été utilisées pour résoudre ce problème, tel que les méthodes primitives (les directions faisables, gradient, projection, gradient réduit), les méthodes de pénalité, les méthodes duales, et la méthode des multiplicateurs de Lagrange. En pratique, la méthode la plus couramment utilisée est l'algorithme HLRF et ces variantes [11, 12].

### **1.3.1 L'algorithme HLRF**

Pour résoudre le problème d'optimisation (Eq. (1-6)), Hasofer et Lind [11], Rackwitz et Fiessler [12] (d'où le nom HLRF) ont proposé une procédure itérative pour chercher le MPP en démarrant de l'origine :

$$u^{k+1} = \frac{1}{\|\nabla G(u^k)\|^2} [\nabla G(u^k)^T u^k - G(u^k)] \nabla G(u^k) \quad (1-9)$$

L'algorithme HLRF est très répandu dans la pratique due à son efficacité. Par contre, l'algorithme ne garantit pas la convergence, et en réalité, il échoue à résoudre un grand nombre de problèmes[18]. Pour contourner ce problème de convergence, Liu et Kiureghian [18] ont proposé l'algorithme M-HLRF en effectuant une recherche suivant cette direction :

$$d^k = \frac{1}{\|\nabla G(u^k)\|^2} \left[ \nabla G(u^k)^T u^k - G(u^k) \right] \nabla G(u^k) - u^k \quad (1-10)$$

jusqu'à ce qu'une baisse suffisante de la fonction de mérite,  $m(u)$ , soit atteinte. La fonction de mérite suivante a été considérée :

$$m(u) = \frac{1}{2} \left| u - \frac{\nabla G(u)^T u}{\|\nabla G(u)\|^2} \nabla G(u) \right|^2 + \frac{1}{2} c \cdot G(u)^2 \quad (1-11)$$

Bien que la robustesse a été améliorée, Liu et Kiureghian [18] ont reconnu que la convergence de la méthode ne peut pas être garantie, puisque la fonction de mérite  $m(u)$  peut avoir des minimas qui ne sont pas des solutions du problème d'origine (Eq. 1-6). De plus,  $d^k$  peut ne pas être une direction de descente pour la fonction de mérite (Eq. 1-11) dans certains cas.

### 1.3.2 L'algorithme iHLRF

Une amélioration de l'algorithme appelée iHLRF (improved HLRF) a été présentée par Zhang et Kiureghian [19] est également basée sur la recherche linéaire dans la direction de HLRF:

$$u^{k+1} = u^k + \lambda^k d^k \quad (1-12)$$

A noter que la formule récursive de l'algorithme de HLRF est obtenue quand une longueur complète du pas est considéré ( $\lambda^k = 1$ ). La fonction de mérite non-différentiable suivante est utilisée dans la recherche linéaire :

$$m(u) = \frac{1}{2} u^T u + c |G(u)|. \quad (1-13)$$

Un pas  $\lambda^k$  appropriée est obtenu en utilisant la méthode d'Armijo, telle que la première valeur doit satisfaire une décroissance suffisante de la fonction de mérite.

$$m(u^{k+1}) < m(u^k) + m_1 \lambda^k \nabla m(u^k)^T d^k \quad (1-14)$$

où

$$\nabla m(u^k) = u^k + c \cdot \text{sign}(G(u^k)) \nabla G(u^k) \quad (1-15)$$

et  $m_1 \in (0,1) \forall u \in R^n$ . Zhang et Kiureghian [19] ont montré que pour  $\forall u \in R^n$ , la direction  $d^k$ , donnée dans l'Eq. (1-10) est une direction de descente à  $m(u)$  donnée dans l'Eq. (1-13), aussi longtemps que la condition suivante est satisfaite :

$$c > \frac{\|u\|}{\|\nabla G(u)\|} \quad (1-16)$$

Afin de satisfaire la condition Eq. (1-16), Zhang et Kiureghian [19] ont établi les règles suivantes pour la mise à jour  $c^k$  :

$$\left\{ \begin{array}{l} \text{Si } |G(u^k)| \geq \xi \\ c^k = \eta \max \left\{ \frac{\|u^k\|}{\|\nabla G(u^k)\|}, \frac{1}{2} \frac{\|u^k + d^k\|^2}{|G(u^k)|} \right\} \\ \text{Sinon} \\ c^k = \eta \frac{\|u^k\|}{\|\nabla G(u^k)\|} \end{array} \right. \quad (1-17)$$

Les valeurs des paramètres  $\xi$  et  $\eta$  suggérées par Zhang et Kiureghian [19] sont  $10^{-3} |G(u^0)|$  et 2, respectivement.

L'algorithme iHLRF a été prouvé pour être plus efficace et plus fiable que les algorithmes HLRF et M-HLRF [19]. Cependant, un inconvénient important de l'algorithme est le fait que la fonction de mérite utilisée est non-différentiable. Afin de vérifier si une réduction suffisante de la fonction de mérite a été atteinte durant la recherche linéaire, il est nécessaire d'évaluer le gradient de la fonction de mérite (Eq. 1-13) qui n'est pas défini dans tous les points du domaine.

### 1.3.3 L'algorithme nHLRF

L'algorithme nHLRF (new HLRF) proposé par Santos et *al.* [21] a été construit en introduisant une fonction de mérite différentiable :

$$m(u) = \frac{1}{2} u^T u + \frac{c}{2} G(u)^2 \quad (1-18)$$

avec  $c > 0$ .

Cette équation a été démontrée qu'elle est différentiable, et que la direction de recherche  $d^k$  de l'algorithme HLRF est une direction de descente de la fonction de mérite du nHLRF (Eq. 1-18) [21].

Pour garantir la convergence, la première condition qui doit être imposée, celle qui existe déjà dans l'algorithme iHLFR, est  $\lambda$  doit satisfaire la condition exprimée dans l'Eq. (1-14). Cependant, cette condition peut réduire les performances de l'algorithme dues aux faibles longueurs de pas générées. Pour soulever cette lacune, une condition de courbure est imposée :

$$\{m(u^k + \lambda d^k) < m_2 \nabla m(u^k)^T d^k\} \quad (1-19)$$

pour  $m_2 \in ]m_1, 1]$ , et  $m_1$  est donné dans l'Eq. (1-14).

La condition de descente suffisante de l'Eq. (1-14) et de courbure de l'Eq. (1-19), toutes les deux forment ce qu'on appelle les conditions de Wolf. L'algorithme nHLRF est présenté dans l'Algorithme 1.1.

### Algorithme 1.1

L'algorithme nHLRF.

- 
1. Introduire  $u^0 \in R^n$ ,  $\eta > 0$ ,  $\lambda \in ]0, 1]$ ,  $0 < m_1 < m_2 < 1$
  2. Initialiser  $k = 0$
  3. *Tant que* les conditions d'arrêt ne sont pas satisfaites
  4.     Calculer la direction de recherche  $d^k$  (Eq. 1-10)
  5.     Déterminer  $c^k$
  6.     *Si*  $G(u^k) = 0$
  7.         Choisir  $c^k > 0$
  8.     *Sinon*
  9.         
$$c^k = \eta \left| \frac{1}{G(u^k)} \cdot \frac{u^{kT} \nabla G(u^k)}{\|\nabla G(u^k)\|^2} \right|$$
  10.     *Fin de SI*
  11.     *Tant que*  $\{m(u^{k+1}) > m(u^k) + m_1 \lambda^k \nabla m(u^k)^T d^k\}$  et  $\{m(u^k + \lambda d^k)^T < m_2 \nabla m(u^k)^T d^k\}$
  13.         *Si*  $\{m(u^k + \lambda d^k)^T - m(u^k) > m_1 \lambda \nabla m(u^k)^T d^k\}$
  14.              $\lambda = 0.5 \lambda$
  15.         *Sinon Si*  $\{m(u^k + \lambda d^k) < m_2 \nabla m(u^k)^T d^k\}$
  16.              $\lambda = 0.2 \lambda$
  17.         *Fin de Si*
  18.          $\lambda^k = \lambda$
  19.          $u^{k+1} = u^k + \lambda^k d^k$
  20.     *Fin de Tant que*
  21.      $\beta = \|u^{k+1}\|$
  22.      $k = k+1$
  23.     *Fin de Tant que*
  24.     Traitement des résultats
- 

### 1.4 Méthode de fiabilité du second ordre (SORM)

En général, FORM est une fonction correcte lorsque la surface d'état limite à un seul minimum, et elle est approximativement linéaire au voisinage du point de conception considéré. Par contre, si la surface a de larges courbures (hautement non-linéaires), la probabilité de défaillance estimée par FORM en utilisant l'indice de fiabilité  $\beta$ , peut être inexacte [27]. Pour soulever ce problème, la méthode de fiabilité du second ordre utilise une approximation de deuxième ordre de  $g(x)$  dans l'espace normal comme la Fig. 1.2 le montre.

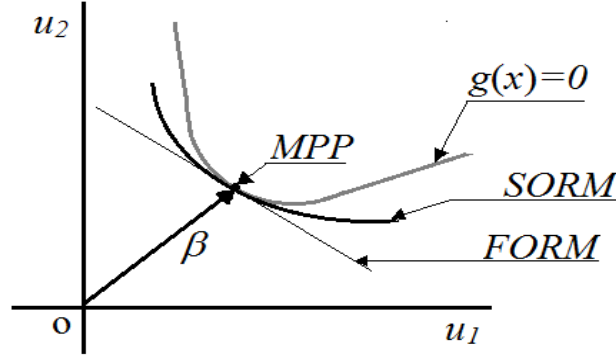


Fig. 1.2 Schéma hypothétique de la méthode SORM.

Dans l'espace normal, SORM approxime la fonction d'état limite au point le plus probable (MPP) comme suit :

$$G(U) = G(u^*) + \nabla G(u^*)^T (U - u^*) + \frac{1}{2} (U - u^*)^T \nabla^2 G(u^*) (U - u^*) \quad (1-20)$$

où  $u^*$  est le MPP, et  $\nabla^2 G(u^*)$  représente une matrice symétrique des dérivées secondes de la fonction limite :

$$\nabla^2 G(u^*)_{ij} = \frac{\partial^2 G(u^*)}{\partial u_i \partial u_j} \quad (1-21)$$

Cependant, la méthode SORM nécessite la connaissance du MPP ainsi que l'indice de fiabilité et les courbures ( $a_i$ ) de la surface d'état-limite au point le plus probable. En général, l'algorithme HLRF et ces variantes sont utilisés pour la recherche du MPP. Ils existent plusieurs approximations du second ordre de la probabilité de défaillance, dont les plus connues sont celles de Breitung [13] et Tvedt [14].

#### 1.4.1 Approximation de Breitung

Breitung a proposé une formule d'approximation de la probabilité de défaillance en introduisant les courbures de la surface d'état limite au point le plus probable :

$$P_f \approx \Phi(-\beta) \prod_i^{n-1} (1 + a_i \beta)^{-1/2} \quad (1-22)$$

où  $a_i$  représente les valeurs des courbures de la surface d'état limite au MPP.

### 1.4.2 Approximation de Tvedt

L'approximation de la probabilité de défaillance par la formule de Tvedt est :

$$A_1 = \Phi(-\beta) \prod_i^{n-1} (1 + a_i \beta)^{-1/2} \quad (1-21)$$

$$A_2 = [\beta \Phi(-\beta) - \phi(\beta)] \left\{ \prod_i^{n-1} (1 + a_i \beta)^{-1/2} - \prod_i^{n-1} (1 + a_i (\beta + 1))^{-1/2} \right\} \quad (1-22)$$

$$A_3 = (\beta + 1) [\beta \Phi(-\beta) - \phi(\beta)] \left\{ \prod_i^{n-1} (1 + a_i \beta)^{-1/2} \right\} - \operatorname{Re} \left( \prod_i^{n-1} (1 + a_i (\beta + 1))^{-1/2} \right) \quad (1-23)$$

$$P_f \approx A_1 + A_2 + A_3 \quad (1-24)$$

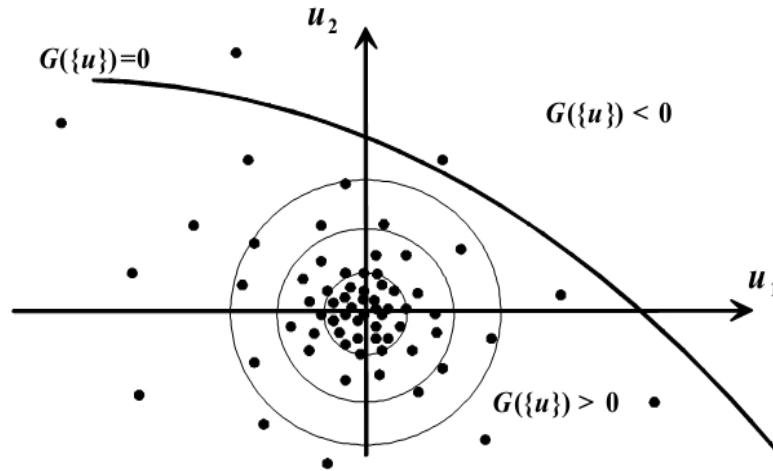
où  $\operatorname{Re}(\cdot)$  représente la partie réelle tandis que le premier terme  $A_1$  représente l'approximation de Breitung. Selon [1], cette approximation donne de bons résultats dans certains cas.

## 1.5 Méthode de Simulation de Monte Carlo

La méthode de Monte-Carlo est particulièrement utilisée pour évaluer des intégrales en dimensions plus grandes que 1 (en particulier, pour calculer des surfaces et des volumes). C'est le moyen le plus coûteux, mais certainement le plus robuste pour estimer la probabilité de défaillance, surtout pour des comportements complexes. Il est à noter que les simulations ne fournissent aucune information supplémentaire sur le point de défaillance et les facteurs d'importance des variables aléatoires (les cosinus directeurs). Pour chaque tirage, un calcul mécanique permet l'évaluation de la fonction d'état limite. Un indicateur de défaillance est activé si le jeu de données appartient au domaine de défaillance. Une estimation de la probabilité est ainsi donnée en fonction des défaillances observées.

L'évaluation de l'intégrale est directement effectuée au prix d'un certain nombre d'appels à la fonction d'état limite (i.e. appels mécaniques). Nous développons ici la méthode la plus générale et la plus coûteuse, dite *Simulations de Monte Carlo classique* (Monte Carlo Simulation, MCS). Les tirages sont effectués dans tout l'espace suivant la loi de distribution de chaque variable. Pour un  $N$  tirages aléatoires, l'espérance de l'intégrale est évaluée par le ratio des échantillons défaillants sur le nombre total de tirages.





**Fig. 1.3** Principe de la simulation de Monte Carlo.

Pour obtenir la probabilité de défaillance, nous introduisons un indicateur de défaillance à l'Eq. (1-3). Cette intégrale peut être écrite sous la forme :

$$P_f = \int_{g(x) \leq 0} I f_x(x) dx = E[I] \quad (1-25)$$

ou  $E[.]$  est l'espérance mathématique et  $I$  est l'indicateur de défaillance donné par :

$$I = \begin{cases} 1 & \text{si } g(x) \leq 0 \\ 0 & \text{si } g(x) > 0 \end{cases} \quad (1-26)$$

Pour  $N$  tirages aléatoires, la moyenne empirique de  $I$  est une estimation non biaisée de  $P_f$ :

$$P_f \approx \frac{1}{N} \sum_{i=1}^N I_i \quad (1-27)$$

dont la variance est :

$$\sigma_I^2 \approx \frac{1}{N^2} \left[ \sum_{i=1}^N I_i^2 - \left( \sum_{i=1}^N I_i \right)^2 \right] \quad (1-28)$$

L'erreur commise dans l'évaluation de la probabilité de défaillance est donnée par :

$$err = 200 \sqrt{\frac{1-P_f}{N P_f}} \quad (1-29)$$

D'une manière générale, pour évaluer correctement une probabilité de l'ordre de  $10^{-n}$ , il faut effectuer de  $10^{n+2}$  à  $10^{n+3}$  tirages (i.e. évaluation de la fonction d'état limite). Selon la formule ci-dessus, l'erreur est de 20% pour  $10^{n+2}$  tirages et de 6.3% pour  $10^{n+3}$  tirages. Il est évident que cette méthode est impossible à utiliser pour les grands systèmes à faible probabilité de défaillance.

## **1.6 Epilogue**

Les méthodes d'approximation analytique de la probabilité de défaillance (FORM et SORM), ne nécessitent pas un nombre élevé d'évaluations de l'équation d'état limite. Cependant, elles ne permettent pas de contrôler l'erreur commise sur les résultats. Cette erreur peut s'amplifier lorsque l'équation d'état limite est fortement non-linéaire ou elle contient plusieurs points optimaux. La simulation de Monte Carlo peut obtenir des résultats très fiables, mais elle est très gourmande en nombre d'évaluations de l'équation d'état limite, ce qui augmente significativement le temps de calcul. Ils existent d'autres méthodes de tirage moins coûteuses en temps de calcul telles que le tirage d'importance et la simulation par sous-ensemble, mais lorsque la valeur de la probabilité de défaillance est très faible, son estimation devient très coûteuse par ces dernières.

# **Chapitre 2**

# **Algorithme des chauves-souris**

# **directionnelles**

*Il n'y a rien de plus beau qu'une clef tant qu'on ne sait pas ce qu'elle ouvre (Maurice Maeterlinck, 1862-1949)*

## 2.1 Introduction

L'optimisation dans les applications du monde réel implique des problèmes complexes fortement non linéaires avec de nombreuses variables et des contraintes complexes. L'objectif d'un problème d'optimisation peut généralement être associé à la minimisation des déchets, les coûts, les délais, ou la maximisation des avantages, du rendement et des bénéfices. En raison du fait que les méthodes ou algorithmes analytiques traditionnels ne peuvent pas résoudre un grand nombre de problèmes pratiques, et surtout quand la fonction objectif est multimodale avec de nombreux optima locaux, les scientifiques se sont tournés vers la nature pour rechercher de nouvelles idées et d'inspiration pour trouver des nouvelles méthodes de résolution.

Depuis, plusieurs algorithmes ont été inspirés des différents processus naturels. L'algorithme génétique (Genetic Algorithm, GA) [39] et l'algorithme d'évolution du différentiel (Differential Evolution, DE) [40] sont inspirés du processus de l'évolution biologique. L'algorithme d'optimisation par essaim de particules (Particle Swarm Optimization, PSO) [41, 42] et l'algorithme des colonies de fourmis (Ants Colony Optimization, ACO) [43] sont inspirés du comportement des essaims. L'algorithme de recherche d'harmonie (Harmony Search, HS) [44] a été inspiré du processus de composition de musique des musiciens. Ces algorithmes métaheuristiques appelés également les algorithmes inspirés de la nature ou des algorithmes bio-inspirés sont devenus très populaires en raison de leur structure simple et leur capacité à obtenir une solution là où les algorithmes analytiques traditionnels échouent.

Les algorithmes issus de l'intelligence des essaims sont basés sur le comportement social et le travail collaboratif des agents de l'essaim tels que les fourmis, les oiseaux, les poissons, les insectes et les chauves-souris. Par exemple, l'ACO est basé sur le comportement alimentaire des fourmis [43], tandis que l'algorithme d'optimisation par essaim de chat (Cat Swarm Optimisation, CSO) [45]

imité le comportement des chats quand ils traquent et chassent leurs proies. L'algorithme de recherche des coucous (Cuckoo Search, CS) introduit par Yang et Deb [46] a été inspiré du comportement de reproduction parasitaire de coucous, par contre, l'algorithme des lucioles (Firefly Algorithm, FA) a été fondé à partir des motifs de lumière clignotants créés par les lucioles tropicales dans la saison de reproduction [47].

L'algorithme des chauves-souris (Bat Algorithm, BA) proposé par Xin-She Yang [48] est également un algorithme basé sur l'intelligence des essaims. Il se base sur le processus d'écholocation des micro-chauves-souris. En vol, les chauves-souris émettent des impulsions d'ultrasons à l'environnement. Les ondes sonores voyagent à travers l'air jusqu'à ce qu'ils atteignent une proie ou un objet puis rebondit comme un écho. Des études montrent que les informations extraites des échos permettent aux chauves-souris de construire une image précise de leur environnement et de déterminer avec précision les distances, les formes et l'emplacement des proies. La capacité de l'écholocation des micro-chauves-souris est fascinante car ces dernières peuvent trouver leur proies et discriminer différents types d'insectes, même dans l'obscurité totale [48]. Des études antérieures ont montré que BA peut résoudre des problèmes d'optimisation sous contrainte et sans contrainte avec beaucoup plus d'efficacité et de robustesse par rapport à GA et PSO [48-50].

Bien que l'algorithme des chauves-souris ait été proposé récemment (2010), il existe déjà plusieurs variantes de BA dans la littérature [51-67]. Malgré le fait que BA est un algorithme très puissant et peut produire des solutions robustes à des problèmes de faibles dimensions, son rendement diminue de façon significative lorsque la dimension du problème augmente, ce qui est également vrai pour d'autres algorithmes tels que PSO et GA. Des études ont montré que BA peut être très efficace dans la résolution d'une large gamme de problèmes [51-67], et pourtant, le problème de convergence prématurée peut se produire sous certaines conditions. Comme tous les algorithmes basés sur l'intelligence des essaims, le mécanisme de recherche de solution d'un algorithme est régi par deux éléments essentiels : l'exploitation et l'exploration. Brièvement parlant, l'exploration est la capacité d'un algorithme à explorer l'espace de recherche et balayer de nouvelles régions inconnues, tandis que l'exploitation est la capacité d'améliorer les solutions en exploitant les informations obtenues à partir de solutions existantes.

Dans ce chapitre, nous fixons notre objectif qui consiste à améliorer les performances de l'algorithme des chauves-souris standard en augmentant ses capacités d'exploitation et d'exploration.

De nouvelles modifications et règles idéalisées ont été intégrées au BA standard en utilisant le processus d'écholocation directionnelle et d'autres caractéristiques. Le nouvel algorithme de chauve-souris directionnelle proposé (directional Bat Algorithm, dBA) sera testé sur plusieurs benchmarks et comparé avec d'autres algorithmes. Dans la section qui suit, l'algorithme des chauves-souris standard est présenté, suivi par une enquête sur les différentes variantes de BA qui existent dans la littérature. Le nouvel algorithme des chauves-souris directionnelles (dBA) est décrit en détail dans la Section 2.4. Les résultats de simulation sont présentés dans la Section 2.5, et on termine par les discussions et les conclusions à la Section 2.6.

## 2.2 L'algorithme des chauves-souris standard

L'algorithme des chauves-souris a été inspiré du processus d'écholocation des chauves-souris. En observant le comportement et les caractéristiques des micro-chauves-souris, Yang [48] a proposé le BA standard en accordance aux trois caractéristiques majeures du processus d'écholocation des micro-chauves-souris. Les règles idéalisées utilisées dans BA sont :

- a) Toutes les chauves-souris utilisent l'écholocation pour détecter la distance et elles savent aussi la différence entre les aliments (proies) et les obstacles d'une certaine façon magique [48].
- b) Les chauves-souris volent au hasard avec une vitesse  $v_i$  à la position  $x_i$  tout en variant la fréquence  $f_i$  des ultrasons émises (à partir d'un minimum  $f_{min}$  à une fréquence maximale  $f_{max}$ ) ou la longueur d'onde  $\lambda$  et l'intensité  $A$  lors de la recherche d'une proie. Elles peuvent ajuster automatiquement la longueur d'onde (ou fréquence) des impulsions émises et le taux d'émission d'impulsions  $r$  en fonction de la proximité de la cible [48].
- c) L'intensité sonore varie d'une grande valeur positive  $A_0$  à une valeur minimale constante  $A_{min}$  [48].

Pour chaque chauve-souris ( $i$ ), sa position ( $x_i$ ) et sa vitesse ( $v_i$ ) dans un espace de recherche à  $d$ -dimensions doivent être définies.  $x_i$  et  $v_i$  devraient ensuite être mises à jour au cours des itérations. Les règles de mise à jour de la position et de la vitesse d'une chauve-souris virtuelle ( $i$ ) sont données comme suit [48] :

$$f_i = f_{min} + (f_{max} - f_{min})rand \quad (2-1)$$

$$v_i^{t+1} = v_i^t + (x^* - x_i^t) f_i \quad (2-2)$$

$$x_i^{t+1} = x_i^t + v_i^{t+1} \quad (2-3)$$

où  $rand \in [0,1]$  est un vecteur à valeurs aléatoires tiré d'une distribution uniforme. Ici  $x^*$  représente la meilleure position (solution) qui est obtenue après la comparaison de toutes les solutions de chaque chauve-souris. Une nouvelle solution est générée localement (recherche locale) en utilisant la marche aléatoire donnée par :

$$x_{new} = x_{old} + \varepsilon \langle A_i^{t+1} \rangle \quad (2-4)$$

où  $\varepsilon \in [-1,1]$  est un nombre aléatoire, tandis que  $\langle A_i^{t+1} \rangle$  est l'intensité sonore moyenne de toutes les chauves-souris à cette étape d'itération.

L'intensité  $A_i$  et le taux d'impulsions d'émission  $r_i$  sont mis à jour lors du processus d'itération. L'intensité diminue et le taux des impulsions augmentent lorsque la chauve-souris se rapproche de sa proie. Les équations de mise à jour de ces derniers sont :

$$A_i^{t+1} = \alpha A_i^t, \quad (2-5)$$

$$r_i^{t+1} = r_i^0 [1 - \exp(-\gamma t)] \quad (2-6)$$

Où  $0 < \alpha < 1$  et  $\gamma > 0$  sont des constantes. Lorsque  $t \rightarrow \infty$ , on a  $A_i^t \rightarrow 0$  and  $r_i^t \rightarrow r_i^0$ . L'intensité  $A_0$  peut typiquement être  $A_0 \in [1, 2]$ , tandis que le taux d'impulsion initiale  $r^0 \in [0, 1]$ .

Les étapes de base de l'algorithme des chauves-souris standard sont résumées dans le pseudo code comme indiqué dans l'algorithme 2.1.

### Algorithme 2.1

Algorithme des chauves-souris standard.

---

1. Définir la fonction objectif
  2. Initialiser la population des chauves-souris  $L_i \leq (x_i, v_i) \leq U_i (i=1,2,..,n)$
  3. Définir les fréquences  $f_i$
  4. Initialiser le taux d'impulsion  $r_i$  et l'intensité  $A_i$
  5. *Tant que* ( $t \leq t_{max}$ )
  6.     Ajuster les fréquences (Eq. 2-1)
  7.     Mise à jour de la vitesse (Eq. 2-2)
  8.     Mise à jour de la position/solution (Eq. 2-3)
  9.     *Si* ( $rand > r_i$ )
  10.         Sélectionner une solution parmi les meilleures solutions aléatoirement
  11.         Générer une solution locale autour de la solution sélectionnée (Eq. 2-4)
  12.     *Fin de Si*
  13.     Générer une nouvelle solution en volant aléatoirement
  14.     *Si* ( $rand < A_i \& F(x_i) < F(x^*)$ )
  15.         Accepter la nouvelle solution
  16.         Augmenter  $r_i$  (Eq. 2-5)
  17.         Diminuer  $A_i$  (Eq. 2-6)
  18.     *Fin de Si*
  19.     Classer les chauves-souris et trouver la nouvelle meilleure position  $x^*$
  20. *Fin de Tant que*
  21. Traitement des résultats
- 

### 2.3 Enquête sur les différentes variantes de BA existant dans la littérature

L'algorithme des chauves-souris standard est un outil d'optimisation très puissant et il peut produire des solutions robustes pour les problèmes à faibles dimensions [51-67], Toutefois, certaines études ont rapporté que sa performance peut diminuer à mesure que les dimensions du problème augmentent. Ces problèmes de convergence prématurée liés à la dimensionnalité sont également rencontrés dans presque tous les autres algorithmes, mais il existe des remèdes possibles qui seront explorés dans cette section. Pour pallier cette lacune, plusieurs méthodes et techniques ont été proposées. Dans [51], l'auteur a incorporé onze fonctions chaotiques pour trouver la meilleure variante de l'algorithme de chauve-souris chaotique (Chaotic Bat Swarm Optimisation CBSO). La meilleure stratégie trouvée est que l'équation de mise à jour de l'intensité ( $A_i$ ) soit une fonction linéairement décroissante multipliée par une carte chaotique itérative avec des fonctions d'effondrements infinies. D'après les résultats, l'auteur a rapporté que CBSO peut surpasser le BA, le CS, le GA l'algorithme de Big Bang- Big Crunch (BBBC) [68] et l'algorithme de recherche gravitationnelle (GSA) [69].



Mirjalili et *al.* [52] ont présenté une version binaire de BA nommée l'algorithme des chauves-souris binaire (Binary Bat Algorithm, BBA). Le BBA a un système de navigation et de chasse artificielle dans un espace de recherche binaire en changeant les positions des chauves-souris de "0" à "1" et vice-versa. Leurs résultats ont montré la supériorité de BBA par rapport au PSO binaire et l'algorithme génétique. He et *al.* [53] ont introduit à la fois l'algorithme du recuit simulé (Simulated Annealing, SA) [70] et la perturbation gaussienne dans le BA standard pour accélérer la vitesse de convergence globale (SAGBA). Une fois qu'une population initiale est générée, les meilleures solutions sont remplacées par de nouvelles solutions avec les équations de SA, puis mises à jour par les équations de BA standard. Ensuite, le processus de perturbation gaussienne est utilisé pour perturber les positions des chauves-souris pour générer un ensemble de nouvelles solutions. Les résultats montrent que SAGBA est plus performant que le BA standard.

Dans [54], les auteurs proposent un nouvel algorithme de chauve-souris en introduisant l'opérateur différentiel et les vols de Lévy (DLBA). Le DLBA a la même structure que BA, mais les équations de mouvement des chauves-souris ainsi que la recherche locale sont différentes. L'opérateur différentiel qui est similaire à l'opération de mutation de l'algorithme d'évolution différentielle (DE) a été introduit à la fréquence pour la faire fluctuer de haut en bas et change automatiquement. De plus, le vol de Lévy est intégré à l'équation de recherche locale pour permettre aux chauves-souris de sauter hors des optimums locaux.

En se basant sur le concept de communication des processeurs lors du calcul en parallèle, Nguyen et *al.* [55] ont proposé un algorithme hybride basé sur le BA standard et l'algorithme de la colonie d'abeilles artificielles (BA-ABC). Chaque algorithme évolue indépendamment, après avoir exécuté un nombre fixe d'itérations, les deux algorithmes échangent ensuite les données. Les chauves-souris avec des solutions proches de la meilleure solution remplacent les pires agents artificiels de l'ABC (Artificial Bee Colony). En revanche, les meilleurs agents artificiels de l'ABC remplacent les chauves-souris à mauvaise position de BA. En outre, les chauves-souris dans BA ne savent pas l'existence des abeilles artificielles de l'ABC et vice versa. Les résultats montrent que l'hybride BA-ABC augmente la précision et la convergence est meilleure que celle du BA (Jusqu'à 78%) et du ABC (Jusqu'à 11%). En utilisant la même idée, Pan et *al.* [56] ont hybridé PSO avec BA. Dao et *al.* [57] ont proposé une version compacte de l'algorithme des chauve-souris (CoBA) adressée aux matériels (ordinateur, micro-processeur) avec des ressources limitées telles que la taille de la

mémoire ou les équipements à bas prix. La population des chauves-souris est remplacée par un vecteur de probabilité mis à jour par un seul calcul. Ceci a conduit à un algorithme qui fonctionne avec un usage de mémoire modeste. Les résultats montrent que les performances CoBA sont aussi bonnes que celles de BA standard en dépit de son modeste usage de mémoire.

Dans [58], l'algorithme des chauves-souris adaptatif (ABA) est présenté. Chaque chauve-souris peut automatiquement ajuster sa vitesse et la direction du vol. Les auteurs ont introduit un facteur d'inertie à l'équation de la vitesse en fonction de la distance entre la position réelle de chauve-souris et la position de la meilleure chauve-souris. Ils ont également proposé d'ajouter un facteur rétrécissement à la formule de recherche locale pour réduire l'espace de recherche au cours du processus itératif. Avec une certaine similitude à l'ABA, l'algorithme de chauve-souris avec souvenir (Recollection Bat Algorithm, RBA) [59] a un coefficient d'inertie qui varie en fonction du temps imposé sur la vitesse afin que les chauve-souris avancent avec une finesse vers la chauve-souris à meilleure position. Chen et *al.* [60] ont introduit l'effet Doppler au BA standard pour améliorer l'efficacité de la résolution (DBA). L'effet Doppler produit un décalage de fréquence provoqué par la vitesse entre la source sonore et l'observateur. Ils ont proposé d'ajouter le concept de conscience pour permettre aux chauves-souris de détecter le changement dans l'environnement par des lois physiques selon l'effet Doppler.

L'algorithme de chauve-souris hybride (HBA) proposé par Fister et *al.* [61] a été adapté pour devenir l'algorithme des chauves-souris hybride auto-adaptatif (HSABA) [62]. L'algorithme a deux caractéristiques principales. Tout d'abord, les auteurs ont introduit une technique d'auto-adaptation aux paramètres de contrôle de BA (l'intensité sonore et le taux d'impulsion). Deuxièmement, ils ont remplacé l'équation de recherche locale par l'opérateur d'évolution différentielle où quatre stratégies des opérateurs ont été considérées, à savoir *DE/rand/1/bin*, *DE/randToBest/1/bin*, *DE/best/2/binet* enfin *DE/best/1/bin*. Les résultats ont montré que l'algorithme de chauve-souris hybride basé sur "*DE/best/2/bin*" a obtenu les meilleurs résultats par rapport aux trois autres stratégies. L'algorithme de chauve-souris amélioré (EnBA) proposé dans [63] a été développé à travers trois modifications différentes. Un facteur d'inertie a été proposé pour équilibrer les capacités de recherche pendant le processus d'optimisation en fonction de l'exigence de BA. Une nouvelle équation pour contrôler la vitesse des chauves-souris a été introduite avec une aptitude de contribuer à la dispersion des solutions dans l'espace de recherche. Enfin, une hybridation avec l'algorithme de l'herbe envahissant

(IWO) [71] a été adopté pour augmenter la capacité de l'exploitation plutôt que de l'exploration avec rectification vers la fin du processus d'optimisation.

L'algorithme de chauve-souris avec mutation auto-adaptative (BA-SAM) proposé par Alam et Kabir [64] utilise la mutation aléatoire de Gauss et de Cauchy à la place de l'équation de recherche locale d'origine. L'idée centrale de l'auto-adaptation est une sélection probabiliste d'un des deux régimes de mutation en utilisant une stratégie d'apprentissage. L'algorithme de chauve-souris modifié (MBA) proposé par Yilmaz et al [65] a le même organigramme que l'algorithme BA standard. Une nouvelle stratégie a été proposée pour la mise à jour des paramètres de contrôle (taux d'impulsion et l'intensité). De plus, une nouvelle équation a été proposée pour la recherche locale. Les résultats ont montré une amélioration significative dans la résolution des problèmes à faible dimension, mais pas autant pour des problèmes de large dimensionnalité. Les auteurs dans [66] ont proposé l'algorithme des chauves-souris évolué (EvBA) pour résoudre des problèmes d'optimisation numériques basés sur le cadre de BA. L'EvBA a été construit avec une nouvelle définition des équations du mouvement des chauves-souris où ils ont introduit la vitesse du son pour mesurer les distances et la mise à jour du mouvement des chauves-souris.

Dans [67], les auteurs ont utilisé un codage par des nombres complexes pour construire l'algorithme des chauves-souris à valeurs complexes (CVBA). L'idée de base du codage complexe est d'utiliser deux paramètres, une partie réelle et une partie imaginaire, pour représenter une variable. Ces deux parties (réelle et imaginaire) sont mises à jour en parallèle. Les variables indépendantes de la fonction objectif sont déterminées par les modules et les angles de leur nombre complexe correspondant. Par conséquent, la diversité de la population est considérablement améliorée.

Tous ces algorithmes ont été testés sur les fonctions de référence classiques. La majorité de ces fonctions ont l'optimum situé dans le centre de l'espace de recherche avec le vecteur nul ( $x = [0...0]^d$ ) comme solution, i.e. la fonction de Griewank, la fonction de Rastrigin, la fonction d'Ackley, etc. Certains d'entre eux ont montré une amélioration significative par rapport à l'algorithme de chauve-souris standard, mais aucun d'entre eux avait été testé sur des fonctions de référence non-standards, comme ceux conçus pour la compétition du Congrès international sur le calcul évolutif (CEC'2005 tests suite [72]) où les fonctions ont été transférées, tournées et déplacées afin d'assurer que leur optimum ne peut jamais être trouvé dans le centre de l'espace de recherche.

## 2.4 Le nouvel algorithme des chauves-souris directionnelles

Le nouvel algorithme de chauve-souris directionnelle a le même organigramme ou procédure que l'algorithme des chauves-souris standard. Pour renforcer les capacités d'exploitation et d'exploration de l'algorithme, nous introduisons quatre modifications dont le but est d'améliorer les performances de BA standard.

### 2.4.1 La 1<sup>ère</sup> modification

Au cours de leurs vols, les chauves-souris émettent des impulsions sonores en continu qui durent quelques millisecondes. En analysant l'écho, ils peuvent créer une image mentale en 3D de leur environnement. Nous supposons que chaque chauve-souris connaît la position des autres, et peut en quelque sorte savoir si la nourriture existe autour de chaque chauve-souris ou non. Nous supposons également que chaque chauve-souris émet deux impulsions dans deux directions différentes avant de décider dans quelle direction qu'elle volera. De plus, nous supposons également que toutes les chauves-souris émettent une impulsion dans le sens de la meilleure chauve-souris (solution) où la nourriture est considérée comme existante, et l'autre impulsion à la direction d'une chauve-souris choisie au hasard.

Comme montrée sur la Fig. 2.1, une chauve-souris émet deux impulsions dans deux directions différentes, une dans la direction de la chauve-souris à meilleure position (la meilleure solution), et l'autre à la direction d'une chauve-souris choisie au hasard. De l'écho (feedback), la chauve-souris peut savoir si la nourriture existe autour de ces deux chauves-souris ou non. Habituellement, autour de la chauve-souris avec la meilleure position, la nourriture existe (car elle a la meilleure valeur de la fonction objectif), mais autour de la chauve-souris choisie au hasard, cela dépend de la valeur de la fonction objectif, si elle a une valeur meilleure que celle de la chauve-souris actuelle, la nourriture est considérée comme existante, sinon il n'y a pas.

Si la nourriture a été confirmée d'exister autour des deux chauves-souris (cas 1), la chauve-souris actuelle se déplace à une direction au voisinage des deux chauves-souris où la nourriture est censée être abondante. Si non (cas 2), elle se déplace vers la meilleure chauve-souris. Les équations du mouvement des chauves-souris sont :

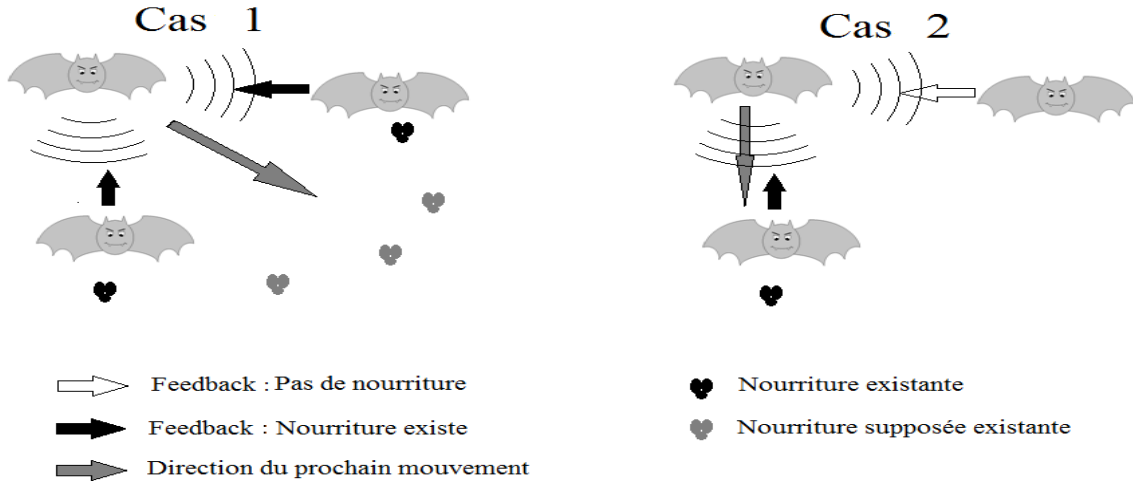


Fig. 2.1 Schématisme de l'écholocation directionnelle.

$$\begin{cases} x_i^{t+1} = x_i^t + (x^* - x_i^t)f_1 + (x_k^t - x_i^t)f_2 & (\text{Si } F(x_k^t) < F(x_i^t)) \\ x_i^{t+1} = x_i^t + (x^* - x_i^t)f_1 & \text{Sinon} \end{cases} \quad (2-7)$$

où  $x_k^t$  est la position (solution) de chauve-souris choisie au hasard ( $k \neq i$ ) et  $x^*$  est la meilleure solution.  $F(.)$  est la fonction objectif.  $f_1$  et  $f_2$  sont les fréquences des deux impulsions émises, et sont mises à jour comme suit :

$$\begin{cases} f_1 = f_{\min} + (f_{\max} - f_{\min})rand1 \\ f_2 = f_{\min} + (f_{\max} - f_{\min})rand2 \end{cases} \quad (2-8)$$

où  $rand1$  et  $rand2$  sont deux vecteurs aléatoires tirés d'une distribution uniforme entre 0 et 1.

La première partie de l'Eq. (2-7) augmente la capacité d'exploration de l'algorithme, tandis que la seconde partie renforce la capacité d'exploitation du nouvel algorithme des chauves-souris.

#### 2.4.1 La 2<sup>ème</sup> modification

La deuxième modification concerne la partie de la recherche locale. Les chauves-souris sont autorisées à se déplacer de leur position actuelle à une nouvelle position aléatoire avec l'équation suivante :

$$x_i^{t+1} = x_i^t + \langle A' \rangle \varepsilon w_i^t \quad (2-9)$$

où  $\langle A^t \rangle$  est l'intensité moyenne de toutes les chauves-souris et  $\varepsilon \in [-1,1]$  est un vecteur aléatoire.  $w_i$  est un paramètre utilisé pour réduire l'espace de recherche lors du processus itératif. Il démarre d'une grande valeur (environ un quart de la longueur de l'espace) et il diminue à environ 1% du quart de la longueur de l'espace. L'équation de sa mise à jour est la suivante :

$$w_i^t = \left( \frac{w_{i0} - w_{i\infty}}{1 - t_{\max}} \right) (t - t_{\max}) + w_{i\infty} \quad (2-10)$$

Ici,  $w_{i0}$  et  $w_{i\infty}$  sont les valeurs initiale et finale que  $w_i$  peut prendre encours du processus itératif. Nous définissons  $w_{i0}$  et  $w_{i\infty}$  comme suit :

$$w_{i0} = (U_i - L_i) / 4 \quad (2-11)$$

$$w_{i\infty} = w_{i0} / 100 \quad (2-12)$$

où  $t$  est l'itération actuelle, et  $t_{\max}$  est le nombre maximum des itérations.  $U_i$  et  $L_i$  sont les limites supérieure et inférieure.

Au début du processus itératif,  $w_i$  commence par une grande valeur. Il permet aux chauves-souris de se déplacer au hasard de manière à augmenter la capacité d'exploration de l'algorithme et donc être en mesure d'explorer la totalité de l'espace de recherche avec plus d'efficacité. A la fin du processus itératif, la valeur de  $w_i$  diminue, ce qui réduit l'espace de recherche autour de la meilleure solution. Ainsi, la capacité d'exploitation de l'algorithme est également améliorée.

### 2.4.3 La 3<sup>ème</sup> modification

Les Eqs. (2-5) et (2-6), proposées par Yang [48] pour mettre à jour le taux d'impulsion et l'intensité, atteignent leur valeurs finales pendant le processus itératif très rapidement, réduisant ainsi la possibilité de l'auto-commutation entre la marche aléatoire (Eqs. de mouvement des chauves-souris) et la recherche locale due à un taux d'impulsion très élevé, et l'acceptation d'une nouvelle solution (faible intensité). Par conséquent, nous proposons d'utiliser ces fonctions monotones croissante et décroissante pour la mise à jour du taux d'impulsion et l'intensité sonore :

$$r^t = \left( \frac{r_0 - r_\infty}{1 - t_{\max}} \right) (t - t_{\max}) + r_\infty \quad (2-13)$$

$$A^t = \left( \frac{A_0 - A_\infty}{1 - t_{\max}} \right) (t - t_{\max}) + A_\infty \quad (2-14)$$

où l'indice 0 et  $\infty$  désignent la valeur initiale et finale, respectivement.

#### **2.4.4 La 4<sup>ème</sup> modification**

La dernière amélioration que nous avons introduite au BA standard est de permettre aux chauves-souris de mettre à jour le taux d'impulsion et l'intensité, et à accepter une nouvelle solution si leur mouvement produit une solution meilleure que l'ancienne au lieu de la meilleure solution globale, comme dans l'algorithme original. Cette modification a aussi été proposé par [73]. En outre, l'acceptation d'une nouvelle solution nécessite l'accomplissement de deux conditions. Tout d'abord, la solution doit produire une valeur de la fonction objectif inférieure à celle en cours (pour les problèmes de minimisation). D'autre part, un nombre généré de façon aléatoire doit être inférieur à l'intensité sonore correspondante. Il existe une probabilité que le mouvement de la chauve-souris produit une meilleure solution, même à la meilleure solution globale et ne peut être acceptée parce que le nombre généré aléatoirement est plus élevé que l'intensité sonore actuel, en particulier à la fin du processus itératif lorsque la valeur de l'intensité sonore est très faible.

Par conséquent, nous permettons à l'algorithme de mettre à jour la meilleure position globale à chaque fois que le mouvement des chauves-souris produit une solution meilleure avec une valeur de la fonction objectif, même si elle n'a pas été acceptée pour mettre à jour la position de la chauve-souris actuelle. Le pseudocode du nouvel algorithme des chauves-souris directionnelles est illustré dans l'algorithme 2.

### Algorithme 2.2

Le nouvel algorithme des chauves-souris directionnelles (dBA).

---

1. Définir la fonction objectif
  2. Initialiser la population des chauves-souris  $L_i \leq x_i \leq U_i$  ( $i = 1, 2, \dots, n$ )
  3. Evaluer la fonction objectif  $F_i(x_i)$
  4. Initialiser le taux d'impulsions  $r_i$  et l'intensité sonore  $A_i$  et  $w_i$
  5. *Tant que* ( $t \leq t_{\max}$ )
  6.     Sélectionner une chauve-souris aléatoire ( $k \neq i$ )
  7.     Générer les fréquences (Eq. 2-8)
  8.     Mise à jour des positions/solutions (Eq. 2-7)
  9.     *Si* ( $rand > r_i$ )
  10.         Générer une solution locale (Eq. 2-9)
  11.         Mise à jour de  $w_i$  (Eq. 2-10)
  12.     *Fin de Si*
  13.     *Si* ( $rand < A_i \& F(x_i^{t+1}) < F(x_i^t)$ )
  14.         Accepter la nouvelle solution
  15.         Augmenter  $r_i$  (Eq. 2-13)
  16.         Diminuer  $A_i$  (Eq. 2-14)
  17.     *Fin de Si*
  18.     *Si* ( $F(x_i^{t+1}) < F(x^*)$ )
  19.         Mise à jour de la meilleure solution  $x^*$
  20.     *Fin de Si*
  21. *Fin de Tant que*
  22. Traitement des résultats
- 

## 2.5 Simulations et discussions

Pour évaluer les capacités du nouvel algorithme des chauves-souris directionnelles (dBA), nous avons considérés trois expériences. Dans la première, nous analysons les performances de dBA dans la minimisation des fonctions classiques tout en comparons les résultats avec ceux d'une certaine classe d'algorithme standard y compris BA. Dans la deuxième expérience, nous considérons la minimisation de l'ensemble des fonctions benchmarks de la compétition du CEC'2005 [72]. Dans cette expérience, nous effectuons une comparions avec certains algorithmes d'optimisation sophistiqués, tel que l'algorithme d'évolution différentiel auto-adaptatif différentiel (Self-adaptive Differential Evolution, SaDE) [74]. La dernière expérience consiste en une comparaison avec des variantes améliorées de l'algorithme des chauves-souris sur des fonctions de référence classiques.



### 2.5.1 La 1<sup>ère</sup> expérience

Dans cette première expérience, vingt (20) fonctions de références populaires, indiquées dans le Tableau 2.1, ont été utilisées pour vérifier les performances de dBA par rapport à celles de BA, PSO, HS, CS, GA, et DE. Les réglages imposés sur les paramètres de contrôle de chaque algorithme sont obtenus de la littérature et définis comme suit :

- **dBA** : Une analyse approfondie a été réalisée pour régler les paramètres du contrôle de dBA. Pour un meilleur résultat, nous recommandons le réglage des paramètres comme suit :  $r_0 = 0.1$ ,  $r_\infty = 0.7$ ,  $A_0 = 0.9$ ,  $A_\infty = 0.6$ ,  $f_{min} = 0$  et  $f_{max} = 2$ .
- **BA**: L'algorithme de chauve-souris standard a été appliqué comme il est décrit dans [48] avec :  $r_0 = 0.1$ ,  $A_0 = 0.9$ ,  $\alpha = \gamma = 0.9$ ,  $f_{min} = 0$  et  $f_{max} = 2$ .
- **PSO**: L'algorithme d'optimisation par essaim de particules classique [41, 42] a été considéré. Les paramètres de contrôle sont définis comme suit:  $c_1 = 1.5$ ,  $c_2 = 1.2$  et le coefficient d'inertie  $w$  est une fonction décroissante de manière monotone à partir de 0.9 à 0.4.
- **HS**: L'algorithme de recherche de l'harmonie standard a été considéré comme il est décrit dans [44] avec  $BW = 0.2$ ,  $HMCR = 0.95$ ,  $PAR = 0.3$ .
- **CS**: L'algorithme de recherche des coucous avec le vol de Lévy décrit en [46] est considéré, avec une probabilité de la découverte d'œuf étranger  $p_a = 0.25$ .
- **GA**: Algorithme génétique classique [39] avec une probabilité de croisement = 0.95 et la probabilité de mutation = 0.05.
- **DE**: L'algorithme d'évolution différentielle classique tel que décrit dans [40] avec la stratégie "DE/rand/1/bin" est considéré. Le réglage des paramètres est comme suit :  $CR = rand [0.2, 0.9]$  et  $F = rand [0.4, 1]$ .

Pour une comparaison équitable, les paramètres communs sont identiques. La taille de la population a été fixée à  $N = 30$ . La dimension de toutes les fonctions de référence est  $D = 30$ , et le nombre d'évaluation de la fonction objectif est le même (15000) sans compter l'évaluation initiale. Donc, le nombre maximum d'itérations/génération est  $t_{max} = 500$  sauf pour l'algorithme CS où  $t_{max} = 250$ , puisque ce dernier, à chaque itération, il utilise un nombre  $2N$  évaluations de la fonction objectif.

**Tableau 2.1**

Les fonctions de référence classiques (F01-F17).

Nom de la fonction	Formule mathématique	Limites
Sphere	$F01 = \sum_{i=1}^d x_i^2$	$[-100, 100]^d$
Sum of different powers	$F02 = \sum_{i=1}^d  x_i ^{i+1}$	$[-100, 100]^d$
Rotated hyper-ellipsoid	$F03 = \sum_{i=1}^d \sum_{j=1}^i x_j^2$	$[-65, 65]^d$
Griewank	$F04 = \sum_{i=1}^d \frac{x_i^2}{4000} - \prod_{i=1}^d \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	$[-600, 600]^d$
Trid	$F05 = \sum_{i=1}^d (x_i - 1)^2 - \sum_{i=2}^d x_i x_{i-1}$	$[-d^2, d^2]^d$
Rastrigin	$F06 = 10d + \sum_{i=1}^d [x_i^2 - 10 \cos(2\pi x_i)]$	$[-5.12, 5.12]^d$
Levy	$F07 = \sin^2(\pi w_1) + \sum_{i=1}^{d-1} (w_i - 1)^2 [1 + 10 \sin^2(\pi w_i + 1)] + (w_d - 1)^2 [1 + 10 \sin^2(\pi w_d)]$ where $w_i = 1 + (x_i - 1) / 4$	$[-5.12, 5.12]^d$
Ackley	$F08 = -20 \exp\left(-0.2 \sqrt{\frac{1}{d} \sum_{i=1}^d x_i^2}\right) - \exp\left(\frac{1}{d} \sum_{i=1}^d \cos(2\pi x_i)\right) + 20 + \exp(1)$	$[-32, 32]^d$
Schwefel	$F09 = 418.9829d - \sum_{i=1}^d x_i \sin(\sqrt{ x_i })$	$[-500, 500]^d$
Rosenbrock	$F10 = \sum_{i=1}^{d-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	$[-10, 10]^d$
Zakharov	$F11 = \sum_{i=1}^d x_i^2 + \left(\sum_{i=1}^d 0.5ix_i\right)^2 + \left(\sum_{i=1}^d 0.5ix_i\right)^4$	$[-5, 10]^d$
Dixon-price	$F12 = (x_1 - 1)^2 + \sum_{i=2}^d i(2x_i^2 - x_{i-1})^2$	$[-10, 10]^d$
Michalewicz	$F13 = -\sum_{i=1}^d \sin(x_i) \sin^{20}\left(\frac{ix_i^2}{\pi}\right)$	$[0, \pi]^d$
Powell	$F14 = \sum_{i=1}^{d/4} [(x_{4i-3} + 10x_{4i-2})^2 + 5(x_{4i-1} - x_{4i})^2 + (x_{4i-2} - 2x_{4i-1})^4 + 10(x_{4i-3} + x_{4i})^4]$	$[-10, 10]^d$
Bent cigar	$F15 = x_1^2 + 10^6 \sum_{i=2}^d x_i^2$	$[-10, 10]^d$
Alpine	$F16 = \sum_{i=1}^d  x_i \sin(x_i) + 0.1x_i $	$[-10, 10]^d$
Weierstrass	$F17 = \sum_{i=1}^d \left( \sum_{k=0}^{20} [0.5^k \cos(2\pi \cdot 3^k (x_i + 0.5))] \right) - d \sum_{k=0}^{20} [0.5^k \cos(2\pi \cdot 3^k \cdot 0.5)]$	$[-0.9, 0.9]^d$

**Tableau 2.1 (suite)**

Les fonctions de référence classiques (F18-F20).

Nom de la fonction	Formule mathématique	Limites
Styblinski-Tang	$F18 = 0.5 \sum_{i=1}^d (x_i^4 - 16x_i^2 + 5x_i) + 39.16599d$	$[-10, 10]^d$
Salomon	$F19 = 1 - \cos \left( 2\pi \sum_{i=1}^d x_i \right) + 0.1 \sum_{i=1}^d x_i^2$	$[-100, 100]^d$
Schaffer F7	$F20 = \frac{1}{d-1} \sum_{i=1}^{d-1} \left[ (x_i^2 + x_{i+1}^2)^{0.25} + (x_i^2 + x_{i+1}^2)^{0.25} \sin^2 \left( 50(x_i^2 + x_{i+1}^2)^{0.1} \right) \right]$	$[-100, 100]^d$

Pour une analyse statistique significative, chaque algorithme a été exécuté 51 fois en utilisant une population initiale différente à chaque tour. Le minimum global obtenu après chaque essai a été enregistré pour effectuer des analyses statistiques. Par la suite, la moyenne de la valeur du minimum global, l'écart type (E.T.), la meilleure solution (Min), la médiane et la pire des solutions ont été évalués et présentés dans les Tableaux 2.2 et 2.3.

A partir des résultats présentés dans les Tableaux 2.2 et 2.3, le nouvel algorithme des chauves-souris directionnelles a obtenu les meilleurs résultats pour les 9 fonctions (F1, F3, F4, F10, F12, F14, F15, F18 et F19) et il a obtenu le meilleur médian pour F2 et la meilleure solution pour F16, alors que le GA a obtenu de meilleurs résultats pour les 5 fonctions (F6, F13, F16, F17, et F20). Le DE a de meilleurs résultats pour 3 fonctions (F5, F7 et F8). HS obtient les meilleurs résultats pour F9 et BA pour F11.

Les Figs. 2.2 et 2.3 représentent l'évolution de la moyenne du minimum obtenu à chaque itération pour 51 essais de toutes les fonctions de référence classiques. Au début du processus itératif, et en raison de la faible valeur du taux d'impulsion, l'algorithme favorise la phase d'exploration qui permet au dBA d'explorer une vaste zone de l'espace de recherche. Alors que le processus itératif continue, la valeur du taux d'impulsion augmente, ce qui maintient un équilibre entre l'exploitation et l'exploration. Vers la fin, à cause de la valeur élevée du taux d'impulsion, la phase d'exploitation favorise ce qui permet à l'algorithme d'améliorer la qualité des résultats obtenus et accélérer la vitesse de convergence. La comparaison avec les autres algorithmes montre la supériorité de dBA sur plusieurs fonctions de référence.

**Tableau 2.2**Résultats statistiques de la minimisation des fonctions classiques ( $F01\sim F10$ ).

Fonction		dBA	BA	PSO	HS	CS	GA	DE
F01	Min	<b>1.927E-03</b>	3.052E-01	1.118E+03	5.919E+03	2.340E+02	5.517E+00	2.481E+01
	Médian	<b>1.408E-02</b>	5.480E+04	2.554E+03	9.621E+03	4.357E+02	6.560E+02	4.120E+01
	Max	<b>2.233E+00</b>	6.569E+04	5.626E+03	1.568E+04	6.119E+02	7.964E+03	8.028E+01
	Moyenne	<b>2.256E-01</b>	4.920E+04	2.852E+03	9.618E+03	4.153E+02	1.678E+03	4.411E+01
	E.T	4.869E-01	1.859E+04	1.105E+03	2.226E+03	9.518E+01	2.032E+03	1.259E+01
F02	Min	1.011E+06	3.313E+09	1.609E+20	2.573E+33	3.229E+17	<b>7.488E+04</b>	9.080E+08
	Médian	<b>8.171E+09</b>	1.294E+45	1.085E+28	7.580E+37	7.654E+19	9.245E+29	1.177E+11
	Max	1.713E+13	5.893E+50	1.724E+34	8.664E+42	2.433E+22	2.390E+41	<b>1.553E+12</b>
	Moyenne	1.363E+12	4.310E+49	1.046E+33	3.533E+41	2.263E+21	1.049E+40	<b>3.051E+11</b>
	E.T	4.261E+12	1.461E+50	3.737E+33	1.697E+42	5.976E+21	4.671E+40	4.102E+11
F03	Min	<b>1.634E-02</b>	8.563E+00	4.828E+03	4.124E+04	1.062E+03	8.280E+01	9.877E+01
	Médian	<b>3.115E-01</b>	2.996E+05	1.383E+04	5.220E+04	1.996E+03	5.373E+03	1.618E+02
	Max	<b>1.256E+02</b>	4.370E+05	3.416E+04	7.472E+04	3.409E+03	3.294E+04	3.850E+02
	Moyenne	<b>1.461E+01</b>	2.612E+05	1.562E+04	5.336E+04	2.138E+03	8.130E+03	1.742E+02
	E.T	3.456E+01	1.348E+05	7.676E+03	8.132E+03	5.493E+02	8.472E+03	6.173E+01
F04	Min	<b>5.049E-03</b>	3.210E+02	3.041E+01	4.375E+01	3.026E+00	1.080E-01	9.989E-03
	Médian	<b>8.544E-02</b>	5.949E+02	7.258E+01	8.306E+01	4.448E+00	1.507E+01	8.997E-02
	Max	<b>5.630E-01</b>	6.848E+02	1.684E+02	1.201E+02	6.797E+00	5.574E+01	2.136E+00
	Moyenne	<b>1.405E-01</b>	5.816E+02	7.481E+01	8.040E+01	4.567E+00	1.900E+01	2.303E-01
	E.T	1.481E-01	7.884E+01	2.717E+01	1.588E+01	9.934E-01	1.828E+01	4.210E-01
F05	Min	1.685E+03	2.967E+06	3.078E+05	5.169E+05	2.831E+04	6.326E+03	<b>-3.276E+03</b>
	Médian	3.553E+04	4.529E+06	5.827E+05	8.395E+05	4.084E+04	2.920E+05	<b>3.007E+03</b>
	Max	9.707E+04	5.495E+06	1.223E+06	1.329E+06	8.620E+04	7.001E+05	<b>2.215E+04</b>
	Moyenne	3.423E+04	4.436E+06	6.204E+05	8.815E+05	4.242E+04	3.194E+05	<b>4.901E+03</b>
	E.T	2.590E+04	6.360E+05	2.312E+05	1.932E+05	1.118E+04	1.916E+05	6.627E+03
F06	Min	6.812E+01	2.420E+02	1.707E+02	1.330E+02	1.129E+02	<b>2.994E+01</b>	2.998E+01
	Médian	1.057E+02	3.074E+02	2.517E+02	1.625E+02	1.378E+02	<b>5.895E+01</b>	1.575E+02
	Max	2.471E+02	3.670E+02	3.456E+02	1.845E+02	1.644E+02	<b>9.913E+01</b>	2.047E+02
	Moyenne	1.193E+02	3.086E+02	2.599E+02	1.580E+02	1.366E+02	<b>5.746E+01</b>	1.551E+02
	E.T	4.023E+01	3.603E+01	3.756E+01	1.558E+01	1.349E+01	1.825E+01	3.368E+01
F07	Min	1.518E+00	3.024E+01	2.126E+01	1.366E+01	2.414E+00	1.093E+00	<b>1.053E+00</b>
	Médian	4.901E+00	6.876E+01	3.604E+01	2.384E+01	4.475E+00	4.073E+00	<b>1.928E+00</b>
	Max	9.997E+00	1.135E+02	8.057E+01	3.540E+01	8.813E+00	1.562E+01	<b>3.388E+00</b>
	Moyenne	4.716E+00	7.176E+01	3.979E+01	2.417E+01	5.153E+00	5.675E+00	<b>2.017E+00</b>
	E.T	1.826E+00	1.927E+01	1.681E+01	5.004E+00	1.865E+00	3.920E+00	5.223E-01
F08	Min	3.214E+00	1.996E+01	1.252E+01	1.338E+01	8.691E+00	2.595E+00	<b>2.302E+00</b>
	Médian	5.681E+00	1.996E+01	1.462E+01	1.559E+01	1.200E+01	5.744E+00	<b>3.191E+00</b>
	Max	8.801E+00	1.996E+01	1.737E+01	1.640E+01	1.750E+01	1.145E+01	<b>3.648E+00</b>
	Moyenne	5.839E+00	1.996E+01	1.474E+01	1.540E+01	1.209E+01	5.920E+00	<b>3.191E+00</b>
	E.T	1.730E+00	7.062E-04	1.235E+00	7.839E-01	1.753E+00	2.453E+00	2.904E-01
F09	Min	2.895E+03	5.685E+03	7.293E+03	<b>2.281E+03</b>	4.522E+03	2.736E+03	4.745E+03
	Médian	4.492E+03	9.365E+03	8.803E+03	<b>3.698E+03</b>	5.045E+03	4.228E+03	5.370E+03
	Max	5.646E+03	1.017E+04	9.480E+03	<b>4.624E+03</b>	5.426E+03	5.993E+03	6.006E+03
	Moyenne	4.357E+03	8.940E+03	8.712E+03	<b>3.722E+03</b>	5.056E+03	4.208E+03	5.407E+03
	E.T	6.414E+02	1.242E+03	5.463E+02	5.060E+02	1.747E+02	7.320E+02	3.363E+02
F10	Min	<b>2.911E+01</b>	3.336E+01	8.566E+03	8.437E+04	6.691E+02	1.048E+02	4.637E+02
	Médian	<b>1.038E+02</b>	2.473E+02	5.394E+04	1.588E+05	9.105E+02	2.756E+03	6.892E+02
	Max	<b>1.011E+03</b>	2.944E+03	2.811E+05	2.346E+05	2.290E+03	4.793E+04	1.304E+03
	Moyenne	<b>1.645E+02</b>	4.916E+02	8.159E+04	1.597E+05	1.073E+03	5.961E+03	7.193E+02
	E.T	1.926E+02	6.275E+02	6.481E+04	4.048E+04	3.967E+02	9.588E+03	2.121E+02

**Tableau 2.3**Résultats statistiques de la minimisation des fonctions classiques ( $F11\sim F20$ ).

Fonction		dBA	BA	PSO	HS	CS	GA	DE
F11	Min	7.536E+01	<b>4.799E+00</b>	5.754E+02	2.960E+02	1.337E+02	1.122E+01	1.414E+02
	Médian	1.561E+02	<b>3.103E+01</b>	1.108E+03	4.023E+02	2.190E+02	7.707E+06	1.879E+02
	Max	2.506E+02	<b>1.334E+02</b>	1.616E+03	5.713E+02	3.009E+02	9.316E+08	2.352E+02
	Moyenne	1.515E+02	<b>4.629E+01</b>	1.054E+03	4.052E+02	2.214E+02	9.884E+07	1.937E+02
	E.T	4.105E+01	3.971E+01	2.706E+02	6.898E+01	4.094E+01	2.076E+08	2.433E+01
F12	Min	<b>7.448E-01</b>	1.323E+00	6.914E+03	4.057E+04	1.059E+02	1.207E+01	2.650E+01
	Médian	<b>5.528E+00</b>	2.187E+01	3.195E+04	7.688E+04	2.200E+02	1.836E+03	6.164E+01
	Max	<b>1.044E+02</b>	9.385E+02	1.202E+05	1.282E+05	6.159E+02	3.631E+04	1.438E+02
	Moyenne	<b>1.911E+01</b>	1.181E+02	3.864E+04	7.853E+04	2.611E+02	6.494E+03	6.790E+01
	E.T	<b>2.917E+01</b>	2.293E+02	2.495E+04	2.813E+04	1.384E+02	9.520E+03	2.559E+01
F13	Min	-2.094E+01	-9.637E+00	-1.307E+01	-1.477E+01	-1.673E+01	<b>-2.473E+01</b>	-1.259E+01
	Médian	-1.470E+01	-8.037E+00	-9.785E+00	-1.396E+01	-1.444E+01	<b>-2.154E+01</b>	-1.123E+01
	Max	-1.017E+01	-6.880E+00	-7.011E+00	-1.208E+01	-1.333E+01	<b>-1.871E+01</b>	-1.011E+01
	Moyenne	-1.495E+01	-8.179E+00	-9.620E+00	-1.376E+01	-1.455E+01	<b>-2.186E+01</b>	-1.121E+01
	E.T	3.135E+00	6.847E-01	1.732E+00	7.115E-01	7.923E-01	<b>1.793E+00</b>	7.085E-01
F14	Min	<b>1.344E+00</b>	8.475E+00	2.180E+03	1.269E+04	1.778E+02	7.206E+01	2.073E+03
	Médian	<b>2.815E+01</b>	4.501E+01	7.595E+03	3.006E+04	3.536E+02	1.455E+03	4.076E+03
	Max	<b>1.918E+02</b>	7.523E+02	3.698E+04	5.492E+04	7.081E+02	6.457E+03	9.651E+03
	Moyenne	<b>4.898E+01</b>	1.695E+02	9.522E+03	3.091E+04	3.554E+02	2.093E+03	4.532E+03
	E.T	5.028E+01	2.220E+02	7.144E+03	1.079E+04	1.311E+02	1.975E+03	1.861E+03
F15	Min	<b>4.499E+01</b>	1.382E+04	3.980E+07	6.110E+07	1.542E+08	1.474E+06	1.283E+05
	Médian	<b>3.283E+02</b>	4.133E+05	7.507E+07	8.407E+07	3.709E+08	1.061E+07	2.388E+05
	Max	<b>2.518E+03</b>	1.247E+07	1.557E+08	1.111E+08	6.179E+08	8.335E+07	3.528E+05
	Moyenne	<b>4.926E+02</b>	1.929E+06	8.307E+07	8.696E+07	3.760E+08	1.803E+07	2.392E+05
	E.T	5.304E+02	3.115E+06	3.129E+07	1.489E+07	1.192E+08	2.213E+07	6.522E+04
F16	Min	<b>3.462E-02</b>	6.731E+00	1.609E+01	9.950E+00	1.112E+01	1.029E-01	1.271E+01
	Médian	3.239E+00	1.505E+01	2.714E+01	1.437E+01	1.404E+01	<b>7.002E-01</b>	1.522E+01
	Max	2.046E+01	2.832E+01	3.970E+01	1.846E+01	1.834E+01	<b>4.277E+00</b>	1.912E+01
	Moyenne	3.716E+00	1.647E+01	2.690E+01	1.433E+01	1.452E+01	<b>1.305E+00</b>	1.528E+01
	E.T	4.409E+00	5.854E+00	6.382E+00	2.381E+00	1.746E+00	1.157E+00	1.682E+00
F17	Min	2.719E+01	3.067E+01	8.767E+00	2.230E+01	2.157E+01	<b>2.534E+00</b>	1.483E+01
	Médian	3.085E+01	3.181E+01	3.105E+01	2.809E+01	2.788E+01	<b>6.883E+00</b>	2.232E+01
	Max	3.320E+01	3.270E+01	3.283E+01	3.144E+01	2.988E+01	<b>1.464E+01</b>	2.805E+01
	Moyenne	3.053E+01	3.178E+01	2.885E+01	2.776E+01	2.718E+01	<b>7.684E+00</b>	2.276E+01
	E.T	1.668E+00	4.720E-01	5.410E+00	2.174E+00	2.463E+00	3.387E+00	3.374E+00
F18	Min	<b>1.131E+02</b>	1.637E+02	5.202E+02	3.915E+02	2.564E+02	2.611E+02	2.948E+02
	Médian	<b>1.979E+02</b>	2.550E+02	6.376E+02	7.343E+02	3.163E+02	3.582E+02	3.687E+02
	Max	<b>2.686E+02</b>	4.179E+02	9.585E+02	1.137E+03	3.596E+02	6.594E+02	4.154E+02
	Moyenne	<b>1.959E+02</b>	2.651E+02	6.776E+02	7.305E+02	3.168E+02	3.621E+02	3.627E+02
	E.T	3.767E+01	6.899E+01	1.252E+02	1.644E+02	2.533E+01	8.306E+01	3.236E+01
F19	Min	<b>3.554E-01</b>	5.082E-01	4.302E+02	6.724E+02	2.426E+01	9.307E+00	3.377E+00
	Médian	<b>1.328E+00</b>	5.697E+03	9.703E+02	9.243E+02	4.450E+01	1.757E+02	5.082E+00
	Max	<b>2.357E+00</b>	7.542E+03	2.292E+03	1.277E+03	8.646E+01	7.373E+02	8.308E+00
	Moyenne	<b>1.417E+00</b>	5.172E+03	1.009E+03	9.074E+02	4.591E+01	2.319E+02	5.193E+00
	E.T	4.826E-01	1.981E+03	4.147E+02	1.325E+02	1.265E+01	1.940E+02	1.241E+00
F20	Min	3.861E+00	5.453E+00	5.735E+00	5.296E+00	5.729E+00	<b>7.870E-01</b>	2.769E+00
	Médian	5.319E+00	5.964E+00	6.729E+00	5.877E+00	6.177E+00	<b>2.228E+00</b>	3.347E+00
	Max	6.766E+00	6.693E+00	7.411E+00	6.838E+00	6.674E+00	<b>3.680E+00</b>	3.761E+00
	Moyenne	5.262E+00	6.019E+00	6.617E+00	5.946E+00	6.187E+00	<b>2.064E+00</b>	3.324E+00
	E.T	7.905E-01	3.268E-01	5.041E-01	3.786E-01	2.475E-01	9.230E-01	2.582E-01

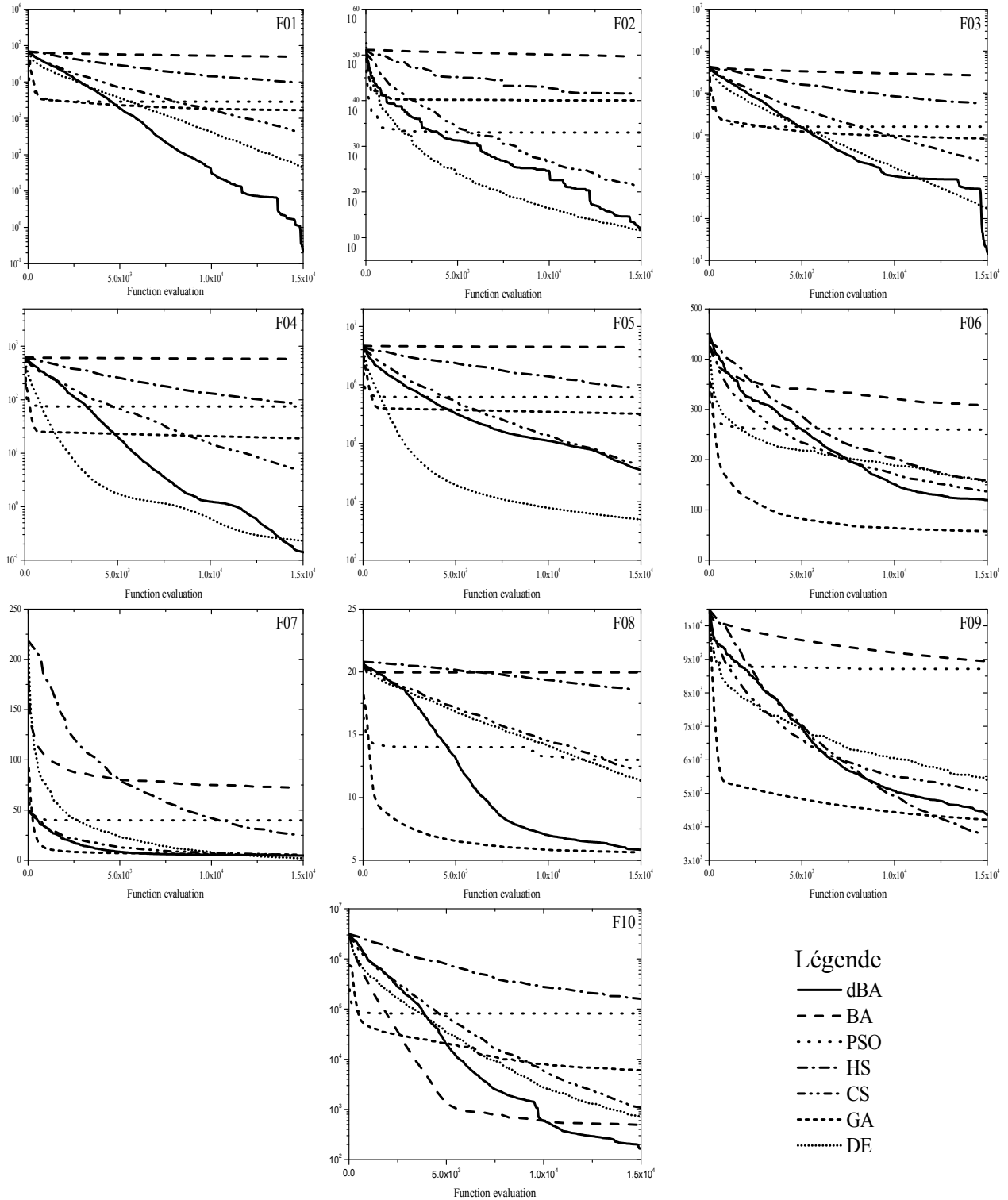


Fig. 2.2 Evolution de la minimisation lors du processus itératif (F01-F10).

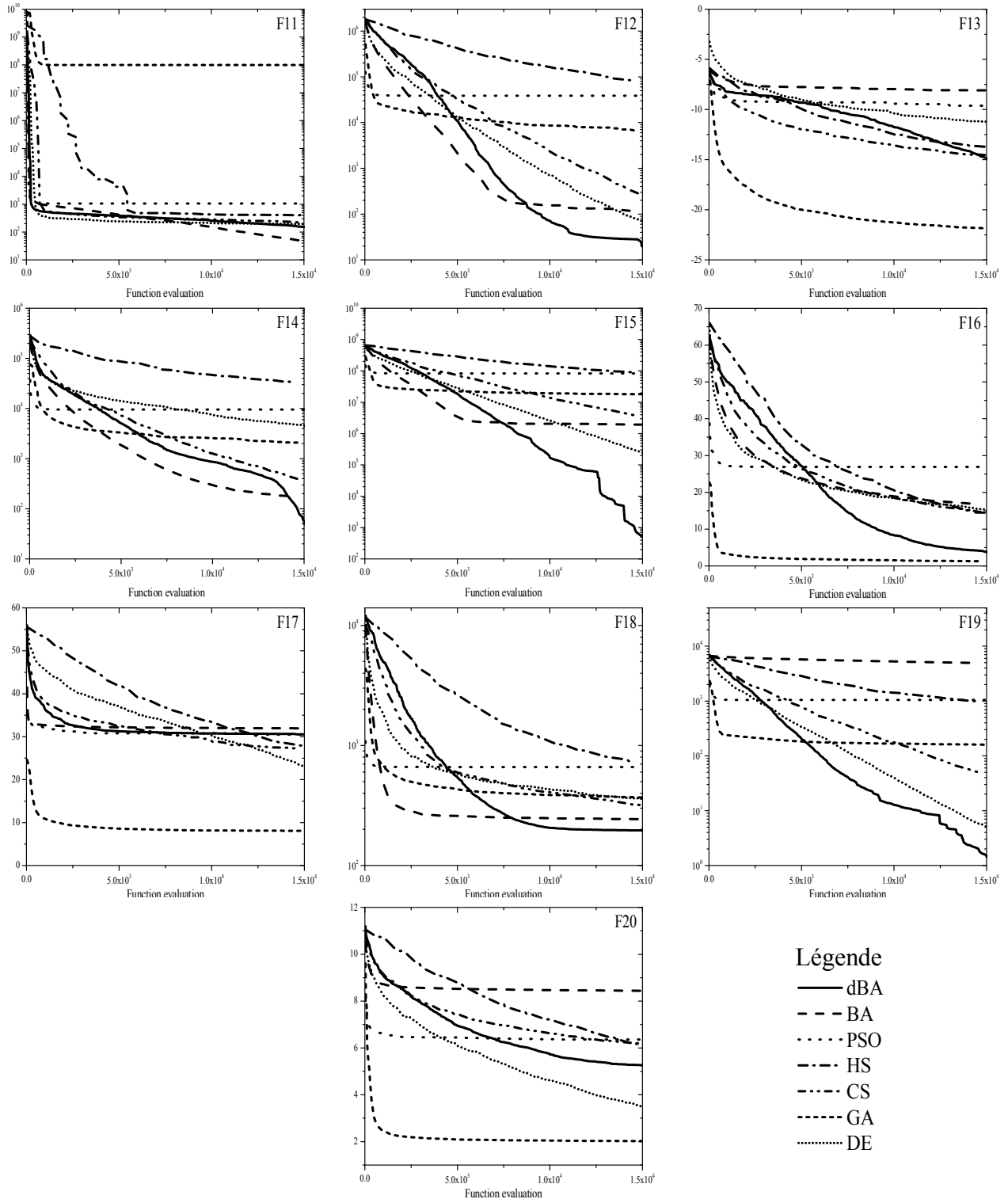


Fig. 2.3 Evolution de la minimisation lors du processus itératif (F11-F20).

**Tableau 2.4**Résultats de comparaison par paires (1<sup>ère</sup> expérience).

dba vs.	BA	PSO	HS	CS	GA	DE
Victoires / Défaites	18/2*	18/2*	17/3*	18/2*	13/7	13/7
<i>p-value</i> du test des signes	<b>4.005E-05</b>	<b>4.005E-05</b>	<b>4.025E-04</b>	<b>4.005E-05</b>	1.153E-01	1.153E-01
<i>p-value</i> du test de Wilcoxon	<b>3.385E-04</b>	<b>1.204E-04</b>	<b>6.806E-04</b>	<b>1.629E-04</b>	<b>9.996E-03</b>	5.691E-02

\*Niveau de signification inférieur à  $\alpha = 0.05$ 

Pour évaluer la performance du dBA, nous avons effectué une série de tests statistiques non paramétriques. Des comparaisons par paires ( $1 \times 1$ ) et des comparaisons multiples ( $1 \times N$ ) ont été considérées. Pour effectuer les comparaisons par paires (un-par-un), nous avons opté sur deux tests, le test des signes (Sign test) et le test de Wilcoxon, tandis que pour les comparaisons multiples, nous utilisons le test de Friedman, le test de Friedman Aligné et le test Quade avec les procédures post-hoc associées, Holland, Rom, Finner et Li [75, 76].

Le Tableau 2.4 présente les résultats de comparaison par paires. La première ligne présente le nombre de victoires et de défaites de dBA par rapport au reste des algorithmes sur chaque problème répertorié dans le Tableau 2.1. L'algorithme est considéré comme un gagnant si la moyenne de 51 points sur un seul problème est mieux que l'autre. Puisque nous avons 20 problèmes, un algorithme est considéré mieux que l'autre au niveau de signification  $\alpha = 0.05$  s'il a au moins 15 victoires (voir Tableau 4 en [75]). D'après les résultats, dBA surpasse nettement le BA standard, PSO, HS, et CS.

La deuxième ligne représente la valeur de *p-value* obtenue par le test des signes (Sign test). Cette dernière signifie la valeur de la probabilité qu'une hypothèse soit vraie (dans notre cas, un algorithme X est meilleur que dBA). Si cette valeur est inférieur à un niveau préalablement défini (traditionnellement  $\alpha = 0.05$ ), on rejette l'hypothèse considéré en faveur de l'hypothèse alternative (dans notre cas, dBA est meilleur que X), sinon, on ne peut rien conclure quant aux hypothèses formulées. D'après les résultats, selon la théorie du test des signes, dBA est supérieur à BA, PSO, HS, et CS. A l'instar du test des signes qui compare seulement les valeurs obtenues par chaque algorithme, le test de Wilcoxon prend en considération aussi le classement de chaque algorithme sur un problème donné. Les valeurs de probabilité de défaillance (*p-value*) obtenues par le test de Wilcoxon sont présentées dans la dernière ligne. Les résultats montrent que dBA surpasse BA, PSO, HS, CS et GA avec un niveau de signification  $\alpha = 0.05$  et DE avec  $\alpha = 0.1$ .



**Tableau 2.5**

Classement des algorithmes selon les tests de Friedman, Friedman Aligné et Quade (1<sup>ère</sup> expérience).

Algorithme	Friedman	Friedman Aligné	Quade
dBA	1.85	50.90	1.58
BA	5.40	93.20	4.99
PSO	5.65	89.05	5.45
HS	5.30	90.35	5.64
CS	3.65	60.45	3.69
GA	3.40	56.55	4.10
DE	2.75	53.00	2.55
<i>p</i> -value	3.51E-10	0.01345	3.91E-10

Pour la réalisation des tests de Friedman, Friedman Aligné et Quade, nous avons utilisé un programme de Java nommé CONTROLTEST proposé et élaboré par Derrac et *al.* [75]. Nous avons téléchargé ce programme du site-web du laboratoire des Systèmes d'information intelligents et Soft Computing de l'université de Granada en Espagne ([www.sci2s.ugr.es/sicidm/](http://www.sci2s.ugr.es/sicidm/)). Le Tableau 2.5 présente le classement de chaque algorithme obtenu par les tests de Friedman, Friedman Alignés et Quade. Pour chaque test, un algorithme est considéré comme meilleur s'il a un classement supérieur (faible valeur). D'après les résultats, dBA a eu le classement le plus haut pour les trois tests, ce qui signifie qu'il est le meilleur en termes de performance. La dernière rangée présente la valeur de *p*-value dont l'hypothèse vraie et la similarité de performance. Si cette dernière est très faible (inférieure à 0.05), le test suggère l'existence de grandes différences entre les algorithmes considérés [75].

Le Tableau 2.6 présente les valeurs de *p*-value obtenues par différentes procédures Post-hoc (Holland, Rom, Finner et Li) et les valeurs *z*-value associées à chaque *p*-valeur. Les *z*-values dans tous les cas, sont utilisées pour trouver leurs correspondants *p*-values dans une distribution normale (Gaussienne) avec une moyenne nulle et un écart type égale à 1 [75]. Dans ce test, l'hypothèse vraie considérée est que un algorithme X soit supérieur à dBA. Les procédures Post-hoc sont appliquées pour ajuster les valeurs de *p*-value pour qu'elles soient plus représentatives de la réalité. Ces procédures sont appliquées aux trois tests de comparaison multiple considérés. L'analyse des *p*-values non-ajustées du test Friedman montre des différences significatives entre dBA et cinq algorithmes (PSO, BA, HS, CS et GA), alors que le test de Friedman aligné considère l'existence de différence avec PSO, HS et BA seulement. Le test Quade révèle que dBA est significativement supérieur à BA, HS, PSO et GA.

**Tableau 2.6**

Valeur de  $p$ -value entre dBA et les autres algorithmes selon les différentes procédures Post-hoc considérées avec un niveau de signification  $\alpha = 0.05$  (1<sup>ère</sup> expérience).

Test	$i$	Algorithme	$z$ -value	$p$ -value	$p_{Holl}$	$p_{Rom}$	$p_{Finn}$	$p_{Li}$
Friedman	7	PSO	5.562630	2.66E-08	1.59E-07	1.52E-07	1.59E-07	3.27E-08
	6	BA	5.196668	2.03E-07	1.01E-06	9.65E-07	6.09E-07	2.50E-07
	5	HS	5.050283	4.41E-07	1.76E-06	1.68E-06	8.82E-07	5.43E-07
	4	CS	2.634930	0.008415	0.025035	0.025246	0.012597	0.010254
	3	GA	2.268968	0.023270	0.045999	0.046541	0.027859	0.027849
	2	DE	1.317465	0.187683	0.187683	0.187683	0.187683	0.187683
Aligned Friedman	7	BA	3.298051	0.000974	0.005827	0.005554	0.005827	0.007430
	6	HS	3.075842	0.002099	0.010451	0.009981	0.006284	0.015883
	5	PSO	2.974484	0.002935	0.011688	0.011194	0.006284	0.022067
	4	CS	0.744596	0.456516	0.839469	0.869941	0.599336	0.778274
	3	GA	0.440520	0.659561	0.884101	0.869941	0.725560	0.835289
	2	DE	0.163733	0.869941	0.884101	0.869941	0.869941	0.869941
Quade	7	HS	3.005520	0.002651	0.015803	0.015126	0.015803	0.005014
	6	PSO	2.867943	0.004131	0.020487	0.019645	0.015803	0.007792
	5	BA	2.525766	0.011545	0.045385	0.044032	0.022956	0.021474
	4	GA	1.866103	0.062027	0.174777	0.186081	0.091582	0.105470
	3	CS	1.562729	0.118116	0.222281	0.236233	0.140010	0.183357
	2	DE	0.716104	0.473927	0.473927	0.473927	0.473927	0.473927

Dans [75], les auteurs ont rapporté que les  $p$ -values non-ajustées ne sont pas adaptés pour les comparaisons multiples en raison de l'accumulation des erreurs, par conséquent, ils ont suggéré d'utiliser les  $p$ -values ajustées. Plusieurs procédures Post-hoc ont été proposées, nous avons choisi les méthodes les plus puissantes. Les valeurs de  $p$ -value ajustées du test de Friedman montrent que les performances de dBA sont nettement supérieures à PSO, BA, HS, CS et GA, tandis que les  $p$ -values du test de Friedman aligné accentuent la différence seulement avec BA, HS et PSO. Sachant que le test Quade prend en compte les difficultés relatives à la résolution des problèmes, les  $p$ -values ajustées correspondantes suggèrent que dBA peut obtenir de meilleurs résultats dans la résolution des problèmes les plus durs que les algorithmes HS, PSO et BA, et qu'il a les mêmes capacités ou mieux dans la résolution des problèmes les plus difficiles que CS, GA et DE.

Pour mesurer à quel point un algorithme est performant que l'autre, l'estimation du contraste a été réalisée. Les résultats sont résumés dans le Tableau 2.7. Chaque ligne présente les résultats de la comparaison entre l'algorithme correspondant dans la première colonne et le reste des algorithmes. Si la valeur estimée est positive, cela signifie que l'algorithme de la première colonne est mieux, et

vice versa si la valeur est négative. En outre, une valeur de contraste élevée signifie qu'il ya de grandes différences entre les algorithmes. En analysant la rangée du nouvel algorithme chauve-souris directionnelle, nous pouvons voir que dBA surpasse le reste des algorithmes.

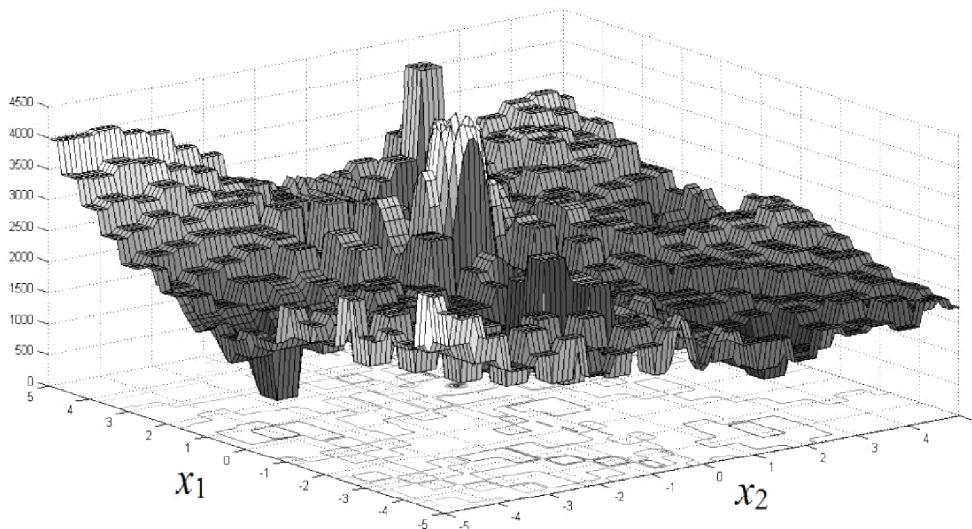
**Tableau 2.7**

Résultats de l'estimation de contraste (1<sup>ère</sup> expérience).

	dBA	BA	PSO	HS	CS	GA	DE
dBA	0.000	<b>295.9</b>	<b>650.2</b>	<b>409.4</b>	<b>165.8</b>	<b>220.9</b>	<b>78.14</b>
BA	<b>-295.9</b>	0.000	354.3	113.5	-130.1	-74.99	-217.8
PSO	<b>-650.2</b>	-354.3	0.000	-240.8	-484.4	-429.3	-572.1
HS	<b>-409.4</b>	-113.5	240.8	0.000	-243.5	-188.4	-331.2
CS	<b>-165.8</b>	130.1	484.4	243.5	0.000	55.10	-87.68
GA	<b>-220.9</b>	74.99	429.3	188.4	-55.10	0.000	-142.8
DE	<b>-78.14</b>	217.8	572.1	331.2	87.68	142.8	0.000

### 2.5.2 La 2<sup>ème</sup> expérience

Dans cette deuxième expérience, nous considérons la série des fonctions complexes utilisées dans la compétition d'optimisation du CEC'2005. L'ensemble se compose de 25 fonctions différentes avec des caractéristiques variantes, telles que l'extensibilité, la séparabilité et la multi-modalité. Toutes ces fonctions sont générées à partir de la fonction de base énumérés dans [72] par décalage, rotation ou hybridation. Ces opérations ajoutent plus de complexité aux problèmes. Les descriptions de ces fonctions sont répertoriées dans le Tableau 2.8. Ces fonctions de référence sont disponibles sur ce site-web [www.ntu.edu.sg/home/EPNSugan](http://www.ntu.edu.sg/home/EPNSugan). Pour avoir une idée sur la complexité de ces fonctions, la Fig. 2.4 est une représentation en 3D de la fonction SF23 à deux dimensions.



**Fig. 2.4** Représentation de SF23 à 2 dimensions.

**Tableau 2.8**

Les fonctions de référence (benchmark) de la compétition de CEC'2005 (en anglais) [72].

Fonction	Limites	$f_{opt}$
Unimodal Functions		
SF01 Shifted Sphere Function	[-100, 100]	-450
SF02 Shifted Schwefel's Problem 1.2	[-100, 100]	-450
SF03 Shifted Rotated High Conditioned Elliptic Function	[-100, 100]	-450
SF04 Shifted Schwefel's Problem 1.2 with Noise in Fitness	[-100, 100]	-450
SF05 Schwefel's Problem 2.6 with Global Optimum on Bounds	[-100, 100]	-310
Multimodal Functions		
Basic Functions		
SF06 Shifted Rosenbrock's Function	[-100, 100]	390
SF07 Shifted Rotated Griewank's Function without Bounds	[0, 600]*	-180
SF08 Shifted Rotated Ackley's Function with Global Optimum on Bounds	[-32, 32]	-140
SF09 Shifted Rastrigin's Function	[-5, 5]	-330
SF10 Shifted Rotated Rastrigin's Function	[-5, 5]	-330
SF11 Shifted Rotated Weierstrass Function	[-0.5, 0.5]	90
SF12 Schwefel's Problem 2.13	$[-\pi, \pi]$	-460
Expanded Functions		
SF13 Expanded Extended Griewank's plus Rosenbrock's Function (F8F2)	[-3, 1]	-130
SF14 Expanded Rotated Extended Scaffe's F6	[-100, 100]	-300
Hybrid Composition Functions		
SF15 Hybrid Composition Function 1	[-5, 5]	120
SF16 Rotated Hybrid Composition Function 1	[-5, 5]	120
SF17 Rotated Hybrid Composition Function 1 with Noise in Fitness	[-5, 5]	120
SF18 Rotated Hybrid Composition Function 2	[-5, 5]	10
SF19 Rotated Hybrid Composition Function 2 with a Narrow Basin for the Global Optimum	[-5, 5]	10
SF20 Rotated Hybrid Composition Function 2 with the Global Optimum on the Bounds	[-5, 5]	10
SF21 Rotated Hybrid Composition Function 3	[-5, 5]	360
SF22 Rotated Hybrid Composition Function 3 with High Condition Number Matrix	[-5, 5]	360
SF23 Non-Continuous Rotated Hybrid Composition Function 3	[-5, 5]	360
SF24 Rotated Hybrid Composition Function 4	[-5, 5]	260
SF25 Rotated Hybrid Composition Function 4 without Bounds	[2, 5]*	260

\* Initial range

En suivant les mêmes critères imposés par le comité d'organisation de la compétition de CEC'2005 [72], l'algorithme des chauves-souris directionnelles a été exécuté 25 fois pour la minimisation de chaque fonction. La dimension considérée est  $D = 10$  et les critères d'arrêt sont après 100 000 évaluations de la fonction objectif. La valeur de l'erreur de fonction ( $f_{min} - f_{opt}$ ) est enregistrée après chaque exécution, puis à la fin des exécutions, un classement de la plus petite à la plus grande est organisé. L'erreur de fonction est considérée comme étant la différence entre le résultat d'optimisation obtenue par dBA et l'optimum global réel (Tableau 2.8). Le Tableau 2.9 présente la 1<sup>ère</sup> (meilleure), la 7<sup>ème</sup>, la 13<sup>ème</sup> (moyenne), la 19<sup>ème</sup> et la 25<sup>ème</sup> (pire) valeurs d'erreur de fonction, en plus de la moyenne et l'écart type des 25 essais pour toutes les fonctions de référence de la compétition de CEC'2005.

**Tableau 2.9**

Valeurs des erreurs des fonctions de référence de la compétition de CEC'2005 pour 25 exécutions de dBA (2<sup>ème</sup> expérience).

Fonction:	SF01	SF02	SF03	SF04	SF05	SF06	SF07	SF08	SF09
1 (Min)	0.000E+00	0.000E+00	1.426E+04	0.000E+00	0.000E+00	1.254E-02	1.232E-02	2.025E+01	9.950E-01
7	0.000E+00	0.000E+00	1.037E+05	1.023E-12	0.000E+00	1.572E+00	1.872E-01	2.035E+01	5.970E+00
13 (Médian)	0.000E+00	0.000E+00	1.992E+05	1.037E-09	0.000E+00	4.191E+00	3.371E-01	2.036E+01	7.960E+00
19	0.000E+00	0.000E+00	3.771E+05	2.524E-06	0.000E+00	2.674E+01	5.612E-01	2.042E+01	1.094E+01
25 (Max)	0.000E+00	0.000E+00	5.250E+05	1.649E-03	0.000E+00	1.139E+03	1.139E+00	2.049E+01	1.393E+01
Moyenne	0.000E+00	0.000E+00	2.389E+05	1.134E-04	0.000E+00	6.646E+01	3.870E-01	2.038E+01	7.880E+00
E.T.	0.000E+00	0.000E+00	1.523E+05	3.546E-04	0.000E+00	2.265E+02	2.744E-01	5.757E-02	3.304E+00
Fonction:	SF10	SF11	SF12	SF13	SF14	SF15	SF16	SF17	SF18
1 (Min)	4.379E+00	1.934E+00	1.725E-02	3.447E-01	2.015E+00	1.140E+02	9.683E+01	9.886E+01	3.000E+02
7	9.108E+00	2.706E+00	1.001E+01	5.705E-01	2.769E+00	1.654E+02	1.086E+02	1.124E+02	3.480E+02
13 (Médian)	1.001E+01	3.169E+00	1.186E+02	8.795E-01	2.969E+00	1.974E+02	1.130E+02	1.236E+02	4.399E+02
19	1.221E+01	3.751E+00	2.125E+02	1.107E+00	3.258E+00	2.302E+02	1.189E+02	1.302E+02	5.303E+02
25 (Max)	1.731E+01	4.613E+00	6.420E+02	1.699E+00	3.470E+00	4.108E+02	1.366E+02	1.474E+02	8.006E+02
Moyenne	1.087E+01	3.223E+00	1.545E+02	8.937E-01	2.950E+00	2.075E+02	1.151E+02	1.219E+02	4.853E+02
E.T.	3.198E+00	7.465E-01	1.740E+02	3.802E-01	3.692E-01	6.305E+01	1.083E+01	1.105E+01	1.689E+02
Fonction:	SF19	SF20	SF21	SF22	SF23	SF24	SF25		
1 (Min)	3.000E+02	3.000E+02	3.000E+02	5.280E+02	4.252E+02	2.000E+02	2.002E+02		
7	3.000E+02	3.000E+02	3.183E+02	5.322E+02	5.539E+02	2.000E+02	3.741E+02		
13 (Médian)	3.693E+02	3.560E+02	4.189E+02	5.381E+02	5.540E+02	2.000E+02	3.786E+02		
19	4.088E+02	3.966E+02	5.000E+02	7.712E+02	5.595E+02	2.000E+02	3.842E+02		
25 (Max)	1.039E+03	1.038E+03	5.004E+02	8.011E+02	5.836E+02	2.000E+02	5.000E+02		
Moyenne	4.205E+02	3.964E+02	4.080E+02	6.307E+02	5.464E+02	2.000E+02	3.608E+02		
E.T.	1.847E+02	1.677E+02	8.121E+01	1.203E+02	3.703E+01	8.343E-13	7.884E+01		

Pour évaluer les performances de dBA, une pratique courante dans le domaine de l'optimisation est de comparer les différents algorithmes en utilisant des benchmarks. Derrac *et al.* [75] ont présenté un tutoriel pratique sur l'utilisation de tests statistiques non paramétriques pour comparer les algorithmes métaheuristiques. Ils ont utilisé les fonctions de la compétition du CEC'2005 pour illustrer l'utilisation des tests de comparaison un-à-un et multiples. Un ensemble composé d'algorithmes évolutionnaires sophistiqués et ceux basés sur l'intelligence des essais ont été utilisés pour comparaison. Nous allons utiliser les résultats de ces algorithmes fournis par [75] pour évaluer les performances de dBA. Les algorithmes considérés sont: l'algorithme de PSO [75], l'algorithme de la stratégie évolutive avec redémarrage de la matrice de covariance et augmentation automatique de la population (IPOP-CMA-ES) [75, 77], l'algorithme CHC [75, 78, 79], l'algorithme génétique stationnaire (SSGA) [75, 80, 81], deux variantes de l'algorithme de recherche de dispersion SS-Arit et SS-BLX [75, 82, 83], l'algorithme d'évolution différentielle classique avec deux stratégies de croisement *Rand/1/exp* (DE-Exp) et *Rand/1/bin* (DE-Bin) [75, 84] et enfin l'algorithme d'évolution différentielle auto-adaptatif (SaDE) [74, 75]. L'IPOP-CMA-ES est le gagnant de la compétition de CEC'2005 sur l'optimisation des problèmes à paramètres réels.

Dans [76], les auteurs ont rapporté que l'exécution de 25 fois un algorithme pour un seul problème est très peu pour effectuer une analyse statistique équitable, mais il était une exigence pour la compétition du CEC'2005 [72]. Par conséquent, les auteurs [75] ont exécuté chaque algorithme énumérés dans le paragraphe précédent 50 fois pour une analyse significative. Par conséquent, nous allons exécuter le dBA 50 fois sur chaque problème figurant dans le Tableau 2.7. Les réglages des paramètres de dBA sont comme elles sont mentionnées dans la Section 2.5.1.

Le Tableau 2.10 présente une comparaison de l'erreur de fonction moyenne obtenue pour les 25 fonctions de référence avec dBA et les autres algorithmes. Les algorithmes sont comparés sur la base de la valeur moyenne de 50 essais et pour  $10^5$  évaluations de la fonction objectif avec  $D = 10$ . Les résultats ont trois chiffres de précision. Si l'erreur est inférieure à  $10^{-10}$ , elle est considérée comme 0. Comme on peut le voir, dBA a surpassé ou exécuté aussi mieux que les autres algorithmes pour douze fonctions : SF01, SF02, SF05, SF07, SF13, SF18 et SF20-25 ce qui représente 48% des problèmes.

**Tableau 2.10**

Comparaison de la moyenne d'erreur obtenue par dBA avec d'autres algorithmes (2<sup>ème</sup> expérience).

Fonction	dBA	PSO	IPOP-CMA-ES	CHC	SSGA	SS-BLX	SS-Arit	DE-Bin	DE-Exp	SaDE
SF01	<b>0.000E+00</b>	1.234E-04	<b>0.000E+00</b>	2.464E+00	8.420E-09	3.402E+01	1.064E+00	7.716E-09	8.260E-09	8.416E-09
SF02	<b>0.000E+00</b>	2.595E-02	<b>0.000E+00</b>	1.180E+02	8.719E-05	1.730E+00	5.282E+00	8.342E-09	8.181E-09	8.208E-09
SF03	2.356E+05	5.174E+04	<b>0.000E+00</b>	2.699E+05	7.948E+04	1.844E+05	2.535E+05	4.233E+01	9.935E+01	6.560E+03
SF04	1.215E-03	2.488E+00	2.932E+03	9.190E+01	2.585E-03	6.228E+00	5.755E+00	<b>7.686E-09</b>	8.350E-09	8.087E-09
SF05	<b>0.000E+00</b>	4.095E+02	8.104E-10	2.641E+02	1.343E+02	2.185E+00	1.443E+01	8.608E-09	8.514E-09	8.640E-09
SF06	3.538E+01	7.310E+02	<b>0.000E+00</b>	1.416E+06	6.171E+00	1.145E+02	4.945E+02	7.956E-09	8.391E-09	1.612E-02
SF07	<b>4.314E-01</b>	2.678E+01	1.267E+03	1.269E+03	1.271E+03	1.966E+03	1.908E+03	1.266E+03	1.265E+03	1.263E+03
SF08	2.035E+01	2.043E+01	<b>2.001E+01</b>	2.034E+01	2.037E+01	2.035E+01	2.036E+01	2.033E+01	2.038E+01	2.032E+01
SF09	8.216E+00	1.438E+01	2.841E+01	5.886E+00	<b>7.286E-09</b>	4.195E+00	5.960E+00	4.549E+00	8.151E-09	8.330E-09
SF10	1.049E+01	1.404E+01	2.327E+01	<b>7.123E+00</b>	1.712E+01	1.239E+01	2.179E+01	1.228E+01	1.118E+01	1.548E+01
SF11	3.758E+00	5.590E+00	<b>1.343E+00</b>	1.599E+00	3.255E+00	2.929E+00	2.858E+00	2.434E+00	2.067E+00	6.796E+00
SF12	1.885E+02	6.362E+02	2.127E+02	7.062E+02	2.794E+02	1.506E+02	2.411E+02	1.061E+02	6.309E+01	<b>5.634E+01</b>
SF13	<b>1.045E+00</b>	1.503E+00	1.134E+00	8.297E+01	6.713E+01	3.245E+01	5.479E+01	1.573E+00	6.403E+01	7.070E+01
SF14	2.965E+00	3.304E+00	3.775E+00	<b>2.073E+00</b>	2.264E+00	2.796E+00	2.970E+00	3.073E+00	3.158E+00	3.415E+00
SF15	2.166E+02	3.398E+02	1.934E+02	2.751E+02	2.920E+02	1.136E+02	1.288E+02	3.722E+02	2.940E+02	<b>8.423E+01</b>
SF16	1.182E+02	1.333E+02	1.170E+02	<b>9.729E+01</b>	1.053E+02	1.041E+02	1.134E+02	1.117E+02	1.125E+02	1.227E+02
SF17	1.266E+02	1.497E+02	3.389E+02	<b>1.045E+02</b>	1.185E+02	1.183E+02	1.279E+02	1.421E+02	1.312E+02	1.387E+02
SF18	<b>4.471E+02</b>	8.512E+02	5.570E+02	8.799E+02	8.063E+02	7.668E+02	6.578E+02	5.097E+02	4.482E+02	5.320E+02
SF19	4.499E+02	8.497E+02	5.292E+02	8.798E+02	8.899E+02	7.555E+02	7.010E+02	5.012E+02	<b>4.341E+02</b>	5.195E+02
SF20	<b>3.946E+02</b>	8.509E+02	5.264E+02	8.960E+02	8.893E+02	7.463E+02	6.411E+02	4.928E+02	4.188E+02	4.767E+02
SF21	<b>4.135E+02</b>	9.138E+02	4.420E+02	8.158E+02	8.522E+02	4.851E+02	5.005E+02	5.240E+02	5.420E+02	5.140E+02
SF22	<b>5.889E+02</b>	8.071E+02	7.647E+02	7.742E+02	7.519E+02	6.828E+02	6.941E+02	7.715E+02	7.720E+02	7.655E+02
SF23	<b>5.595E+02</b>	1.028E+03	8.539E+02	1.075E+03	1.004E+03	5.740E+02	5.828E+02	6.337E+02	5.824E+02	6.509E+02
SF24	<b>2.000E+02</b>	4.120E+02	6.101E+02	2.959E+02	2.360E+02	2.513E+02	2.011E+02	2.060E+02	2.020E+02	<b>2.000E+02</b>
SF25	<b>3.184E+02</b>	5.099E+02	1.818E+03	1.764E+03	1.747E+03	1.794E+03	1.804E+03	1.744E+03	1.742E+03	1.738E+03

**Tableau 2.11**Résultats de comparaison par pairs (2<sup>ème</sup> expérience).

dBA vs.	PSO	IPOP-CMA-ES	CHC	SSGA	SS-BLX	SS-Arit	DE-Bin	DE-Exp	SaDE
Victoires	24	17	18	18	16	21	17	17	17
Nuls	0	2	0	0	1	0	0	0	1
Défaites	1	6	7	7	8	4	8	8	7
$\alpha^*$	<b>0.05</b>	<b>0.05</b>	<b>0.05</b>	<b>0.05</b>	--	<b>0.05</b>	0.1	0.1	0.1
$p$ -value <sup>#</sup>	<b>1.55E-06</b>	<b>0.03469</b>	<b>0.04329</b>	<b>0.04329</b>	0.15159	<b>0.00091</b>	0.10775	0.10775	0.06391
$p$ -value <sup>\$</sup>	<b>0.00022</b>	<b>0.01497</b>	<b>0.00040</b>	<b>0.01725</b>	<b>0.04867</b>	<b>0.00060</b>	0.07356	0.19190	0.10960

\*Niveau de signification selon le nombre des victoires [75]; #:  $p$ -value selon le test des signes ; \$:  $p$ -value selon le test de Wilcoxon**Tableau 2.12**Classement des algorithmes selon les tests de Friedman, Friedman Aligné et Quade (2<sup>ème</sup> expérience).

Algorithme	Friedman	Friedman Aligné	Quade
dBA	3.32	85.20	3.07
PSO	7.72	150.08	7.33
IPOP-CMA-ES	5.32	129.64	5.32
CHC	7.08	179.48	8.24
SSGA	6.36	150.56	6.72
SS-BLX	5.50	124.30	5.92
SS-Arit	6.32	127.56	6.52
DE-Bin	4.44	102.52	4.15
DE-Exp	4.24	102.32	3.75
SaDE	4.70	103.34	3.99
$p$ -value	5.32E-07	0.01052	2.40E-11

Les résultats des tests statistiques non-paramétriques des comparaisons un-par-un sont présentés dans le Tableau 2.11. Ce dernier présente le nombre des victoires, des nuls et des défaites, les  $p$ -values selon le test des signes ainsi que le test de Wilcoxon. L'analyse des valeurs de  $p$ -value du test des signes révèlent que dBA est nettement mieux que PSO, CHC, SSGA, SS-BLX, SS-Arit et le gagnant de la compétition du CEC'2015, l'IPOP-CMA-ES. Selon le nombre de victoires et le nombre de problèmes (25 fonctions de référence), un algorithme est considéré comme meilleur que l'autre avec un niveau de signification  $\alpha = 0.05$  s'il a 18 victoires et avec  $\alpha = 0.1$  s'il a 17 victoires. Nous notons que le nombre des nuls est divisé entre les deux algorithmes, et aussi, s'il ya un nombre impair d'entre eux, le un subliment doit être ignoré.

Par conséquent, dBA surpasse PSO, IPOP-CMA-ES, CHC, SSGA et SS-Arit avec  $\alpha = 0.05$ , et aussi DE-Bin, DE-Exp et SaDE avec  $\alpha = 0.1$ . Les valeurs de  $p$ -value selon le test de Wilcoxon montrent la supériorité de dBA par rapport à PSO, IPOP-CMA-ES, CHC, SSGA, SS-BLX et SS-Arit. Le

Tableau 2.12 présente le classement selon le test Friedman, Friedman Aligné et Quade des dix algorithmes. Comme on peut voir, dBA a la première classe, et les  $p$ -values suggèrent l'existence de grandes différences entre les algorithmes.

Les  $p$ -values ajustés des trois tests sont présentés dans le Tableau 2.13. L'analyse des probabilités des tests de Friedman et de Friedman alignés montre la supériorité de dBA sur PSO, CHC, SSGA, SS-Arit, SS-BLX et IPOP-CMA-ES, tandis que les probabilités du test Quade accentuent seulement les différences entre dBA et CHC. D'après les résultats, nous pouvons conclure que dBA a les mêmes capacités ou mieux que SaDE, DE-Bin et DE-Exp. A partir des résultats d'estimation de contraste présentés dans le Tableau 2.14, le nouvel algorithme de chauve-souris directionnelle surpasse les algorithmes restants, et le plus proche à lui est SaDE et le plus loin est le CHC.

**Tableau 2.13**

Les valeurs de  $p$ -value entre dBA et les autres algorithmes selon les différentes procédures Post-hoc considérées avec un niveau de signification  $\alpha = 0.05$  (2<sup>ème</sup> expérience).

Test	$i$	Algorithme	$z$ -value	$p$ -value	$p_{Holl}$	$p_{Rom}$	$p_{Finn}$	$p_{Li}$
Friedman	10	PSO	5.138093	2.78E-07	2.50E-06	2.37E-06	2.50E-06	3.87E-07
	9	CHC	4.390734	1.13E-05	9.04E-05	8.59E-05	5.08E-05	1.57E-05
	8	SSGA	3.549955	0.000385	0.002694	0.002564	0.001155	0.000537
	7	SS-Arit	3.503245	0.000460	0.002755	0.002622	0.001155	0.000640
	6	SS-BLX	2.545692	0.010906	0.053354	0.051858	0.019545	0.014976
	5	IPOP-CMA-ES	2.335497	0.019517	0.075814	0.074441	0.029133	0.026488
	4	SaDE	1.611493	0.107072	0.288051	0.282675	0.135502	0.129880
	3	DE-Bin	1.307878	0.190915	0.345381	0.282675	0.212059	0.210203
	2	DE-Exp	1.074329	0.282675	0.345381	0.282675	0.282675	0.282675
	Friedman Aligné	10	CHC	4.609548	4.04E-06	3.63E-05	3.45E-05	3.63E-05
9		SSGA	3.195588	0.001395	0.011109	0.010071	0.006264	0.002330
8		PSO	3.172120	0.001513	0.011109	0.010071	0.006264	0.002527
7		IPOP-CMA-ES	2.172765	0.029798	0.165987	0.169999	0.065800	0.047508
6		SS-Arit	2.071070	0.038352	0.177606	0.182363	0.067972	0.060323
5		SS-BLX	1.911681	0.055917	0.205598	0.213271	0.082692	0.085586
4		SaDE	0.886903	0.375131	0.756013	0.402574	0.453685	0.385716
3		DE-Bin	0.846811	0.397100	0.756013	0.402574	0.453685	0.399286
2		DE-Exp	0.837033	0.402574	0.756013	0.402574	0.453685	0.402574
Quade		10	CHC	2.486896	0.012886	0.110175	0.110254	0.110175
	9	PSO	2.048467	0.040514	0.281696	0.308134	0.169818	0.136364
	8	SSGA	1.755194	0.079226	0.438860	0.527272	0.219345	0.235921
	7	SS-Arit	1.658918	0.097132	0.458318	0.554145	0.219345	0.274601
	6	SS-BLX	1.372310	0.169967	0.606017	0.743411	0.284892	0.398463
	5	IPOP-CMA-ES	1.082000	0.279253	0.730144	0.743411	0.388108	0.521147
	4	DE-Bin	0.519893	0.603138	0.937495	0.743411	0.695235	0.701546
	3	SaDE	0.442872	0.657858	0.937495	0.743411	0.700786	0.719405
	2	DE-Exp	0.327340	0.743411	0.937495	0.743411	0.743411	0.743411



**Tableau 2.14**Résultats de l'estimation de contraste (2<sup>ème</sup> expérience).

	dBA	PSO	IPOP-CMA-ES	CHC	SSGA	SS-BLX	SS-Arit	DE-Bin	DE-Exp	SaDE
dBA	0.000	<b>34.97</b>	<b>23.51</b>	<b>72.87</b>	<b>27.00</b>	<b>11.61</b>	<b>15.95</b>	<b>12.13</b>	<b>16.75</b>	<b>10.16</b>
PSO	<b>-34.97</b>	0.000	-11.46	37.90	-7.967	-23.36	-19.02	-22.84	-18.22	-24.81
IPOP-CMA-ES	<b>-23.51</b>	11.46	0.000	49.36	3.493	-11.90	-7.557	-11.38	-6.757	-13.35
CHC	<b>-72.87</b>	-37.90	-49.36	0.000	-45.87	-61.26	-56.92	-60.74	-56.12	-62.71
SSGA	<b>-27.00</b>	7.967	-3.493	45.87	0.000	-15.39	-11.05	-14.87	-10.25	-16.84
SS-BLX	<b>-11.61</b>	23.36	11.90	61.26	15.39	0.000	4.342	0.523	5.143	-1.448
SS-Arit	<b>-15.95</b>	19.02	7.557	56.92	11.05	-4.342	0.000	-3.820	0.800	-5.790
DE-Bin	<b>-12.13</b>	22.84	11.38	60.74	14.87	-0.523	3.820	0.000	4.620	-1.971
DE-Exp	<b>-16.75</b>	18.22	6.757	56.12	10.25	-5.143	-0.800	-4.620	0.000	-6.591
SaDE	<b>-10.16</b>	24.81	13.35	62.71	16.84	1.448	5.790	1.971	6.591	0.000

### 2.5.3 La 3<sup>ème</sup> expérience

La troisième expérience consiste à comparer dBA avec 6 autres variantes de l'algorithme de chauve-souris qui existent dans la littérature, pour être précis : le CBSO [51], BBA [52], SAGBA [53], HSABA [62], EnBA [63] et MBA [65]. Six fonctions de référence classiques qui ont été couramment utilisées dans différentes variantes de BA ont été considérées, à savoir la fonction sphérique, la fonction de Griewank, Rastrigin, Ackley, Rosenbrock et celle de Zakharov (voir Tableau 2.1).

Chaque fonction a été évaluée, par chaque variante de BA, dans différentes conditions de limites, de dimension, de population, du nombre maximum d'itérations et le nombre des essais. Pour une comparaison équitable, nous exécutons le dBA dans les mêmes conditions que chaque auteur a utilisé pour vérifier sa variante de BA proposé. Les résultats sont présentés dans le Tableau 2.15. Comme on le voit, dBA a surpassé les autres algorithmes dans 27 situations sur 32, ce qui représente 84,37% des situations.

La plus récente variante proposée, le CBSO [51], a surpassé dBA dans 3 des situations sur 4. Pour surmonter le déficit d'exploration de CBSO, l'auteur utilise un grand nombre de chauves-souris pour explorer l'espace de recherche, ce qui augmente le nombre d'évaluations de la fonction objectif. Par conséquent, pour le même nombre d'évaluations de fonction qui est  $10^6$ , nous allons exécuter le dBA avec des réglages différents de la taille de la population et du nombre des itérations. Pour chaque situation, nous exécutons dBA 30 fois, et nous calculons la moyenne, l'écart type et le taux de

réussite. Etant donné que l'optimum global des quatre fonctions est 0 (Sphérique, Griewank, Rosenbrock et Zakharov), nous considérons une exécution est réussie si le résultat est inférieur à  $10^{-10}$ . Les résultats sont présentés dans le Tableau 2.16.

**Tableau 2.15**

 Comparaisons entre dBA et les différentes variantes de BA amélioré (3<sup>ème</sup> expérience).

F	Limites	D	N	$t_{max}$	N° essais	(dBA)		Littérature		
						Moyenne	E.T.	Moyenne	E.T	Réf.
F01	[-5.12, 5.12]	30	10000	100	30	<b>1.53E-12</b>	1.17E-12	7.92E-06	8.06E-07	CBSO [51]
	[100, 100]	5	30	500	30	<b>5.55E-38</b>	2.81E-37	1.8518	2.4981	BBA [52]
	[100, 100]	30	40	2000	50	<b>7.32E-26</b>	8.46E-26	4.61E-05	Inconnu	SAGBA [53]
	[-600, 600]	10	100	100	25	<b>6.61E-06</b>	1.17E-05	6.98E-01	2.62E-01	HSABA [62]
	[-5.12, 5.12]	10	50	2000	30	<b>0.00E+00</b>	0.00E+00	9.95E-24	Inconnu	EnBA[63]
	[-5.12, 5.12]	60	50	6000	30	<b>8.99E-17</b>	2.31E-16	1.08E+01	3.70E+00	MBA [65]
F04	[-600, 600]	30	10000	100	30	1.97E-02	1.37E-02	<b>0.0104</b>	0.0021	CBSO [51]
	[-600, 600]	5	30	500	30	<b>9.46E-02</b>	7.43E-02	0.2463	0.0839	BBA [52]
	[-600, 600]	2	40	2000	50	<b>0.00E+00</b>	0.00E+00	3.64E-08	Inconnu	SAGBA [53]
	[-600, 600]	10	100	100	25	<b>1.08E-01</b>	5.64E-02	4.05E-01	2.27E-01	HSABA [62]
	[-600, 600]	10	50	2000	30	<b>1.08E-01</b>	6.83E-02	1.14E+00	Inconnu	EnBA[63]
	[-600, 600]	60	50	6000	30	<b>9.81E-02</b>	1.04E-01	3.19E+02	5.64E+01	MBA [65]
F06	[-5.12, 5.12]	5	30	500	30	1.99E+00	1.26E+00	<b>1.585</b>	1.3352	BBA [52]
	[-5.12, 5.12]*	2	40	2000	50	<b>0.00E+00</b>	0.00E+00	1.58E-07	Inconnu	SAGBA [53]
	[-15, 15]	10	100	100	25	<b>2.50E+01</b>	1.05E+01	8.99E+01	5.87E+01	HSABA [62]
	[-5.12, 5.12]	10	50	2000	30	<b>1.00E+01</b>	3.43E+00	1.19E+01	Inconnu	EnBA[63]
	[-5.12, 5.12]	60	50	6000	30	<b>3.71E+02</b>	3.91E+01	3.84E+02	1.21E+02	MBA [65]
F08	[-32, 32]	5	30	500	30	<b>1.66E-05</b>	4.90E-05	1.16E+00	7.28E-01	BBA [52]
	[-32, 32]*	2	40	2000	50	<b>8.88E-16</b>	0.00E+00	6.19E-04	N.A.	SAGBA [53]
	[-32, 32]	10	100	100	25	<b>5.79E-01</b>	8.71E-01	1.84E+01	2.00E+01	HSABA [62]
	[-32, 32]*	10	50	2000	30	<b>4.22E-08</b>	4.22E-08	8.56E-08	Inconnu	EnBA[63]
	[-32, 32]	60	50	6000	30	<b>1.09E+01</b>	5.28E+00	1.45E+01	8.07E-01	MBA [65]
F10	[-5, 10]	30	10000	100	30	5.76E+01	8.45E+01	<b>0.2194</b>	0.018	CBSO [51]
	[-30, 30]	5	30	500	30	<b>1.54E+00</b>	1.76E+00	25.0743	28.443	BBA [52]
	[-30, 30]*	2	40	2000	50	<b>3.94E-32</b>	7.89E-32	4.68E-06	Inconnu	SAGBA [53]
	[-15, 15]	10	100	100	25	<b>2.71E+01</b>	4.12E+01	1.75E+04	7.10E+01	HSABA [62]
	[30, 30]*	10	50	2000	30	2.59E+00	1.97E+00	<b>4.13E-01</b>	Inconnu	EnBA[63]
	[-2,408,2,408]	60	50	6000	30	<b>1.16E+02</b>	4.89E+01	2.57E+02	6.19E+01	MBA [65]
F11	[-5,10]	30	10000	100	30	3.25E+01	6.95E+00	<b>1.07E-05</b>	4.52E-07	CBSO [51]
	[-5,10]*	2	40	2000	50	<b>0.00E+00</b>	0.00E+00	6.12E-10	Inconnu	SAGBA [53]
	[-5,10]	10	100	100	25	<b>5.33E-01</b>	4.93E-01	1.45E+01	2.20E+00	HSABA [62]
	[-5,10]	10	50	2000	30	<b>7.24E-43</b>	1.15E-42	1.41E-11	Inconnu	EnBA[63]

\*: Limites non mentionné dans la référence correspondante.

**Tableau 2.16**Comparaison entre dBA et CBSO (3<sup>ème</sup> expérience).

N	$t_{max}$	Sphérique			Zakharov		
		Moyenne	E.T.	Taux de succès	Moyenne	E.T.	Taux de succès
50	20000	0.00E+00	0.00E+00	100.00%	1.13E-22	6.11E-22	100.00%
100	10000	4.25E-273	0.00E+00	100.00%	7.39E-24	3.97E-23	100.00%
500	2000	6.89E-107	1.83E-106	100.00%	1.79E-10	2.66E-10	56.67%
1000	1000	2.60E-65	3.76E-65	100.00%	9.13E-05	9.56E-05	0.00%
5000	200	3.72E-18	2.11E-18	100.00%	1.16E+01	3.92E+00	0.00%
10000	100	1.42E-12	1.25E-12	100.00%	3.20E+01	7.68E+00	0.00%
CBSO[51]		7.92E-06	8.06E-07	0.00%	1.07E-05	4.52E-07	0.00%
N	$t_{max}$	Griewank			Rosenbrock		
		Moyenne	E.T.	Taux de succès	Moyenne	E.T.	Taux de succès
50	20000	5.12E-02	6.76E-02	33.33%	3.02E+01	3.93E+01	0.00%
100	10000	5.07E-02	1.15E-01	30.00%	2.49E+01	2.43E+01	0.00%
500	2000	1.15E-02	1.23E-02	33.33%	3.48E+01	3.75E+01	0.00%
1000	1000	1.21E-02	1.29E-02	30.00%	4.75E+01	7.68E+01	0.00%
5000	200	1.05E-02	1.08E-02	36.67%	5.88E+01	7.96E+01	0.00%
10000	100	1.63E-02	1.80E-02	0.00%	5.96E+01	7.45E+01	0.00%
CBSO[51]		1.04E-02	2.10E-03	0.00%	2.19E-01	1.80E-02	0.00%

A partir de l'analyse des solutions min / max de 30 essais obtenues par CBSO, aucun essai n'a été couronné de succès. D'autre part, dBA a réalisé un taux de réussite de 100% dans la minimisation de la fonction sphérique, pour différentes valeurs des populations de chauves-souris et nombre d'itérations maximales. Pour la fonction de Zakharov, dBA obtient 100% du taux de réussite pour une population de chauves-souris inférieure ou égale à 100 individus. Le taux de réussite de la minimisation de la fonction de Griewank est compris entre 30% et 37%. Pour la fonction de Rosenbrock, aucune exécution n'a été réussie et la meilleure solution est obtenue par CBSO.

## 2.6 Conclusions

Dans cette étude, une version améliorée de l'algorithme des chauves-souris standard, appelé algorithme des chauves-souris directionnelles (directional Bat Algorithm, dBA) a été présentée. Quatre modifications ont été intégrées au BA pour accroître ses capacités d'exploitation et d'exploration et ont par conséquent amélioré considérablement les performances de BA. Trois séries d'expériences ont été réalisées pour prouver la supériorité de dBA. Dans la première expérience, 20 fonctions de référence classiques ont été testées. Les résultats d'optimisation ont été comparés à ceux obtenus par des algorithmes classiques, à savoir PSO, HS, CS, GA, DE en plus de BA standard. Les comparaisons statistiques non-paramétriques prouvent la supériorité de dBA. Dans la deuxième

expérience, les fonctions de la compétition du CEC'2005 ont été considérées. Les résultats de dBA ont été comparés avec d'autres algorithmes sophistiqués, et les tests non-paramétriques ont montré que dBA a surpassé certains des algorithmes avancés tels que le SaDE et IPOP-CMA-ES. La troisième expérience révèle que dBA surpasse de nombreuses variantes de BA qui existent dans la littérature.

# **Chapitre 3**

## **Estimation de la fiabilité des structures par l'algorithme des chauves-souris directionnelles**

*Je vis dans l'approximatif et je m'en rapproche de plus en plus (Julos Beaucarne, 1936.)*

### 3.1 Introduction

La détermination de l'indice de fiabilité est un problème d'optimisation sous contraintes. Cependant, les études sur l'application des algorithmes métaheuristiques pour la résolution de ce problème restent limitées. Ceci est dû aux limitations des ordinateurs dans le passé. Ce n'est que dans cette décennie, due à l'augmentation de la puissance des ordinateurs, la communauté scientifique commence à étudier l'usage de ces algorithmes dans le domaine de la fiabilité des structures. Dans l'une des premières études [85], Elegbede applique la méthode de PSO pour effectuer une analyse de fiabilité structurelle. Il a indiqué que cette méthode peut obtenir le MPP et la probabilité de défaillance avec une bonne précision et un faible temps de calcul.

Wang et Ghosn [86] ont utilisé l'algorithme génétique à liaison-découpage qui combine le processus des liaisons d'apprentissage et l'expression du gène de l'algorithme génétique désordonné (GEMGA) à l'opérateur génétique de découpage, pour résoudre le problème de la fiabilité structurelle. Cheng et Li [87] ont utilisé une nouvelle classe de réseau de neurones artificielle (ANN) pour extraire une formule explicite de l'équation d'état limite des structures en treillis, puis ils ont appliqué l'algorithme génétique pour déterminer l'indice de fiabilité.

Hao et *al.* [88] ont appliqué l'algorithme concurrentiel impérialiste (Imperialist Competitive Algorithm, ICA) pour prédire la fiabilité structurelle. Zhao et *al.* [89] ont utilisé l'optimisation chaotique par essaim de particules (CPSO), tandis que Kaveh et *al.* [90] ont utilisé l'algorithme de recherche des systèmes chargés (Charged System Search, CSS) pour résoudre le problème d'optimisation de l'indice de fiabilité. Dans [91], Kaveh et Ghazaan ont comparé les performances de quatre algorithmes métaheuristiques, à savoir, l'algorithme d'optimisation par des rayons améliorés (Improved Ray Optimisation, IRO), l'optimisation par essaim de particules démocratiques (DPSO), l'algorithme d'optimisation par des corps en collision (Colliding Bodies Optimization, CBO) et l'algorithme d'optimisation par des corps en collision amélioré (Enhanced Colliding Bodies

Optimisation, ECBO), sur des problèmes d'évaluation de la fiabilité des structures. Ils ont rapporté que ces algorithmes ont les mêmes performances.

Nous présentons dans ce chapitre, une nouvelle méthode d'estimation de la fiabilité des structures basée sur l'algorithme des chauves-souris directionnelles. Nous étudions l'utilisation et l'application de cet algorithme. L'objectif est de dévoiler les capacités et les potentialités de dBA. Les performances et la précision de l'algorithme proposé sont évaluées et expérimentées en utilisant plusieurs benchmarks et des cas pratiques. Nous introduisons cette méthode en raison de sa simplicité de mise en œuvre et sa capacité à résoudre des problèmes de fiabilité des structures complexes telles que ceux des ponts roulants.

### 3.2 Adaptation de dBA pour l'analyse de fiabilité structurelle

L'indice de fiabilité  $\beta$  est obtenu à partir de la solution du problème d'optimisation sous contraintes suivant, défini dans l'espace  $U$ :

$$\begin{cases} \text{Minimiser } \beta = \sqrt{\sum_i^n u_i^2} \\ \text{Telque: } G(u) = 0 \end{cases} \quad (3-1)$$

La variable  $u_i$  pour chaque dimension  $i$  de l'espace  $U$  est une variable aléatoire normale standardisée (Gaussienne), qui est en fait définie dans  $]-\infty, +\infty [$ . Selon [85], l'espace de recherche peut être réduit à  $[-10, +10]$  en raison du fait que la probabilité  $P[U_i < -10] < 10^{-16}$ . Cela signifie que des événements rares à très faible probabilité d'occurrence ne sont pas considérés. Cela reste réaliste au moins du point de vue de l'ingénierie.

Cependant, pour appliquer dBA, il faut prendre en considération la contrainte d'égalité  $G(u) = 0$  (Eq. 3-1). Il existe plusieurs techniques de traitement des contraintes pour les algorithmes métaheuristiques, comme: L'ensemble des techniques de traitement de contraintes (Ensemble of Constraint Handling Technique, ECHT) [92],  $\varepsilon$ -contraint ( $\varepsilon$ -Constraint Handling Technique,  $\varepsilon$ -CHT) [93], supériorité des solutions réalisables (Superiority of Feasible Solutions, SFS) [94], fonction de pénalité auto adaptative [95], et autres [96-99]. En raison de sa simplicité, la méthode de pénalité est la méthode la plus couramment utilisée dans l'ingénierie [92, 94, 99]. L'idée de base de

la méthode de pénalité est de convertir le problème d'optimisation sous contraintes à un problème d'optimisation sans contraintes, en définissant une nouvelle fonction objectif étendue. Dans le cas de notre problème (Eq. 3-1), la nouvelle fonction objectif étendue est :

$$\Pi(U) = \sqrt{\sum_{i=1}^n u_i^2} + \xi |G(u)| \quad (3-2)$$

où  $\xi$  est le coefficient de pénalité qui est strictement positif. Lorsque la contrainte d'égalité est violée,  $\Pi(U)$  est lourdement pénalisée en additionnant la contrainte multipliée par le coefficient de pénalité, ce qui augmente sa contribution à la fonction objectif de manière significative. Cependant, quand il satisfait, son effet ou sa contribution est nulle.

Le choix de la valeur du coefficient de pénalité  $\xi$  est crucial pour l'algorithme d'optimisation. Il dépend de la valeur absolue que  $G(U)$  peut avoir dans l'espace de recherche, et sa contribution à la fonction objectif. Si la valeur du coefficient de pénalité est élevée, l'algorithme peut se piéger facilement dans des minimaux locaux. D'autre part, si elle est très faible, la détection des solutions optimales faisables devient très difficile. Par conséquent, pour résoudre un problème particulier, il faut réaliser plusieurs essais pour trouver une valeur appropriée de  $\xi$ . Pour surmonter ce problème, plusieurs techniques ont été proposées, telles que la méthode de pénalité à multi-niveaux [100], la méthode de pénalité dynamique [101], la méthode de pénalité de recuit [102] et autres [103-105].

Dans [106], les auteurs ont analysé les difficultés de plusieurs méthodes de pénalité sophistiquées, et comparé leurs performances sur plusieurs problèmes. Ils ont conclu que la fonction de pénalité statique sans aucune amélioration est plus robuste que les autres. Cela est dû au fait que ces méthodes améliorées peuvent bien travailler sur un problème, mais peuvent ne pas fonctionner sur d'autres problèmes [94]. Selon le théorème de pas de déjeuner gratuit (No Free Lunch Theorem, NFLT) [107] et [92], il n'y a pas de technique de traitement des contraintes sophistiquées qui peut surpasser tous les autres sur tous les problèmes.

Pour les problèmes d'évaluation de la fiabilité, dans plusieurs études [85, 88-91, 108-110], les auteurs utilisent la fonction de pénalité statique pour le traitement des contraintes. Cependant, pour chaque problème, ils ont utilisé un coefficient de pénalité différent. Par exemple, dans [85], l'auteur a examiné 6 problèmes, et pour chaque problème, il définit son coefficient de pénalité approprié



(problème 1:  $\xi = 4$ ; problème 2:  $\xi = 0.1$ ; problème 3:  $\xi = 7.2$ ; problème 4:  $\xi = 0.8$ ; problème 5:  $\xi = 0.2$ ; et problème 6:  $\xi = 150$ ). Dans une étude récente [89], les auteurs ont fait la même chose. Ils ont analysé 5 problèmes où chaque problème a son propre coefficient de pénalité (problème 1:  $\xi = 1000$ ; problème 2:  $\xi = 1000$ ; problème 3:  $\xi = 0.1$ ; problème 4:  $\xi = 1000$ ; problème 5:  $\xi = 10$ ).

Pour surmonter ce problème, nous avons défini une fonction qui relie la contrainte et le coefficient de pénalité. Si la violation de contrainte est trop élevée, une petite valeur de  $\xi$  est utilisée pour réduire sa contribution à la fonction objectif. Si elle est petite, une valeur élevée du coefficient de pénalité est utilisée pour forcer le processus de minimisation à converger vers l'optimum avec un minimum de violation de la contrainte. Nous proposons ce coefficient de pénalité adaptatif :

$$\xi(|g(x)|) = 10^{\left(\frac{4}{3} - \frac{1}{3} \log_{10}(|g(x)|)\right)} \quad (3-3)$$

où  $\xi(\cdot)$  est une fonction de la valeur absolue de la contrainte d'égalité.

A partir des frontières de l'espace de recherche  $([-10, 10]^n)$ , la valeur maximale que l'indice de fiabilité peut avoir est  $10\sqrt{n}$ , où  $n$  est la dimension. Par conséquent, nous limitons la valeur maximale que  $\xi$  peut avoir par  $\xi = 10\sqrt{n}$ .

En général, et du point de vue de l'ingénierie, l'état limite peut être une distance, une pression, une force, et le plus souvent des contraintes où leurs valeurs sont exprimées en méga Pascal (MPa). L'addition de la valeur absolue de la contrainte d'égalité à la valeur objectif peut rendre le processus d'optimisation très difficile et infructueux. Par conséquent, nous proposons d'utiliser cette formulation pour construire la fonction d'état limite :

$$g(X) = 1 - (S(X) / R(X)) \quad (3-4)$$

La particularité de cette formulation est qu'elle représente un rapport ou un pourcentage de la quantité de violation de l'équation d'état limite. Nous avons constaté que cette formulation combinée avec le nouveau coefficient de pénalité adaptatif (Eq. 3-3) peut rendre le processus de minimisation très efficace et robuste comme il est indiqué dans la section suivante.

Dans le texte, on se réfère à Eq. (1-1) comme la première forme où la formulation de  $g(X)$  et à l'Eq. (3-4) est la seconde forme ou la formulation de  $g(X)$ .

**Tableau 3.1**

Paramètre des algorithmes considérés.

Algorithme	Paramètres
dBA	$r_0 = 0.1; r_\infty = 0.6; A_0 = 0.9; A_\infty = 0.4; f_{min} = 0; f_{max} = 2$
BA	$r_0 = 0.1; A_0 = 0.9; f_{min} = 0; f_{max} = 2; \alpha = \gamma = 0.9$
PSO	$c_1 = 1.5; c_2 = 1.2; w = 0.75$
GA	Probabilité de croisement = 0.95; Probabilité de mutation = 0.05
HS	$BW = 0.2; HMCR = 0.95; PAR = 0.3$

### 3.3 Résultats de simulation et validation

Dans cette section, quelques exemples pertinents tirés de la littérature sont utilisés pour évaluer les performances de cet algorithme. Dans les exemples suivants, le réglage paramétrique de dBA, BA, PSO, HS et GA sont comme ils sont présentés dans le Tableau 3.1. Pour les paramètres communs, nous avons fixé la population essaim à 100 individus, le nombre maximal des itérations est 500 pour de meilleurs résultats. Les paramètres de PSO sont les mêmes dans [85] à l'exception du paramètre de pénalité.

#### 3.3.1 Validation et analyse de sensibilité

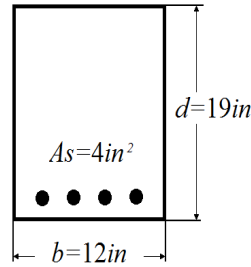
Dans cette section, nous analysons l'effet de la population des chauves-souris, le nombre d'itérations, l'influence de la formulation de l'état limite et l'effet de coefficient de pénalité adaptatif proposé sur l'efficacité et la robustesse de la méthode d'évaluation de la fiabilité basée sur dBA. Nous considérons les exemples suivants :

##### 3.3.1.1 Exemple 1 : poutre en béton armé

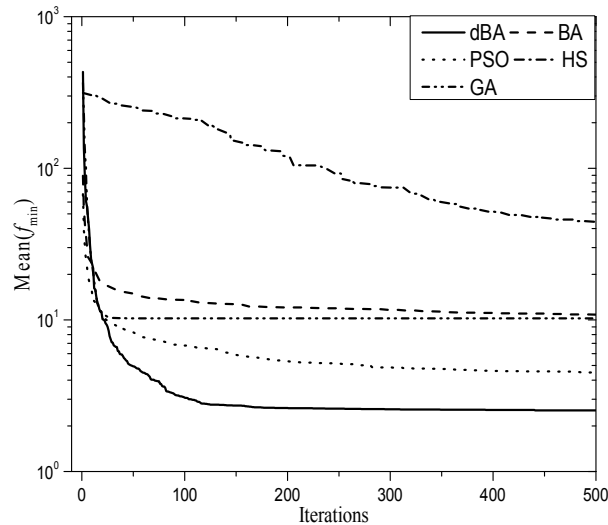
Dans cet exemple, nous effectuons une analyse de la fiabilité d'une poutre en béton armé. La fonction de l'état limite de la poutre est exprimée comme suit :

$$g(F_y, A_s, F_c, Q) = A_s F_y d - 0.59 \frac{(A_s F_y)^2}{F_c b} - Q \quad (3-5)$$

Où  $A_s$ ,  $F_y$ ,  $F_c$  et  $Q$  sont la surface de la section transversale du renforcement, la limite d'élasticité de l'acier, la résistance à la compression du béton et le moment total produit par la charge respectivement. Les propriétés numériques des variables aléatoires sont :  $F_y \sim N(44, 4.62)$ ksi,  $A_s \sim N(4.08, 0.08)$ in<sup>2</sup>,  $F_c \sim N(3.12, 0.44)$ ksi et  $Q \sim N(2052, 246)$ k-in. Les paramètres  $b$  et  $d$  sont respectivement la largeur et la hauteur de la poutre en béton armé comme ils sont indiqué sur la Fig. 3.1.



**Fig. 3.1** La section transversale de la poutre en béton armé.



**Fig. 3.2** Comparaison du minimum obtenu à chaque itération du problème de la poutre en béton armé.

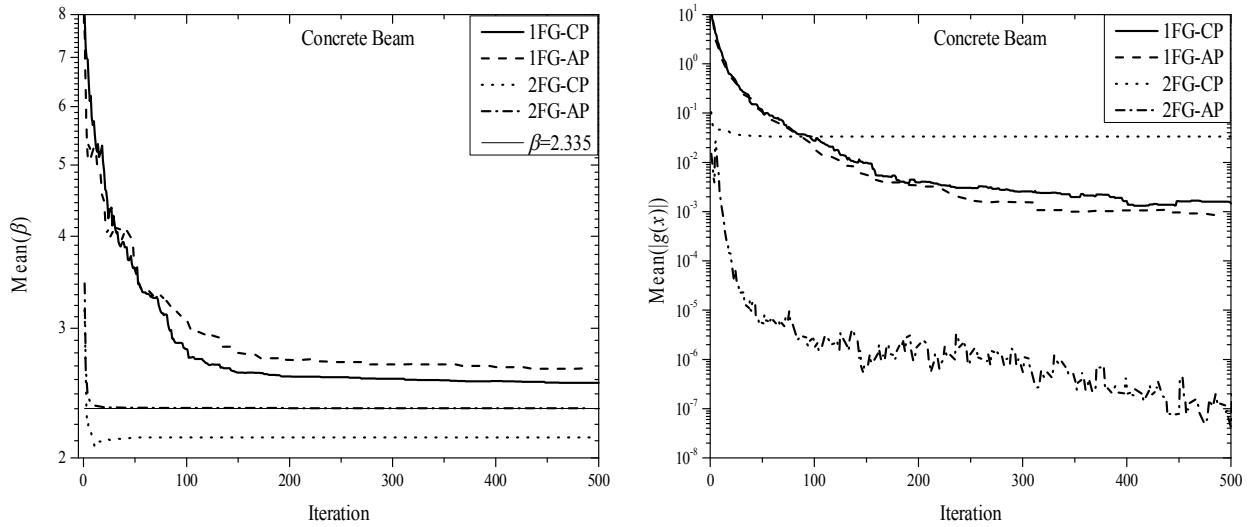
La Fig. 3.2 représente une comparaison de la moyenne des 101 essais, du minimum obtenu à chaque itération de la fonction objectif étendue avec un coefficient de pénalité fixe ( $\xi = 10$ ). La figure montre clairement l'efficacité de dBA contre les algorithmes standards (GA, PSO et HS) et l'algorithme original BA. Nous observons que l'algorithme converge rapidement vers l'optimal par rapport aux autres algorithmes.

Le Tableau 3.2 présente la meilleure, la médiane et la pire solution des 101 exécutions, en plus de la moyenne et l'écart type obtenus avec les différents algorithmes. Les résultats montrent la supériorité de l'algorithme des chauves-souris directionnelles. A partir de l'analyse des écarts-types, nous observons que dBA converge vers la solution finale avec une faible valeur de l'écart-type. Cela signifie que l'erreur commise dans le processus de minimisation est très faible par rapport aux autres algorithmes et les valeurs de l'indice de fiabilité obtenues à chaque exécution sont regroupées étroitement autour de la moyenne, par conséquent, dBA est plus robuste et fiable.

**Tableau 3.2**

Les résultats statistiques d'analyse de fiabilité du problème de la poutre en béton armé.

$\beta$	dBA	BA	PSO	HS	GA
Meilleur	2.355274	2.749308	2.366778	8.126673	2.922244
Médian	2.516543	11.05349	3.855204	27.01016	10.25348
Pire	3.045113	18.73796	9.963497	160.6476	16.78207
Moyenne	2.527775	10.84676	4.499323	43.51636	10.25031
E.T.	0.135245	4.218672	2.000566	36.26901	3.290536


**Fig. 3.3** Effets de la forme de l'équation d'état limite et du coefficient de pénalité sur le processus de minimisation et de violation de contrainte.

Pour augmenter la précision, l'efficacité, la fiabilité, et la robustesse de la méthode d'évaluation de la fiabilité structurelle par l'algorithme des chauves-souris directionnelles, nous récrivons la fonction d'état limite dans la deuxième forme (Eq. 3-4), et nous utilisons le nouveau coefficient de pénalité adaptatif (Eq. 3-3). Donc, Eq. (3-5) dans la deuxième forme de  $g(X)$  devient :

$$g(F_y, A_s, F_c, Q) = 1 - \left( \frac{Q}{A_s F_y d - 0.59 \left( (A_s F_y)^2 / F_c b \right)} \right) \quad (3-6)$$

La Fig. 3.3 représente la moyenne de l'indice de fiabilité ( $\beta$ ) et la violation de contrainte obtenue avec dBA à chaque itération de 100 essais pour 4 cas de la fonction objectif étendue :

- i) 1FG-CP: 1<sup>ère</sup> forme de  $g(X)$  avec un coefficient de pénalité constant,
- ii) 1FG-AP: 1<sup>ère</sup> forme de  $g(X)$  avec un coefficient de pénalité adaptatif,

- iii) 2FG-CP: 2<sup>ème</sup> forme de  $g(X)$  avec un coefficient de pénalité constant,
- iv) 2FG-AP: 2<sup>ème</sup> forme de  $g(X)$  avec un coefficient de pénalité adaptatif.

L'utilisation de la seconde formulation de  $g(x)$  combinée avec le coefficient de pénalité adaptatif (Eq.3-3) permet à l'algorithme d'accélérer la convergence et rend le processus de minimisation plus efficace et robuste avec une faible violation de contrainte.

Pour analyser l'effet du coefficient de pénalité, et la formulation de l'état limite sur la qualité des résultats, dBA a été exécuté 101 fois avec différentes formes de la fonction objectif, et la meilleure solution, la moyenne et le pire sont présentés dans le Tableau 3.3. Les résultats montrent la configuration (2FG-AP) de la fonction objectif conduit à de meilleurs résultats avec une faible violation de contrainte. La comparaison avec l'algorithme de HLRF et la simulation de Monte Carlo révèle que l'indice de fiabilité et le point le plus probable (MPP), obtenus avec l'algorithme des chauves-souris directionnelles combiné avec le coefficient de pénalité adaptatif et la deuxième forme de  $g(X)$ , sont quasi-exacts avec ceux que nous avons obtenus en utilisant l'algorithme HLRF. L'erreur relative qui existe à l'égard de [28] est due au fait que les auteurs ont utilisé la simulation de Monte Carlo pour évaluer l'indice de fiabilité et de la MPP.

**Tableau 3.3**

La comparaison des solutions obtenues avec différentes configurations de la fonction objectif.

Forme de $g(x)$	$\xi$		$\beta$	$ g(x) $	$F_y$	$A_s$	$F_c$	$Q$
$g(x) = R(x) - S(x)$	$\xi=10$	Meilleur	2.349336	0.00E+00	35.61054	4.047048	2.976213	2395.118
		Médiane	2.455894	2.66E-10	37.47352	4.079818	2.653255	2471.684
		Pire	3.028609	6.89E-05	37.93923	3.991933	3.702778	2572.997
	$\xi=f( g(x) )$	Meilleur	2.350067	2.28E-06	36.86609	4.049994	3.026869	2474.732
		Médiane	2.548047	9.75E-08	39.58257	4.066443	2.805759	2604.244
		Pire	3.981556	3.78E-03	42.64968	3.936582	1.657494	2353.822
$g(x) = 1 - \left( \frac{S(x)}{R(x)} \right)$	$\xi=10$	Meilleur	1.696329	9.44E-02	37.88716	4.064392	2.964171	2293.453
		Médiane	2.233071	1.69E-02	36.92790	4.055247	2.876497	2420.293
		Pire	2.372538	0.00E+00	35.26925	4.061632	3.137519	2400.192
	$\xi=f( g(x) )$	Meilleur	2.334826	6.07E-08	36.22243	4.059444	2.970072	2435.890
		Médiane	2.335170	1.89E-08	36.09799	4.059334	2.970710	2428.768
		Pire	2.351603	2.41E-08	35.35674	4.057476	2.964116	2384.349
HLRF		2.335	--	36.23	4.056	2.972	2436	
MCS[28]		2.29	--	35.5	4	3.11	2408.5	

**Tableau 3.4**

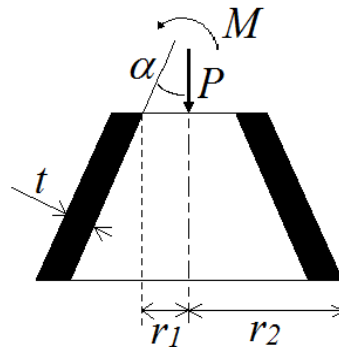
Effet de la taille de la population de chauves-souris et le nombre d'itérations sur la précision de l'indice de fiabilité pour le problème de la poutre en béton.

$t_{max}$	Taille de la population des chauves-souris ( $N$ )									
	25		50		75		100		200	
	Moyenne	E.T.(%)	Moyenne	E.T.(%)	Moyenne	E.T.(%)	Moyenne	E.T.(%)	Moyenne	E.T.(%)
50	2.406678	3.77%	2.351137	0.72%	2.344251	0.90%	2.340749	0.43%	2.335573	0.04%
100	2.402634	2.96%	2.356165	1.29%	2.339174	0.28%	2.336416	0.06%	2.335258	0.02%
250	2.416389	5.29%	2.349446	0.95%	2.337831	0.31%	2.336496	0.16%	2.335037	0.01%
500	2.415098	3.72%	2.345412	0.42%	2.336355	0.08%	2.336274	0.13%	2.334895	0.00%
750	2.400800	2.83%	2.344732	0.65%	2.336238	0.12%	2.336024	0.11%	2.334865	0.00%
1000	2.412979	3.31%	2.350111	1.02%	2.336073	0.07%	2.335408	0.05%	2.334865	0.00%

Le Tableau 3.4 présente une comparaison de la moyenne et l'écart type de l'indice de fiabilité obtenu après 100 exécutions pour différentes tailles de la population des chauves-souris et le nombre des itérations. L'augmentation des populations de chauves-souris peut réduire l'écart-type, ce qui signifie que la solution obtenue à chaque essai est très proche de la moyenne. En outre, la moyenne converge vers l'optimum réel. L'augmentation du nombre des itérations a une contribution minimale à la qualité de l'optimisation par rapport à la taille de la population des chauves-souris.

3.3.1.2 Exemple 2 : une structure conique

Dans cet exemple, nous étudions une structure conique qui représente la tuyère de sortie des gaz d'une fusée, soumise à une charge axiale de compression  $P$ , et un moment de flexion  $M$ . Les caractéristiques géométriques et mécaniques de la structure sont présentées sur la Fig. 3.4. Les paramètres du problème sont définis en tant que variables indépendantes normales et ils sont présentés dans le Tableau 3.5. Cet exemple a été traité en [85].



**Fig. 3.4** La section transversale de la structure conique.

**Tableau 3.5**

Les variables aléatoires de l'exemple 2, structure conique.

$x_i$	Variable	Distribution	$\mu$	$\sigma$	CdV(%)
$x_1$	$E$ (MPa)	Normal	70000.00	3500.00	5.00
$x_2$	$t$ (m)	Normal	0.00250	0.000125	5.00
$x_3$	$\alpha$ (rd)	Normal	0.524	0.010480	2.00
$x_4$	$r_1$	Normal	0.900	0.022500	2.50
$x_5$	$M$ (Nm)	Normal	80000.00	6400.00	8.00
$x_6$	$P$ (N)	Normal	70000.00	5600.00	8.00

Deux phénomènes principaux sont impliqués dans la défaillance de la structure, la perte de résistance et la déformation due à l'instabilité. Pour la perte de la résistance, selon [85], ce mode de défaillance ne sera pas considéré, en raison de la grande marge obtenue dans l'analyse, seulement le flambage de la structure qui sera analysé sous les sollicitations combinées. Selon les règles de conception des véhicules spatiaux de la NASA [111], le critère de flambement est:

$$\frac{P}{P_c} + \frac{M}{M_c} \geq 1 \quad (3-7)$$

$P_c$  et  $M_c$  sont respectivement la charge critique et le moment de flexion critique pour le flambement :

$$P_c = \gamma \frac{2\pi Et^2 \cos(\alpha)^2}{\sqrt{3(1-\mu^2)}} \quad (3-8)$$

$$M_c = \eta \frac{2\pi Et^2 r_1 \cos(\alpha)^2}{\sqrt{3(1-\mu^2)}} \quad (3-9)$$

$\eta$  et  $\gamma$  permettent de corréler les résultats théoriques avec les résultats expérimentaux ( $\gamma = 0.33$ ,  $\eta = 0.41$ ),  $\mu$  et le coefficient de Poisson ( $\mu = 0.334$ ). En utilisant les équations. (3-8 et 9), la fonction d'état limite de la structure est :

$$g(X) = 1 - \frac{\sqrt{3(1-\mu^2)}}{\pi x_1 x_2^2 \cos(x_3)^2} \left( \frac{x_6}{2\gamma} + \frac{x_5}{\eta x_4} \right) \quad (3-10)$$

La grandeur physique correspondant à chaque  $x_i$  et ses propriétés sont illustrées dans le Tableau 3.5. Nous notons que la fonction d'état limite (Eq. 3-10) proposée par Elegbede en [85] est exprimée dans la deuxième forme de  $g(x)$ .

**Tableau 3.6**

Les résultats statistiques de l'exemple 2 avec différentes tailles de la population.

Population	$\beta$	Erreur	$u_1$	$u_2$	$u_3$	$u_4$	$u_5$	$u_6$	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$	
$n \times 25$	Meilleur	4.8773	0.01%	-1.7765	-4.1145	0.4187	-0.6082	1.6096	0.7530	6.38E+10	0.001986	0.5284	0.8863	90301.1	74216.6
	Médiane	4.8811	0.08%	-1.8513	-4.0234	0.5115	-0.5928	1.7002	0.8402	6.35E+10	0.001997	0.5294	0.8867	90881.1	74705.3
	Pire	4.9069	0.61%	-2.0578	-3.8788	0.3008	-0.9592	1.7362	0.8795	6.28E+10	0.002015	0.5272	0.8784	91111.5	74925.3
$n \times 50$	Meilleur	4.8771	0.00%	-1.8113	-4.1183	0.4004	-0.5625	1.5790	0.7579	6.37E+10	0.001985	0.5282	0.8873	90105.8	74244.5
	Médiane	4.8779	0.02%	-1.7367	-4.1354	0.3844	-0.6235	1.6166	0.7254	6.39E+10	0.001983	0.5280	0.8860	90346.4	74062.3
	Pire	4.8955	0.38%	-1.4814	-4.3479	0.3023	-0.5069	1.4427	0.6614	6.48E+10	0.001957	0.5272	0.8886	89233.3	73703.8
$n \times 75$	Meilleur	4.8771	0.00%	-1.7807	-4.1242	0.3977	-0.5661	1.5973	0.7586	6.38E+10	0.001984	0.5282	0.8873	90222.8	74248.1
	Médiane	4.8774	0.01%	-1.8323	-4.0827	0.3917	-0.5823	1.6318	0.7797	6.36E+10	0.00199	0.5281	0.8869	90443.5	74366.4
	Pire	4.8795	0.05%	-1.8330	-4.0405	0.4186	-0.6378	1.6971	0.8134	6.36E+10	0.001995	0.5284	0.8856	90861.5	74555.3
$n \times 100$	Meilleur	4.8770	0.00%	-1.8008	-4.1119	0.3937	-0.5645	1.6029	0.7694	6.37E+10	0.001986	0.5281	0.8873	90258.8	74308.7
	Médiane	4.8773	0.01%	-1.7828	-4.1393	0.3965	-0.5575	1.5559	0.7659	6.38E+10	0.001983	0.5282	0.8875	89957.5	74288.9
	Pire	4.8796	0.05%	-1.9342	-4.0337	0.4175	-0.5718	1.6294	0.8016	6.32E+10	0.001996	0.5284	0.8871	90428.1	74489.0
HLRF	4.8770	--	-1.7956	-4.1159	0.3998	-0.5702	1.5950	0.7692	6.37E+10	0.001986	0.5282	0.8872	90208.2	74307.4	
[85]	4.883	--	-1.7984	-4.1375	0.0437	-0.5584	1.6086	0.7687	6.37E+10	0.001983	0.5245	0.8874	90295.2	131496.4	

Les résultats présentés dans le Tableau 3.6 sont les meilleurs, la pire et la solution médiane de 101 essais avec 100 itérations du problème de la structure conique. La population des chauves-souris varie en fonction de la dimension du problème ( $n$ ), dans ce cas  $n = 6$ . L'erreur présentée dans le tableau est entre l'indice de fiabilité obtenu avec dBA et celle obtenue avec l'algorithme de HLRF. Pour les quatre tailles de la population des chauves-souris, nous observons que la meilleure solution est très précise et la solution médiane a une erreur inférieure à 0,1%, ce qui signifie que 50% des solutions de 101 essais, ont une erreur inférieure à 0,1%. L'erreur maximale de la pire des solutions est de 0,61%, il est enregistré pour le cas où la population de chauve-souris est égale à  $n \times 25$  (150 chauves-souris). En augmentant la taille de la population, l'erreur de la pire des solutions diminue. Cela signifie que pour une seule exécution, nous pouvons obtenir un indice de fiabilité acceptable du point de vue de l'ingénierie d'un système avec une erreur inférieure à 1%. Pour 100 itérations, nous suggérons d'utiliser une population de taille entre  $n \times 50$  et  $n \times 75$  pour une précision maximale d'une seule exécution de l'algorithme et un temps de calcul faible.

Une large erreur est observée dans la comparaison de  $x_6$  et  $u_3$  par rapport aux résultats de [85]. Ces erreurs disparaissent lorsque les résultats obtenus avec l'algorithme des chauves-souris directionnelles sont comparés avec les résultats de l'algorithme de HLRF. Nous croyons que ces erreurs sont dues à une erreur de frappe qui existe dans le papier d'Elegbede [85], et cela est la preuve: puisque  $X$  et  $U$  sont liées par la transformation de Rosenblatt, si une erreur apparaît dans le calcul de  $u_i$ , elle doit apparaître aussi dans  $x_i$ , mais comme nous pouvons le voir, l'erreur pour  $u_3$  est



800% et pour  $x_3$  est 0.71%. La même remarque  $u_6$  et  $x_6$  (la comparaison est effectuée entre la meilleure solution avec population égale à  $n \times 100$  et [85]).

### 3.3.1.3 Exemple 3 : systèmes en parallèle et en série

En pratique, la fonction d'état limite peut être un système d'équations. Selon la structure et la nature de la défaillance, le mécanisme de défaillance peut être considéré comme un système parallèle ou une série. Dans cet exemple mathématique, deux ensembles des fonctions d'état limite ont été considérés, les systèmes parallèles (Eqs. 3-11 et 12), et les systèmes en série (Eqs. 3-13 et 14). Ces fonctions sont considérées comme des systèmes fortement non linéaires avec des MPPs multiples.

Les systèmes parallèles sont :

$$\begin{cases} g_1(X) = 2 - x_2 + \exp(-0.1x_1^2) + (0.2x_1)^4 \\ g_2(X) = 4.5 - x_1x_2 \\ g_{p1}(X) = \max(g_1, g_2) \end{cases} \quad (3-11)$$

$$\begin{cases} g_1(X) = 2.677 - x_1 - x_2 \\ g_2(X) = 2.500 - x_2 - x_3 \\ g_3(X) = 2.323 - x_3 - x_4 \\ g_4(X) = 2.250 - x_4 - x_5 \\ g_{p2}(X) = \max(g_1, g_2, g_3, g_4) \end{cases} \quad (3-12)$$

et les systèmes en série sont :

$$\begin{cases} g_1(X) = 2 - x_2 + \exp(-0.1x_1^2) + (0.2x_1)^4 \\ g_2(X) = 4.5 - x_1x_2 \\ g_{s1}(X) = \min(g_1, g_2) \end{cases} \quad (3-13)$$

$$\begin{cases} g_1(X) = 0.1(x_1 - x_2)^2 - ((x_1 + x_2) / \sqrt{2}) + 3 \\ g_2(X) = 0.1(x_1 - x_2)^2 + ((x_1 + x_2) / \sqrt{2}) + 3 \\ g_3(X) = x_1 - x_2 + 3.5\sqrt{2} \\ g_4(X) = -x_1 + x_2 + 3.5\sqrt{2} \\ g_{s2}(X) = \min(g_1, g_2, g_3, g_4) \end{cases} \quad (3-14)$$

Pour tous ces systèmes d'équations d'état limite, nous considérons que les variables suivent la loi de distribution normale avec  $\mu_i = 0$  et  $\sigma_i = 1$ . La seconde forme de  $g(X)$  des systèmes parallèles et série sont :

Le premier système parallèle Eq. (3-11):

$$\begin{cases} g_1(X) = 1 - \left( (x_2 - \exp(-0.1x_1^2) - (0.2x_1)^4) / 2 \right) \\ g_2(X) = 1 - (x_1x_2 / 4.5) \\ g_{p1}(X) = \max(g_1, g_2) \end{cases} \quad (3-15)$$

Le second système parallèle Eq. (3-12):

$$\begin{cases} g_1(X) = 1 - ((x_1 + x_2) / 2.677) \\ g_2(X) = 1 - ((x_2 + x_3) / 2.500) \\ g_3(X) = 1 - ((x_3 + x_4) / 2.323) \\ g_4(X) = 1 - ((x_4 + x_5) / 2.250) \\ g_{p2}(X) = \max(g_1, g_2, g_3, g_4) \end{cases} \quad (3-16)$$

Le premier système en série Eq. (3-13):

$$\begin{cases} g_1(X) = 1 - \left( (x_2 - \exp(-0.1x_1^2) - (0.2x_1)^4) / 2 \right) \\ g_2(X) = 1 - (x_1x_2 / 4.5) \\ g_{p1}(X) = \min(g_1, g_2) \end{cases} \quad (3-17)$$

Le second système en série Eq. (3-14):

$$\begin{cases} g_1(X) = 1 + \left( (0.1(x_1 - x_2)^2 - ((x_1 + x_2) / \sqrt{2})) / 3 \right) \\ g_2(X) = 1 + \left( (0.1(x_1 - x_2)^2 + ((x_1 + x_2) / \sqrt{2})) / 3 \right) \\ g_3(X) = 1 - ((-x_1 + x_2) / 3.5\sqrt{2}) \\ g_4(X) = 1 - ((x_1 - x_2) / 3.5\sqrt{2}) \\ g_{s2}(X) = \min(g_1, g_2, g_3, g_4) \end{cases} \quad (3-18)$$

**Tableau 3.7**

Les résultats numériques pour les systèmes parallèles et série dans l'exemple 3.

$g(X)$	$g=R-S$		$g=1-(S/R)$		HLRF	Littérature
	Moyenne	E.T.	Moyenne	E.T.		
$g_{p1}(X)$	3.232712	1.27%	3.217487	0.03%	NC*	3.219[112]
$g_{p2}(X)$	2.774386	2.48%	2.732243	1.45%	NC*	2.738[112]
$g_{s1}(X)$	3.005268	0.48%	3.003554	0.24%	3.000000	3 [113]
$g_{s2}(X)$	3.010446	0.53%	3.001499	0.11%	NC*	3 [2]

\*NC: Non Convergent

Le Tableau 3.7 représente les résultats statistiques de l'indice de fiabilité ( $\beta$ ) pour les systèmes parallèle et en série obtenus avec dBA. Une comparaison avec la littérature a été faite [2, 112, 113]. Les résultats montrent que dBA converge vers une meilleure solution que l'algorithme de HLRF (l'algorithme HLRF n'a pas pu converger pour deux systèmes, Eqs. (3-11, 12 et 14)). Les résultats de l'algorithme des chauves-souris directionnelles ont une bonne concordance avec la littérature [2, 112, 113]. Nous notons que la faible différence qui existe entre nos résultats et ceux de Grooteman [112] est due au fait que le dernier a utilisé la méthode de tirage d'importance radial adaptatif (radial-based importance sampling method), qui exclut une sphère de dimension  $n$  appelée " $\beta$ -sphère" de la partie sûre du domaine de l'échantillonnage, où  $\beta$  est le rayon de cette sphère.

### 3.3.2 Application à quelques problèmes de fiabilité des structures

Dans cette section, nous examinons l'exactitude de la méthode proposée sur quelques benchmarks d'analyse de fiabilité des structures. Pour prouver la fiabilité des résultats obtenus par dBA, nous affichons les résultats numériques d'une seule exécution de l'algorithme avec une comparaison avec la littérature. Pour chaque exemple, la population de chauve-souris est  $n \times 50$  et le nombre maximum des itérations est de 100. Nous tenons en compte le coefficient de pénalité adaptatif.

#### 3.3.2.1 Exemple 4 : une poutre en porte à faux

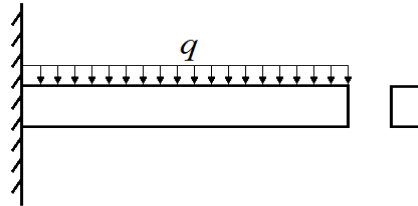
Cet exemple a été analysé par une méthode de surface de réponse moderne nommée CQ2RS, et avec un logiciel commercial RYFES/COMREL dans [114], tandis que dans [85], il a été résolu avec la méthode PSO. La structure en considération est une poutre en porte à faux avec une section transversale rectangulaire, soumise à une charge uniformément répartie (Fig. 3.5). La fonction d'état limite est :

$$g(X) = 18.46154 - 7.476923 \times 10^{10} (x_1/x_2^3) \quad (3-19)$$

où  $x_1$  et  $x_2$  sont deux variables aléatoires normales indépendantes avec  $\mu_1 = 0.001$ ,  $\mu_2 = 250$ ,  $\sigma_1 = 0.0002$  et  $\sigma_2 = 37.5$ . La deuxième forme de l'Eq. (3-19) est :

$$g(X) = 1 - \left( \left( 7.476923 \times 10^{10} (x_1/x_2^3) \right) / 18.46154 \right) \quad (3-20)$$

Les résultats numériques sont présentés dans le Tableau 3.8. La comparaison montre que dBA avec la deuxième forme de l'équation d'état limite, converge vers une solution meilleure que celles obtenues par d'autres méthodes, à savoir, CQ2RS [114], RYFES/COMREL software [114], méthode PSO [85] et le chaotique PSO[89]. En outre, la solution obtenue avec dBA combinée avec la deuxième forme de l'état limite, est plus efficace que les deux autres algorithmes métaheuristiques (PSO et le chaotique PSO). Pour le PSO [85], l'auteur a utilisé 70 particules et 990 itérations (39600 évaluations de la fonction objectif), et pour la chaotique PSO [89], les auteurs ont utilisé 1000 particules et 100 itérations (100000 évaluations de la fonction objectif). Dans notre simulation, nous utilisons 100 chauves-souris et 100 itérations (10000 évaluations de la fonction objectif). Nous notons que dans [85], la fonction d'état limite de ce problème est exprimée dans la 1<sup>ère</sup> forme.



**Fig. 3.5** Une poutre en porte à faux soumise à une charge uniformément répartie.

**Tableau 3.8**

Les résultats numériques de l'exemple 4.

Variable	$g=R-S$	$g=1-(S/R)$	[85]	[89]	[114]*	[114]#
$\beta$	2.332684	2.330846	2.3309	2.3312	2.318	2.331
$P_f$	0.009832	0.009881	0.009879	0.009871	0.010225	0.009877
$u_1$	0.683762	0.584177	0.594605	--	0.586	0.592
$u_2$	-2.230221	-2.256452	-2.253	--	-2.243	-2.255
$x_1$	0.001137	0.001117	0.001119	0.001125	0.001117	0.001118
$x_2$	166.3667	165.3830	165.4372	165.8097	165.8875	165.4375

\*Résultats obtenus par la méthode CQ2RS ; # Résultats obtenus par le logiciel RYFES/COMREL.

3.3.2.2 Exemple 5 : un joint soudé en acier

Cet exemple a été présenté en [115, 116]. Il représente le problème de répartition des contraintes dans un joint soudé en acier soumis à des températures élevées et aux phénomènes de fatigue. La fonction d'état limite est fortement non linéaire et est exprimée comme suit :

$$g(X) = x_1 - 10^4 \left( \frac{x_2(x_4x_5)^{1.71}}{x_3} + \frac{(1-x_2)(x_4x_5)^{1.188}}{x_6} \right) \quad (3-21)$$

En plus de la non-linéarité de la fonction d'état limite, cet exemple est caractérisé par des variables aléatoires avec différentes lois de probabilité. Les variables aléatoires sont listées dans le Tableau 3.9 et la deuxième forme de l'Eq. (3-21) est:

$$g(X) = 1 - \left( \left( 10^4 \left( \frac{x_2(x_4x_5)^{1.71}}{x_3} + \frac{(1-x_2)(x_4x_5)^{1.188}}{x_6} \right) \right) / x_1 \right) \quad (3-22)$$

Le Tableau 3.10 représente la solution obtenue avec une seule exécution de dBA. Les deux formes de l'équation d'état limite ont été considérées avec le coefficient de pénalité adaptatif. L'indice de fiabilité, le MPP et la violation de contrainte sont présentés. Les résultats sont comparés avec [115, 116]. Dans [115], l'auteur a utilisé la méthode de surface de réponse (RSM), et dans [116], les auteurs ont résolu le problème avec une méthode de surface de réponse adaptative basée sur une technique de régression pondérée double. Comme nous pouvons le voir, l'utilisation de la deuxième forme de l'équation d'état limite permet à l'algorithme de converger vers une meilleure solution à faible violation de contrainte, et un bon accord avec la littérature. En fait, la faible violation de contrainte de la solution obtenue avec dBA, en utilisant la deuxième forme de l'état limite, suggère que dBA est plus précis que la méthode de surface de réponse améliorée présentée dans [116].

**Tableau 3.9**  
Caractéristiques des variables aléatoires de l'exemple 5.

Variable	Distribution	$\mu$	$\sigma$
$x_1$	Log-normale	1.0440	0.31320
$x_2$	Normale	0.7000	0.07000
$x_3$	Log-normale	0.2391	0.09564
$x_4$	Log-normale	0.0110	0.15165
$x_5$	Gumbel Max	0.0005	0.00008
$x_6$	Log-normale	1.8020	0.72080

**Tableau 3.10**

Les résultats numériques de l'exemple 5.

Variable	$g = R-S$	$g = 1-(S/R)$	[115]	[116]
$\beta$	2.390477	2.386883	2.385	2.387
$u_1$	-1.219194	-1.283237	-1.267	-1.267
$u_2$	-0.569065	-0.639290	-0.653	-0.653
$u_3$	-0.257523	-0.218084	-0.283	-0.311
$u_4$	0.935855	0.793565	0.831	0.894
$u_5$	0.950312	0.979585	1.01	0.922
$u_6$	-1.434878	-1.415954	-1.382	-1.386
$ G(u^*) $	0.000346	6.00E-06	--	1.10E-04

### 3.4 Pourquoi cet algorithme ?

La détermination de l'emplacement du point le plus probable est crucial. Dans le processus de minimisation, nous rencontrons tous les problèmes communs à l'optimisation non-linéaire comme la non-convergence en raison du bruit numérique, la convergence vers des minima locaux ... etc [117]. Par rapport à l'algorithme de HLRF, dBA est inefficace, mais plus robuste et fiable. En comparaison avec la simulation de Monte Carlo, dBA est moins précis (précision de la méthode de la fiabilité du premier ordre, FORM) mais il nécessite moins de nombre d'évaluations de  $g(X)$  pour obtenir une estimation de la probabilité de défaillance. Par conséquent, nous proposons cet algorithme pour les raisons suivantes :

1. Pour les systèmes à MPP multiples, contrairement à l'algorithme de HLRF, dBA a la capacité de détecter et de converger vers le minimum global (indice de fiabilité) même avec l'existence de bruit ;
2. dBA est plus efficace, robuste et fiable dans le processus de minimisation que d'autres algorithmes de sa classe ;
3. Cet algorithme est très facile à mettre en œuvre, il n'a pas besoin d'un niveau élevé dans la programmation ou de compétences mathématiques, qui est une caractéristique très intéressante du point de vue de l'ingénierie.
4. En divisant la population de chauves-souris en petits groupes, cet algorithme est facilement parallélisé, ce qui permet d'analyser des systèmes de fiabilité complexes avec un grand nombre de variables en utilisant un cluster de calcul à haute performance.

Par conséquent, nous pouvons dire que cette méthode prend sa place entre les méthodes d'approximation classique quand elles ne peuvent pas détecter le MPP, et les méthodes de simulation lorsque le système est très grand et il faut beaucoup de temps pour évaluer la probabilité de défaillance.

### **3.5 Conclusions**

Dans ce chapitre, nous avons proposé une nouvelle méthode d'estimation de la probabilité de défaillance basée sur l'algorithme des chauves-souris directionnelles. La détermination de l'indice de fiabilité et le point le plus probable en utilisant cette méthode a été très réussie. La performance et la robustesse de cet algorithme a été vérifiée par plusieurs benchmarks. La comparaison avec d'autres algorithmes métaheuristiques montre que dBA est plus efficace. Les résultats obtenus sont quasi-exacts par rapport à ceux produits par la méthode de surface de réponse et de la simulation de Monte Carlo. L'utilisation de la deuxième forme de la fonction d'état limite combinée avec le coefficient de pénalité adaptatif augmente l'efficacité et la fiabilité de l'algorithme à converger vers l'indice de fiabilité réel. En plus de la bonne qualité des résultats du dBA, cette méthode est très facile à mettre en œuvre en raison du fait qu'il est un procédé d'ordre zéro, ce qui signifie la non-nécessité de calculer des dérivées.

# **Chapitre 4**

## **Optimisation basée sur la fiabilité par l'algorithme des chauves-souris directionnelles**



*Je ne crains pas des ordinateurs. Je crains de ce qu'il manque en eux (Isaac Asimov, 1920-1992).*

*Les ordinateurs sont inutiles. Ils peuvent seulement vous donner des réponses (Pablo Picasso, 1881-1973).*

## 4.1 Introduction

L'optimisation basée sur la fiabilité (en anglais: Reliability-Based Design Optimization, RBDO) est une méthodologie utilisée dans l'ingénierie pour trouver le meilleur compromis entre les coûts de la fabrication et la sécurité. En général, les méthodes d'optimisation basées sur les approches déterministes peuvent conduire à des conceptions optimales souvent à la limite des contraintes, ce qui se traduit généralement par un produit final avec une forte probabilité de défaillance. Ceci est dû aux incertitudes qui existent dans le processus de fabrication, les propriétés des matériaux, et d'autres en relation avec les conditions de fonctionnement. De plus, ces incertitudes pourraient conduire à de grandes variations dans les performances du système et peuvent finir par une défaillance catastrophique. Par conséquent, ces incertitudes doivent être prises en compte lors de la conception [118]. En utilisant la théorie des probabilités et des statistiques pour modéliser les incertitudes, la méthode RBDO a requis l'usage de l'optimisation afin de trouver la meilleure conception qui satisfait une probabilité de défaillance admissible [119].

Le problème principal rencontré lors de la résolution des problèmes de RBDO est l'évaluation des probabilités de défaillance, car elles nécessitent un effort de calcul considérable. Classiquement, les problèmes RBDO sont formulés en tant que problème d'optimisation stochastique sous contraintes probabilistes, où la résolution est effectuée en incluant l'évaluation de la probabilité de défaillance dans la boucle d'optimisation principale. Cette technique conduit généralement à un problème d'optimisation chevauché, appelé aussi l'approche à double boucle, qui est très coûteux en temps de calcul [120]. Les méthodes les plus connues qui utilisent cette approche sont la méthode de l'indice de fiabilité (Reliability Index Approach, RIA) [121] et la méthode de mesure de la performance (Performance Measure Approach, PMA) [122].

Pour réduire le coût de calcul, deux approches ont été proposées. Le premier est de séparer l'évaluation de la fiabilité de la boucle d'optimisation et de convertir le problème de RBDO à des séquences de cycles d'optimisation déterministes et d'évaluation de la fiabilité. Cette approche, nommée l'approche découplée, est l'idée clé de la méthode d'optimisation et d'évaluation de la fiabilité séquentielle (Sequential Optimization Reliability Assessment, SORA) [123]. La seconde approche, connue comme l'approche à boucle unique, consiste à convertir les contraintes probabilistes à des contraintes déterministes équivalentes. Ainsi, le problème de RBDO devient un problème d'optimisation déterministe standard. Il existe deux méthodes principales qui utilisent cette approche : la méthode d'approche à boucle unique (Single Loop Approach, SLA) proposé par [124, 125] et la méthode de l'espace de conception fiable (Reliable Design Space, RDS) proposée par [126].

En raison de leur grande efficacité, les méthodes analytiques ou les méthodes basées sur le gradient comme la programmation quadratique séquentielle (SQP), ont été utilisées pour résoudre les problèmes RBDO. En général, ces méthodes recherchent une solution dans le voisinage d'un point de départ. Si le problème a des minimums locaux multiples, la solution dépendra du point de départ [127]. En outre, si la fonction objectif et/ou contraintes ont des pics pointus ou multiples, les méthodes analytiques deviennent instables [127, 128]. Afin de surmonter ces lacunes, une nouvelle classe d'algorithmes d'optimisation appelée algorithmes métaheuristiques, a été introduite. Ces algorithmes peuvent avoir une grande capacité à trouver une bonne solution (près de l'optimum), en tirant parti des meilleures caractéristiques réussies des systèmes naturels.

Dans ce chapitre, une nouvelle méthode de RBDO basée sur l'algorithme des chauves-souris directionnelles est présentée. Le concept de l'espace de conception fiable (RDS) est utilisé pour convertir le problème de RBDO en un problème d'optimisation déterministe sous contrainte où la résolution est effectuée en utilisant le dBA. De plus, la méthode  $\varepsilon$ -CHT ( $\varepsilon$ -Constraints Handling Technique) [93] est introduite à dBA de sorte que l'algorithme puisse gérer la satisfaction des contraintes du problème d'optimisation déterministe donné. La suite de ce chapitre est organisée comme suit : dans la section suivante, un aperçu de l'optimisation basée sur la fiabilité est présenté. La méthode RBDO basée sur dBA proposée est décrite dans la Section 4.3, suivie par les résultats expérimentaux dans la Section 4.4, et enfin des conclusions et des discussions seront discutées dans la Section 4.5.

## 4.2 Optimisation basée sur la fiabilité

### 4.2.1 Principes de RBDO

Un problème d'optimisation basé sur la fiabilité peut inclure, en général, des variables d'optimisation (de conception) déterministes qui doivent être déterminées tout en négligeant les incertitudes, un ensemble de paramètres aléatoires et des variables de conception aléatoires dont la moyenne est à déterminer. Les variables et les paramètres de conception aléatoires sont décrits par des distributions de probabilité, dans lequel la variation est contrôlée par la moyenne et l'écart type. Un problème de RBDO typique peut être formulé en tant que problème d'optimisation stochastique, dans lequel la fonction objectif est soumise à des contraintes déterministes et probabilistes. Il est posé comme suit :

$$\left\{ \begin{array}{l} \text{Minimiser } f(d, \mu_x) \\ \text{Tel que:} \\ P(g_i(d, x, p) \leq 0) \leq P_{fa,i}, \quad i = 1 \dots m \\ h_i(d, \mu_x, \mu_p) \leq 0, \quad i = m + 1 \dots n \\ L_{d_j} < d_j < U_{d_j}, \quad j = 1 \dots ND \\ L_{x_j} < \mu_{x_j} < U_{x_j}, \quad j = 1 \dots NX \end{array} \right. \quad (4-1)$$

où  $d = [d_1, d_2, \dots, d_{ND}]^T$  est un vecteur des variables d'optimisation déterministes,  $x = [x_1, x_2, \dots, x_{NX}]^T$  est le vecteur des variables de conception aléatoires et  $p = [p_1, p_2, \dots, p_{NP}]^T$  est le vecteur des paramètres aléatoires.  $\mu$  représente la moyenne de sa variable correspondante.  $L$  et  $U$  désignent les limites inférieure et supérieure, respectivement.  $P(\cdot)$  est la probabilité qu'une condition de se produire et  $P_f$  est la probabilité de défaillance acceptable. La dimension du problème est  $N = ND + NX$ .

L'évaluation d'une contrainte probabiliste n'est pas simple. Elle nécessite l'évaluation de l'intégrale suivante :

$$P(g_i(d, x, p) \leq 0) = \int_{g(d,x,p) \leq 0} f_{x,p}(x, p) dx \quad (4-2)$$

où  $f_{x,p}$  est la fonction de densité de probabilité conjointe (JPDF) des variables et des paramètres de conception aléatoires. Comme l'évaluation exacte de cette intégrale est très difficile, deux ensembles de méthodes d'approximation sont généralement utilisés, à savoir les méthodes de simulation telle

que la simulation de Monte Carlo (MCS) [28] et la méthode de tirage d'importance (Importance Sampling, IS) [113], et les approximations de l'état limite telles que la méthode de fiabilité du premier ordre (First Order Reliability Method, FORM) [11, 12] et la méthode de fiabilité du second ordre (Second Order Reliability Method, SORM) [13, 14]. L'idée principale de FORM est de calculer l'indice de fiabilité  $\beta$  qui représente la distance minimale de la surface d'état limite à l'origine dans l'espace normal (espace Gaussien). L'indice de fiabilité est obtenu en résolvant le problème d'optimisation suivant :

$$\begin{cases} \text{Minimiser } \beta = \sqrt{\sum_i^{NX} u_{xi}^2 + \sum_i^{NP} u_{pi}^2} \\ \text{Tel que: } G(d, u_x, u_p) = 0 \end{cases} \quad (4-3)$$

où  $u$  est la variable aléatoire standard normale (ou gaussienne) obtenue par la transformation de Rosenblat  $u_i = \Phi^{-1}(CDF_i(x_i))$  [129] (même formulation pour  $p$ ).  $\Phi^{-1}$  est la fonction inverse de la distribution normale cumulée, et  $CDF_i(\cdot)$  est la fonction de distribution cumulée de la variable aléatoire  $x_i$ . La solution du problème précédent  $u^*$  est appelée le point le plus probable (Most Probable Point, MPP). L'équation d'état limite normalisée (G) est calculée comme suit :

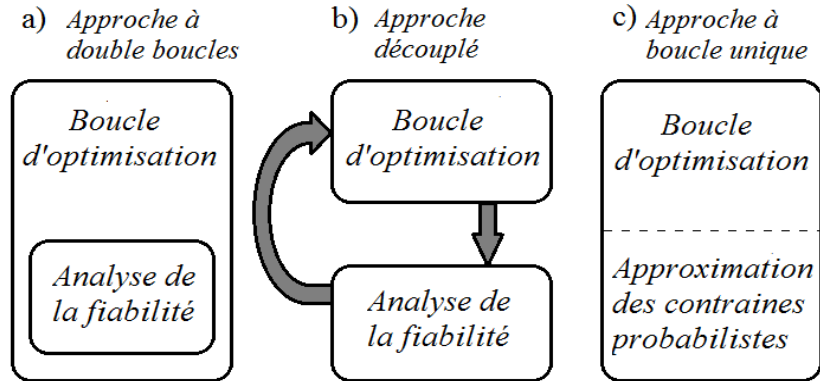
$$G(d, u_x, u_p) = g\left(d, CDF_x^{-1}(\Phi(u_x)), CDF_p^{-1}(\Phi(u_p))\right) \quad (4-4)$$

Par conséquent, la probabilité de défaillance est approximée comme suit :

$$P(g_i(d, x, p) \leq 0) = \Phi(-\beta) \quad (4-5)$$

#### **4.2.2 Revue sur l'ensemble des méthodes de résolution des problèmes de RBDO**

Le problème clé dans la résolution des problèmes RBDO est l'évaluation des contraintes probabilistes, car elles nécessitent des efforts de calcul considérables. En fonction de la technique d'évaluation des contraintes probabilistes, il existe essentiellement trois approches de résolution, à savoir, l'approche à double boucle, l'approche à boucle unique et l'approche découplée.



**Fig. 4.1** Les approches de résolution des problèmes de RBDO.

*L'approche double boucle* (Double Loop Approach, DLA) comme son nom l'indique (Fig. 4.1a) se compose de deux boucles : intérieure et extérieure. Dans la boucle intérieure, l'évaluation de la fiabilité est effectuée en utilisant différentes méthodes itératives et d'échantillonnage, tandis que la boucle extérieure est pour optimiser les variables de conception. Cette approche conduit à un problème d'optimisation chevauché avec un coût de calcul élevé. Deux principaux algorithmes utilisent cette approche. Le premier est l'approche de l'indice de fiabilité (RIA) [121] dans lequel le problème d'optimisation dans l'Eq. (4-3) est résolu pour estimer la probabilité de défaillance. Le second est l'approche de mesure de la performance (PMA) [122] où l'estimation de probabilité est convertie en une mesure de performance en résolvant le problème inverse de l'équation (3). La solution obtenue est appelée le point cible de performance minimum (Minimum Performance Target Point, MPTP). Il est basé sur l'idée que l'optimisation d'une fonction complexe sous une simple contrainte est plus facile que l'inverse.

*L'approche découplée* (Decoupled Approach, DA) consiste à séparer l'évaluation de la fiabilité de la boucle d'optimisation (Fig. 4.1b). L'algorithme le plus prometteur est la méthode d'optimisation et l'évaluation de la fiabilité séquentielle (SORA) [123]. Le procédé SORA transforme le problème de RBDO à des séquences d'optimisation déterministe et d'évaluation de la fiabilité. L'idée est d'utiliser l'information issue de l'analyse de fiabilité du cycle précédent pour déplacer les contraintes déterministes dans le domaine fiable. Il existe de nombreuses autres méthodes qui utilisent l'approche de découplage, telles que l'approche du facteur de sécurité (Safety Factor Approach, SFA) [130, 131] qui à la même idée de déplacement des contraintes fondées sur la base du MPP cible. Le concept de programmation approximative séquentielle (SAP) proposé par [132] formule le

problème de l'évaluation de la fiabilité comme un problème de sous-programmation, où les contraintes probabilistes ont été linéarisées au niveau du MPP.

*L'approche à boucle unique* (Single Loop Approach, SLA) consiste à transformer les contraintes probabilistes en des contraintes déterministes équivalentes (Fig. 4.1c). Dans [133], les auteurs remplacent les contraintes probabilistes par les conditions d'optimalité de Karush-Kuhn-Tucker (KKT) de FORM. Les auteurs de [134] ont indiqué que la méthode KKT présente une stabilité faible et très coûteuse en calcul que les méthodes de double boucle en raison de l'augmentation du nombre des contraintes d'égalité. La méthode à boucle unique vecteur unique (Single Loop Single Vector, SLSV) proposée par [135] consiste à évaluer la fonction d'état limite à un point éloigné du point actuel avec une distance égale à  $\beta$  (indice de fiabilité) dans le sens du MPP à la place du calcul de la probabilité de défaillance. En conséquence, la boucle interne est retirée, et le problème de RBDO devient déterministe. Basé sur le même concept, les auteurs dans [125] ont développé l'approche de boucle simple (SLA), où l'exigence de fiabilité du système a été considérée. Dans [9], les auteurs proposent une nouvelle méthode de boucle unique basée sur l'espace de conception fiable (RDS), où le problème d'RBDO a été converti en problème d'optimisation déterministe dans l'espace de conception fiable. Ce procédé présente le même principe que la SLA proposée dans [124, 125, 135].

#### **4.2.3 RBDO et les algorithmes métaheuristiques**

En raison de la montée en puissance des ordinateurs, plusieurs études ont proposé d'utiliser des algorithmes métaheuristiques pour résoudre les problèmes RBDO [5, 136-145]. En suivant l'approche à double boucle, les auteurs [136] ont utilisé l'algorithme génétique hiérarchique pour résoudre le problème de RBDO des structures composites. Les contraintes de fiabilité ont été évaluées en utilisant FORM. Dans le but de concevoir un système fiable pour la distribution d'eau, les auteurs de [137] ont utilisé l'algorithme génétique comme un outil d'optimisation et FORM pour estimer les contraintes probabilistes. Dans [138], les auteurs ont appliqué l'algorithme PSO modifié, à savoir l'algorithme PSO avec autoréglage et limites rétrécissant (AB-PSO) pour optimiser les structures en treillis avec des variables discrètes sous des contraintes de fiabilité. Les dernières ont été évaluées en utilisant la simulation par sous-ensemble (Subset Simulation, SS). Dans [139], les auteurs ont utilisé la méthode de PSO pour effectuer la RBDO d'un réservoir sous pression en matériau composite. Ils ont utilisé une procédure itérative pour évaluer l'indice de fiabilité de contraintes probabilistes obtenues par un modèle d'éléments finis.

Dans [140], les auteurs proposent une méthode basée sur PSO, la simulation par sous-ensemble (SS) et la machine à vecteurs de support (SVM). En premier, des solutions de conception initiales ont été produites au hasard, et leurs satisfactions des contraintes probabilistes ont été vérifiées en utilisant la simulation par sous-ensemble. Les solutions et leurs étiquettes de faisabilité ont été utilisées par le classificateur SVM comme solution d'apprentissage pour construire des fonctions de décision. Ces fonctions ont été nourries à l'algorithme de PSO pour vérifier la faisabilité des nouvelles solutions générées. Après un certain nombre de générations, les solutions obtenues par PSO sont transférées à la SVM comme de nouvelles solutions d'apprentissage pour la mise à jour des fonctions de décision.

En utilisant l'approche de boucle unique, les auteurs dans [5], ont proposé une méthode d'optimisation basée sur la fiabilité en utilisant l'algorithme de la stratégie d'évolution (ES) comme moteur d'optimisation principale, pour réaliser l'RBDO des systèmes structurels à grande échelle. L'évaluation des contraintes a été obtenue grâce à un réseau neuronal entraîné, où la satisfaction des contraintes déterministes a été vérifiée par une analyse par éléments finis (FEA) et l'évaluation des contraintes probabilistes a été obtenue en utilisant la simulation de Monte Carlo.

Dans [141], les auteurs ont utilisé le PSO binaire pour chercher une conception optimale et fiable des structures en treillis. En considérant que le hasard existe seulement dans la charge, la limite d'élasticité et l'aire en coupe transversale, ils ont obtenu une expression analytique de l'indice de fiabilité. Ainsi, le problème de RBDO a été converti en un problème d'optimisation déterministe à boucle unique.

Dans [146], les auteurs ont converti la contrainte probabiliste multi-variable à une contrainte à une seule variable en utilisant des coefficients de polynômes exponentiels, où l'intégrale de probabilité de défaillance a été évaluée par le moyen des quadratures adaptives de Gauss-Kronrod. Ainsi, le problème de RBDO a été de nouveau converti en un problème d'optimisation déterministe où la solution a été calculée en utilisant l'algorithme d'optimisation par essaim des particules.

Comme pour les problèmes d'RBDO mono-objectif, les algorithmes métaheuristiques ont été appliqués pour résoudre les problèmes de RBDO multi-objectifs (MO-RBDO). Dans [142], les auteurs ont proposé un procédé de résolution de MO-RBDO en utilisant l'algorithme génétique à tri non-dominé II (NSGA-II), où les contraintes probabilistes ont été évaluées à l'aide de PMA-Rapide [147]. Dans [143], les auteurs ont employé le GA multi-objectif combiné avec la méthode de tirage

d'importance (Importance Sampling, SI) pour l'estimation des contraintes probabilistes. Dans [144], les auteurs ont utilisé le PSO multi-objectif où l'évaluation de la fiabilité a été obtenue par méthode hybride à valeur moyenne (Hybride Mean Value, HVM) proposée par [148]. Dans [145], les auteurs ont adopté la méthode SLSV proposée par [135] pour l'estimation de la probabilité de défaillance, tandis que la PSO multi-objectif a été appliquée pour résoudre le problème MO-RBDO.

### 4.3 RBDO avec l'algorithme des chauves-souris directionnelles

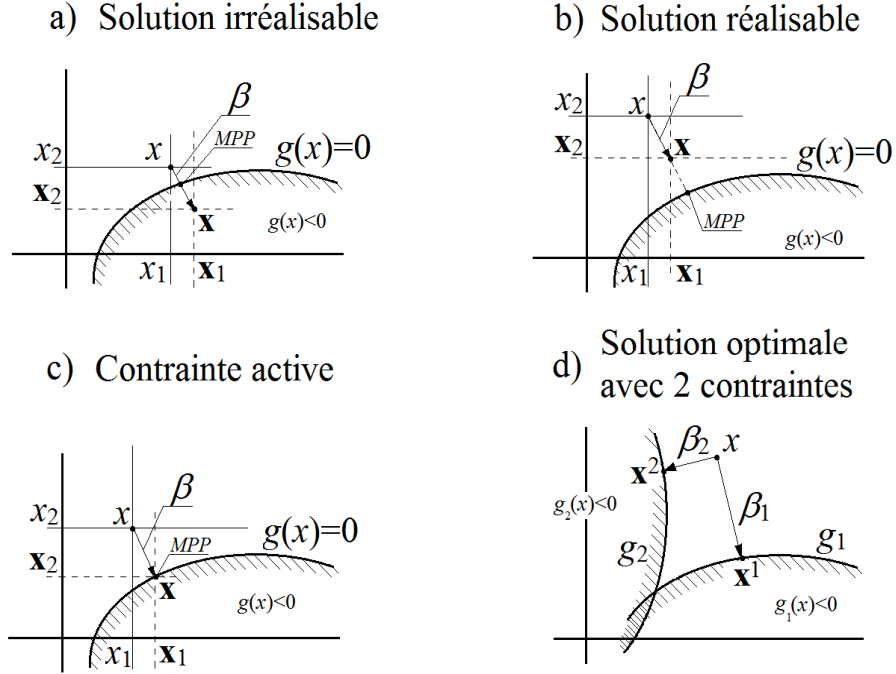
#### 4.3.1 Description de l'approche à boucle unique adoptée

Pour transformer les contraintes probabilistes en déterministes, nous adoptons la technique de l'espace de conception fiable (RDS) proposée par [126]. La Fig. 4.2 présente l'essentiel de cette technique. Pour la simplicité de l'explication, nous supposons qu'il n'y a pas de variables d'optimisation déterministes et de paramètres aléatoires, et le système est considéré comme étant en sécurité si  $g(x) > 0$ , et en danger si  $g(x) < 0$ . L'idée principale de la technique RDS est d'évaluer la fonction d'état limite ( $g(x)$ ) de la contrainte probabiliste à un point, on va dire  $\mathbf{x}$ , qui est loin du point de conception actuel  $x$  avec la distance équivalente à l'indice de fiabilité dans l'espace normal, et dans la direction du MPP. Si  $g(\mathbf{x}) < 0$ , cela signifie que, dans l'espace normal, la distance entre le MPP et  $x$  est inférieure à  $\beta$  qui signifie que  $P(g(x) < 0) > P_{fa}$ . Par conséquent,  $x$  est considéré comme une solution irréalisable (voir Fig. 4.2a. Solution irréalisable). Si  $g(\mathbf{x}) > 0$ , la distance entre  $x$  et le MPP dans l'espace normal est supérieure à  $\beta$  ce qui signifie  $P(g(x) < 0) < P_{fa}$ . Le point de conception  $x$  est considéré comme réalisable (voir Fig. 4.2b. Solution réalisable). Dans le cas de  $g(\mathbf{x}) = 0$ ,  $\mathbf{x}$  est au MPP, et la distance entre  $x$  et le MPP est égale à  $\beta$ , donc  $P(g(x) < 0) = P_{fa}$ , ce qui signifie que la contrainte probabiliste est active (voir Fig. 4.2c. de contrainte active). La Fig. 4.2d présente la solution optimale de  $\min(x)$  avec deux contraintes de fiabilité avec des indices de fiabilité différents.

Par conséquent, nous définissons d'abord l'indice de fiabilité correspondant de chaque contrainte probabiliste comme suit :

$$\beta^i = -\Phi^{-1}(P_{fai}) \quad (4-6)$$





**Fig. 4.2** Schématisation du principe de l'espace fiable.

Ensuite, nous calculons  $\mathbf{x}$  et  $\mathbf{p}$  par la transformation suivante :

$$\mathbf{x}_j^i = \mu_{x_j} - \alpha_{x_j}^i \sigma_{x_j} \beta^i \quad (4-7)$$

$$\mathbf{p}_j^i = \mu_{p_j} - \alpha_{p_j}^i \sigma_{p_j} \beta^i \quad (4-8)$$

où l'indice  $i$  correspond au nombre de contraintes et  $j$  représente la variable,  $\mu$  et  $\sigma$  sont respectivement la moyenne et l'écart type. Les cosinus directeurs ( $\alpha$ ) sont calculés comme suit :

$$\alpha_{x_j}^i = \frac{\left( \sigma_{x_j} \frac{\partial g_i(d, x, p)}{\partial x_j} \right)}{\sqrt{\sum_k^{NX} \left( \sigma_{x_k} \frac{\partial g_i(d, x, p)}{\partial x_k} \right)^2 + \sum_k^{NP} \left( \sigma_{p_k} \frac{\partial g_i(d, x, p)}{\partial p_k} \right)^2}} \quad (4-9)$$

$$\alpha_{p_j}^i = \frac{\left( \sigma_{p_j} \frac{\partial g_i(d, x, p)}{\partial p_j} \right)}{\sqrt{\sum_k^{NX} \left( \sigma_{x_k} \frac{\partial g_i(d, x, p)}{\partial x_k} \right)^2 + \sum_k^{NP} \left( \sigma_{p_k} \frac{\partial g_i(d, x, p)}{\partial p_k} \right)^2}} \quad (4-10)$$

Par conséquent, le problème de RBDO défini dans l'Eq. (4-1) devient un problème d'optimisation déterministe comme suit :

$$\begin{cases} \text{Minimiser } f(d, \mu_x) \\ \text{Tel que :} \\ g_i(d, \mathbf{x}^i, \mathbf{p}^i) \geq 0, i = 1 \dots m \\ h_i(d, \mu_x, \mu_p) \leq 0, i = m + 1 \dots n \\ L_{dj} < d_j < U_{dj}, j = 1 \dots ND \\ L_{xj} < \mu_{xj} < U_{xj}, j = 1 \dots NX \end{cases} \quad (4-11)$$

La transformation de  $x$  et  $p$  présentée ci-dessus (Eq. (4-7 et 4-8)), n'est valable que si les paramètres précédents suivent une loi de probabilité normale. Dans le cas où une variable suit une loi de probabilité différente, nous utilisons la transformation de Rosenblatt [129]. Par conséquent, la transformation  $x \rightarrow \mathbf{x}$  devient :

$$\mathbf{x}_j^i = CDF_j^{-1}(-\beta^i \tilde{\alpha}_{xj}^i) \quad (4-12)$$

où

$$\tilde{\alpha}_{xj}^i = \frac{\left( \hat{\sigma}_{xj} \frac{\partial g_i(d, x, p)}{\partial x_j} \right)}{\sqrt{\sum_k^{NX} \left( \hat{\sigma}_{xk} \frac{\partial g_i(d, x, p)}{\partial x_k} \right)^2 + \sum_k^{NP} \left( \hat{\sigma}_{pk} \frac{\partial g(d, x, p)}{\partial p_k} \right)^2}} \quad (4-13)$$

et

$$\hat{\sigma}_{xj} = \frac{\phi\left(\Phi^{-1}\left[CDF_j(x_j)\right]\right)}{PDF_j(x_j)} \quad (4-14)$$

$PDF$  est la fonction de densité de probabilité de la variable  $x$ ,  $\phi$  est la fonction de densité de probabilité normale. Les mêmes formules sont utilisées en cas de paramètre aléatoire  $p$ , nous changeons  $x$  à  $p$ .

Il est important de noter qu'en réalité, le terme  $\hat{\sigma}$  représente la matrice Jacobienne inverse de la transformation de Rosenblatt. Pour plus de détails mathématiques de la technique RDS, le lecteur pourra consulter cette référence [126].

### 4.3.2 Adaptation de dBA pour la résolution des problèmes d'optimisation sous contraintes

Tout d'abord, pour la commodité de la discussion et pour éviter toute confusion, nous supposons que  $y = [d_1, d_2, \dots, d_{ND}, x_1, x_2, \dots, x_{NX}]^T$  est un vecteur des variables de conception à la fois aléatoire et déterministe de dimension  $N = ND + NX$ . Pour résoudre le problème d'optimisation sous contraintes issu de la technique DRS (Eq. (4-11)), la méthode  $\varepsilon$ -CHT ( $\varepsilon$ -Constraint Handling technique) [93] a été adoptée afin d'assurer la satisfaction des contraintes. L'idée principale de cette méthode est de définir un niveau de comparaison  $\varepsilon$  comme un ordre de relation sur l'ensemble des paires  $(f(y), v(y))$  où  $v(y)$  est la fonction de violation des contraintes qui est définie comme suit :

$$v(y) = \sum_{j=1}^m \min\{0, g_j(d, y, p)\}^s + \sum_{j=1}^n \max\{0, h_j(d, y, p)\}^s \quad (4-15)$$

où  $y = [d_1, d_2, \dots, d_{ND}, x_1, x_2, \dots, x_{NX}]^T$ , et  $s$  est un nombre pair positif (dans cette étude,  $s = 2$  est utilisé).  $v(y)$  indique combien les contraintes ont été violées à un point  $y$ . La fonction de violation des contraintes  $v(y)$  a la propriété suivante:

$$\begin{cases} v(y) = 0 & (y \in \Omega) \\ v(y) > 0 & (y \notin \Omega) \end{cases} \quad (4-16)$$

où  $\Omega$  est l'espace de conception fiable. Le niveau de comparaison  $\varepsilon$  est défini par un ordre lexicographique dans laquelle  $v(y)$  précède  $f(y)$ , parce que la satisfaction des contraintes est plus importante que la minimisation de la fonction objectif [93].

Considérons deux points  $y_1$  et  $y_2$  avec leurs valeurs de la fonction objectif et la violation des contraintes associées  $f_1, f_2$  et  $v_1, v_2$ , respectivement. Alors, pour tout  $\varepsilon$  ( $\varepsilon \geq 0$ ), les niveaux de comparaisons  $<_\varepsilon$  et  $\leq_\varepsilon$  entre  $(f_1, v_1)$  et  $(f_2, v_2)$  sont définis comme suit:

$$(f_1, v_1) <_\varepsilon (f_2, v_1) \Leftrightarrow \begin{cases} f_1 < f_2, & \text{si } v_1, v_2 \leq \varepsilon \\ f_1 < f_2, & \text{si } v_1 = v_2 \\ v_1 < v_2, & \text{sinon} \end{cases} \quad (4-17)$$

$$(f_1, v_1) \leq_\varepsilon (f_2, v_1) \Leftrightarrow \begin{cases} f_1 \leq f_2, & \text{si } v_1, v_2 \leq \varepsilon \\ f_1 \leq f_2, & \text{si } v_1 = v_2 \\ v_1 \leq v_2, & \text{sinon} \end{cases} \quad (4-18)$$

Le niveau  $\varepsilon$  est mis à jour jusqu'à ce que le compteur d'itération  $t$  atteigne une certaine valeur appelée itération de contrôle  $T_c$ . Après que le compteur d'itération dépasse  $T_c$ , le niveau  $\varepsilon$  est mis à zéro pour obtenir une solution sans violation de contrainte. La mise à jour de  $\varepsilon$  est la suivante :

$$\varepsilon^0 = \nu(y_\theta)$$

$$\varepsilon^t = \begin{cases} \varepsilon^0 (1 - (t/T_c))^{cp}, & 0 < t < T_c \\ 0, & t \geq T_c \end{cases} \quad (4-19)$$

où  $y_\theta$  est le  $\theta$ -ème chauve-souris et  $cp \in [2, 10]$ . Dans cette étude  $cp = 5$  est utilisé, et  $T_c = 0.95t_{max}$ .

#### Algorithme 4.1

RBDO par dBA.

---

1. Définir la fonction objectif
  2. Définir l'indice de fiabilité
  3. Définir  $r_0, r_\infty, A_0, A_\infty, f_{min}$  et  $f_{max}$ .
  4. Initialiser la population des chauves-souris  $L_i \leq y_i \leq U_i$  ( $i = 1, \dots, NB$ )
  5. Evaluer la fonction objectif  $f_i(y_i)$  et le taux de violation des contraintes  $\nu(y_i)$
  6. Sélectionner le  $\theta$ -ème chauve-souris et définir  $\varepsilon^0$
  7. Initialiser le taux d'impulsion  $r_i$  et l'intensité sonore  $A_i$  et  $w_i$
  8. *Tant que* ( $t \leq t_{max}$ )
  9.     *Pour*  $i = 1, \dots, NB$
  10.         Générer les fréquences
  11.         Sélectionner aléatoirement une chauve-souris ( $k \neq i$ )
  12.         Mise à jour de la position/solution
  13.         *Si* ( $rand > r_i$ )
  14.             Générer une solution locale
  15.             Mise à jour de  $w_i$
  16.         *Fin de Si*
  17.         *Pour*  $j = 1, \dots, N$
  18.             *Si*  $y_{i,j}^{t+1}$  est discret, arrondir à la plus proche valeur, *Fin de Si*
  19.         *Fin de Pour*
  20.         Evaluer la fonction objectif et le taux de violation de contrainte  $f(y_i^{t+1})$  et  $\nu(y_i^{t+1})$
  21.         *Si*  $\{rand < A_i\} \& \{(f(y_i^{t+1}), \nu(y_i^{t+1})) <_\varepsilon (f(y_i^t), \nu(y_i^t))\}$
  22.             Accepter la nouvelle solution
  23.             Augmenter  $r_i$
  24.             Réduire  $A_i$
  25.         *Fin de Si*
  26.         *Si*  $(f(y_i^{t+1}), \nu(y_i^{t+1})) <_\varepsilon (f(y^*), \nu(y^*))$
  27.             Mise à jour de la meilleure solution  $y^*$
  28.         *Fin de Si*
  30.         *Fin de Pour*
  31.     *Fin de Tant que*
  32. Traitement des résultats
-

Le mouvement des chauves-souris génère des valeurs réelles. Dans le cas d'optimisation à variables discrètes, nous arrondissons les solutions générées par le mouvement aux valeurs discrètes les plus proches. Cependant, le pseudocode de la nouvelle méthode de RBDO basé sur l'algorithme des chauves-souris directionnelles, est présenté dans l'algorithme 4.1.

#### 4.4 Exemples d'application et validation

##### 4.4.1 Exemple 1: résistance à l'impact latéral d'un véhicule

Cet exemple proposé par [149] a été largement utilisé pour tester l'efficacité et la précision des méthodes RBDO [125, 126, 150]. Il représente ce qui peut arriver à une voiture de tourisme quand elle est frappée sur le côté par un autre véhicule à environ 30 miles par heure soit 48,28 kilomètres par heure. Pour formuler le problème de RBDO, les auteurs [149] ont construit un modèle d'éléments finis pour simuler l'impact latéral. Après, en exécutant le modèle d'éléments finis pour certaines combinaisons des variables de conception, des surfaces de réponse en forme polynomiale ont été construites pour approximer la fonction objectif (de poids) et les contraintes (déviations et vitesses à des endroits différents du véhicule et du mannequin). Plus de détails de cet exemple peuvent être trouvés dans [149].

Le problème d'optimisation basé sur la fiabilité de la résistance à l'impact latéral du véhicule est comme suit :

$$\left\{ \begin{array}{l}
 \text{Minimiser } f = 1.98 + 4.90x_1 + 6.67x_2 + 6.98x_3 + 4.01x_4 + 1.78x_5 + 2.73x_7 \\
 \text{Tel que:} \\
 L_i \leq x_i \leq U_i, \quad i = 1, \dots, 7 \\
 P(F_{AL} \geq 1.01) \leq P_{fa} \\
 P(D_{low} \geq 32) \leq P_{fa} \\
 P(D_{middle} \geq 32) \leq P_{fa} \\
 P(D_{up} \geq 32) \leq P_{fa} \\
 P(VC_{low} \geq 0.32) \leq P_{fa} \\
 P(VC_{middle} \geq 0.32) \leq P_{fa} \\
 P(VC_{up} \geq 0.32) \leq P_{fa} \\
 P(F_{PS} \geq 4.0) \leq P_{fa} \\
 P(V_{B-Pillow} \geq 0.99) \leq P_{fa} \\
 P(V_{door} \geq 15.69) \leq P_{fa}
 \end{array} \right. \quad (4-20)$$

**Tableau 4.1**

Caractéristiques des variables d'optimisation et des paramètres aléatoires de l'exemple 1.

$x_i$	$L_i$	$U_i$	E.T.	$p_i$	Moyenne	E.T.
$x_1$	0.5	1.5	0.03	$p_1$	0.345	0.006
$x_2$	0.45	1.35	0.03	$p_2$	0.192	0.006
$x_3$	0.5	1.5	0.03	$p_3$	0.0	10.0
$x_4$	0.5	1.5	0.03	$p_4$	0.0	10.0
$x_5$	0.875	2.625	0.05			
$x_6$	0.4	1.2	0.03			
$x_7$	0.4	1.2	0.03			

où

$$F_{AL} = 1.16 - 0.3717x_2x_4 - 0.00931x_2p_3 - 0.484x_3p_2 + 0.01343x_6p_4$$

$$D_{low} = 46.36 - 9.9x_2 - 12.9x_1p_1 + 0.1107x_3p_3$$

$$D_{middle} = 33.86 + 2.95x_3 + 0.1792p_3 - 5.057x_1x_2 - 11.0x_2p_1 - 0.0215x_5p_3 - 9.98x_7p_1 + 22.0p_1p_2$$

$$D_{up} = 28.98 + 3.818x_3 - 4.2x_1x_2 + 0.0207x_5p_3 + 6.63x_6p_2 - 7.7x_7p_1 + 0.32p_2p_3$$

$$VC_{low} = 0.74 - 0.61x_2 - 0.163x_3p_1 + 0.001232x_3p_3 - 0.166x_7p_2 + 0.227x_2^2$$

$$VC_{middle} = 0.214 + 0.00817x_5 - 0.131x_1p_1 - 0.0704x_1p_2 + 0.03099x_2x_6 - 0.018x_2x_7 + 0.0208x_3p_1 \\ + 0.121x_3p_2 - 0.00364x_5x_6 + 0.0007715x_5p_3 - 0.0005354x_6p_3 + 0.00121p_1p_4$$

$$VC_{up} = 0.261 - 0.0159x_1x_2 - 0.188x_1p_1 - 0.019x_2x_7 + 0.0144x_3x_5 + 0.0008757x_5p_3 + 0.08045x_6p_2 \\ + 0.00139p_1p_4 + 0.00001575p_3p_4$$

$$F_{PS} = 4.72 - 0.5x_4 - 0.19x_2x_3 - 0.0122x_4p_3 + 0.009325x_6p_3 + 0.000191p_4^2$$

$$V_{B-Pillar} = 10.58 - 0.674x_1x_2 - 1.95x_2p_1 + 0.02054x_3p_3 - 0.0198x_4p_3 + 0.028x_6p_3$$

$$V_{door} = 16.45 - 0.489x_3x_7 - 0.843x_5x_6 + 0.0432p_2p_3 - 0.0556p_2p_4 - 0.000786p_4^2$$

Ce problème a sept (7) variables de conception aléatoires et 4 paramètres aléatoires. Les détails de ces paramètres figurent dans le Tableau 4.1. Toutes les quantités aléatoires suivent la loi de probabilité normale.

Pour examiner l'efficacité et la robustesse de la RBDO-dBA qu'on a proposé, trois autres méthodes d'RBDO ont été mises en œuvre en utilisant la même méthode avec d'autres algorithmes d'optimisation, à savoir, l'algorithme des chauves-souris standard (BA), l'algorithme d'optimisation

par essaim de particules (PSO) et l'algorithme d'évolution différentielle (DE). Les paramètres de réglage de chaque algorithme sont les suivants :

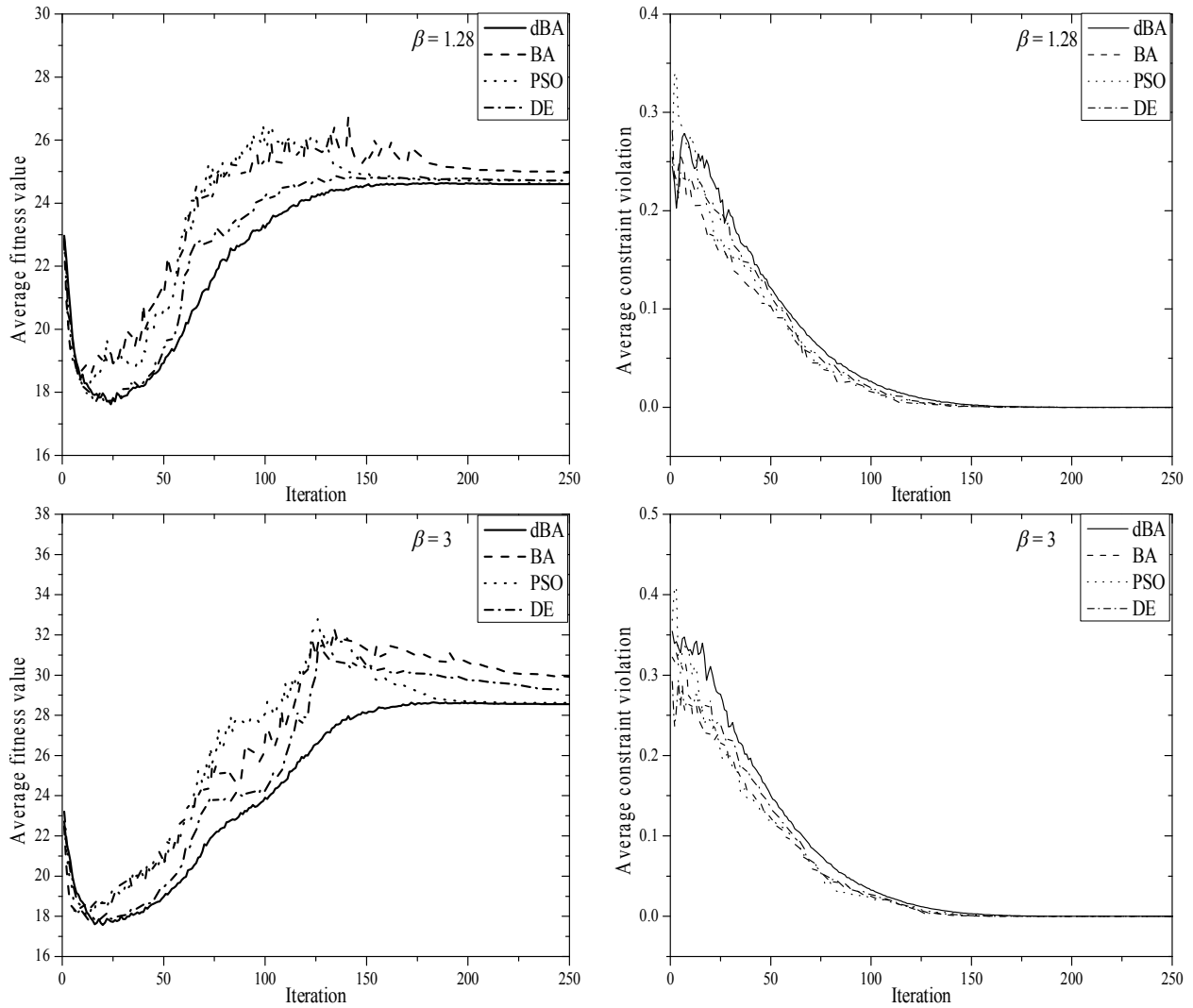
- **dBA:** nous recommandons les paramètres suivants  $r_0 = 0.1$ ,  $r_\infty = 0.7$ ,  $A_0 = 0.9$ ,  $A_\infty = 0.6$ ,  $f_{min} = 0$  et  $f_{max} = 2$ .
- **BA:** l'algorithme de chauve-souris normé tel qu'il est décrit dans[48] avec  $r_0 = 0.1$ ,  $A_0 = 0.9$ ,  $\alpha = \gamma = 0.9$ ,  $f_{min} = 0$  and  $f_{max} = 2$ .
- **PSO :** l'algorithme d'optimisation par essaim de particules classique a été considéré [151]. Le réglage des paramètres sont :  $c_1 = 1.5$ ,  $c_2 = 1.2$  et le coefficient d'inertie  $w$  est une fonction monotone décroissante de 0,9 à 0,4.
- **DE:** l'algorithme d'évolution différentielle classique [40] avec la stratégie "DE/rand/1/bin" est considéré. Les paramètres sont  $CR = rand[0.2, 0.9]$  et  $F = rand[0.4, 1]$ .

Pour une comparaison équitable, les paramètres communs sont considérés être la même (population  $NB= 50$ , le nombre des itérations  $t_{max} = 250$ ). Comme le dBA, BA, PSO et DE sont des algorithmes stochastiques, chacun a été exécuté 25 fois pour réaliser une comparaison statistique significative. Les résultats du processus de minimisation de l'exemple 1 avec des algorithmes différents pour deux valeurs de l'indice de fiabilité  $\beta$  sont présentés dans le Tableau 4.2. Comme on peut le voir, aucune des solutions obtenues par différents algorithmes n'a violé les contraintes. Les résultats obtenus par dBA sont meilleurs que ceux obtenus par les autres algorithmes en terme de précision. La faible valeur de l'écart type des résultats de dBA suggère que l'algorithme converge vers la même solution avec une faible erreur pour chaque essai par rapport aux autres algorithmes, ce qui signifie que dBA est plus robuste.

**Tableau 4.2**

Comparaison entre dBA, BA, PSO et DE pour minimisation de l'Exemple 1.

$\beta$	$f_{min}$	dBA	BA	PSO	DE
$\beta=1.28$	Min	<b>24.59968</b>	24.78268	<b>24.59968</b>	24.61810
	Médiane	<b>24.59975</b>	24.90586	24.59985	24.67582
	Max	<b>24.60003</b>	25.51112	24.95415	24.84464
	Moyenne	<b>24.59977</b>	24.96760	24.61679	24.71182
	E.T.	<b>0.000089</b>	0.154862	0.069798	0.083855
	Moyenne(v)	0	0	0	0
$\beta=3$	Min	<b>28.55267</b>	29.01287	28.55281	28.71363
	Médiane	<b>28.55766</b>	29.79713	28.57274	29.13727
	Max	<b>28.57933</b>	31.37449	28.85582	31.98255
	Moyenne	<b>28.55893</b>	29.90215	28.62164	29.24571
	E.T.	<b>0.006315</b>	0.708308	0.089498	0.615516
	Moyenne(v)	0	0	0	0



**Fig. 4.3** Evolution de la moyenne de 25 essais de la fonction objectif (gauche) et le taux de violation des contraintes (droite) des quatre algorithmes, dBA, BA, PSO et DE.

La Fig. 4.3 représente l'évolution de la moyenne des 25 essais du minimum de la fonction objectif et la valeur de la violation des contraintes, pour les quatre algorithmes. A partir des courbes de violation des contraintes, on observe que les valeurs sont élevées au début du processus d'itération, cela est causé par le niveau de comparaison- $\varepsilon$  qui permet une violation de contrainte admissible si  $\nu < \varepsilon$ . En conséquence, les algorithmes convergent vers une solution plus faible que l'optimum réel. Pendant que le processus d'itération avance, la valeur admissible de la violation des contraintes diminue (en raison de la diminution de la valeur  $\varepsilon$ ) ce qui affecte directement la fonction objectif, forçant l'algorithme à converger vers l'optimum réel sans violation de contrainte.



A la fin du processus d'itération, on constate qu'aucun algorithme a convergé vers une solution irréalisable, tandis que l'algorithme de chauve-souris directionnel converge vers une meilleure solution par rapport à l'autre algorithme dans les deux cas ( $\beta = 1.28$  et 3). De plus, nous avons également observé que l'écart entre les quatre algorithmes à la fin des processus d'itération augmente à mesure que l'indice de fiabilité augmente ( $\beta = 3$ ), en particulier pour BA et DE. Cela est dû à la réduction de l'espace en étant fiable de 6.59% pour  $\beta = 1.28$  à 0.27% pour  $\beta = 3$  (le pourcentage de l'espace fiable est obtenu par la simulation Monte Carlo avec  $10^5$  échantillonnages). En conclusion, dBA est plus efficace, plus fiable et robuste par rapport aux autres algorithmes.

**Tableau 4.3**

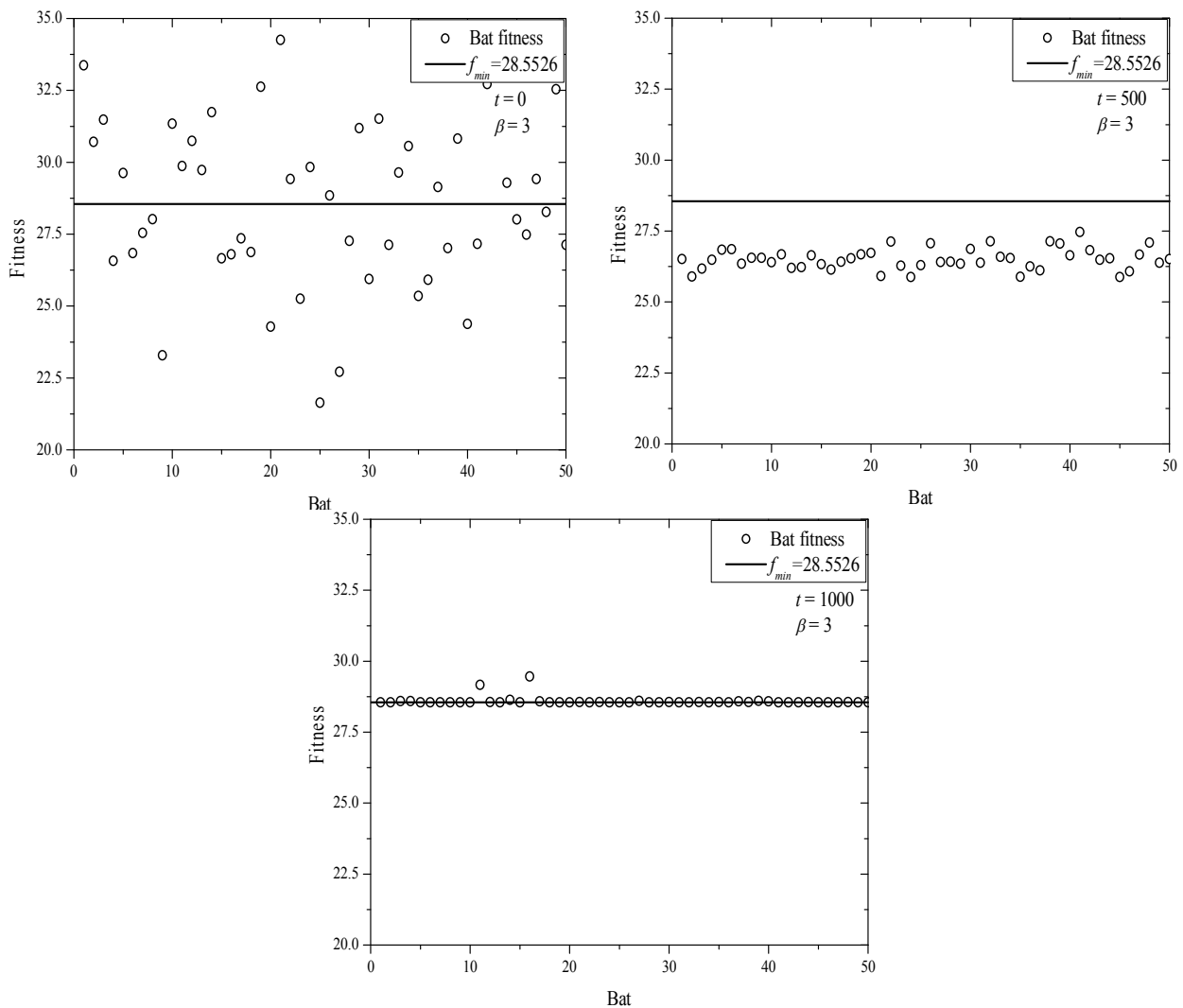
Effet du nombre des itérations et la taille de la population des chauves-souris sur la qualité des résultats d'optimisation de l'Exemple 1.

$t_{max}$		Population ( $NB$ )			
		25	50	75	100
250	Min	28.5526712	28.5529162	28.5527059	28.5527574
	Médiane	28.5576631	28.5560087	28.5541877	28.5538673
	Max	28.5793344	28.5792427	28.5634212	28.5584198
	Moyenne	28.5589327	28.5571123	28.5548233	28.5540514
	E.T.	0.00631511	0.00551702	0.00233401	0.00135970
500	Min	28.5526704	28.5526505	28.5526510	28.5526498
	Médiane	28.5542206	28.5531227	28.5528818	28.5527999
	Max	28.5736677	28.5562962	28.5553020	28.5555970
	Moyenne	28.5555682	28.5534176	28.5530909	28.5529676
	E.T.	0.00422049	0.00088993	0.00060732	0.00057967
750	Min	28.5526498	28.5526497	28.5526514	28.5526499
	Médiane	28.5529929	28.5526902	28.5526819	28.5526587
	Max	28.5756670	28.5537189	28.5530968	28.5527439
	Moyenne	28.5545778	28.5528753	28.5527251	28.5526725
	E.T.	0.00462175	0.00033820	0.00009822	0.00002593
1000	Min	28.5526497	28.5526497	28.5526497	28.5526497
	Médiane	28.5527924	28.5526702	28.5526571	28.5526530
	Max	28.5570547	28.5547097	28.5528296	28.5527264
	Moyenne	28.5531349	28.5528132	28.5526803	28.5526578
	E.T.	0.00091170	0.00043017	0.00004900	0.00001551

Pour analyser l'effet de la population des chauves-souris et le nombre des itérations sur la convergence de l'algorithme, dBA a été exécuté 25 fois avec des paramètres différents de  $NB$  et  $t_{max}$ . L'indice de fiabilité a été fixé à  $\beta = 3$  et les résultats statistiques sont présentés dans le Tableau 4.3. Comme on peut le voir, l'augmentation de la taille de la population et le nombre des itérations donne des résultats plus précis avec un écart type faible. Par exemple, pour  $N = 50$  and  $t_{max} = 1000$ , l'écart

type obtenu de 25 essais est  $E.T. = 4.3 \cdot 10^{-4}$ . Cela signifie qu'il y a une forte probabilité qu'une seule exécution de l'algorithme converge vers l'optimum avec une valeur d'erreur négligeable. En outre, nous observons que la valeur minimale de la fonction objectif la mieux obtenue avec différents réglages est  $f_{min} = 28.5526497$ . Nous croyons que c'est le plus petit minimum qu'on puisse avoir connu jusqu'à présent de l'exemple 1 pour  $\beta = 3$ .

Pour les paramètres suivants :  $N = 50$ ,  $t_{max} = 1000$  et  $\beta = 3$ , la Fig. 4.4 présente la valeur de la fonction objectif de chaque chauve-souris de l'essaim à  $t = 0$  (position initiale),  $t = 500$  (50% du processus d'itération) et  $t = 1000$  (fin du processus d'itération).



**Fig. 4.4** Les valeurs de la fonction objectif pour chaque chauve-souris pour  $t = 0, 500$ , et  $1000$  (Exemple 1).

Comme on le voit, dans la phase initiale ( $t = 0$ ), les chauves-souris sont distribuées au hasard dont la valeur de la fonction objectif associée à chacune d'entre elles est différente de l'autre. Au fur à mesure que les itérations passent ( $t = 500$ ), l'écart entre les valeurs de la fonction objectif de toutes les chauves-souris diminue et approche le minimum réel avec des valeurs inférieures (cela est dû au niveau de comparaison  $-\varepsilon$  qui tolère une violation de contrainte admissible à 50% du processus d'itération). À la fin du processus d'itération,  $\varepsilon$  tend vers 0 et aucune violation des contraintes n'est admissible. La majorité des chauves-souris convergent vers l'optimum réel, et ceux qui ne sont pas dans le voisinage, elles sont dans l'espace fiable.

**Tableau 4.4**

Comparaison entre la meilleure, la médiane et la pire solution obtenues par dBA avec les méthodes classiques pour  $\beta = 1.28$  (Exemple1).

	RBDO-dBA			RDS*	Découplé+ FORM <sup>#</sup>	Découplé+ SORM <sup>#</sup>
	Meilleure	Médiane	Pire			
$x_1$	0.5	0.5	0.5000001	0.5	0.5	0.5
$x_2$	1.3089077	1.3089077	1.3089077	1.3092	1.3091	1.3089
$x_3$	0.5	0.5	0.5	0.5	0.5	0.5
$x_4$	1.3216346	1.3216346	1.3216346	1.3223	1.3229	1.4012
$x_5$	0.875	0.875	0.875	0.895	0.875	0.875
$x_6$	1.2	1.2	1.2	1.2	1.2	1.2
$x_7$	0.4	0.4	0.4	0.4	0.4	0.4
$f_{min}$	24.5996814	24.5996814	24.5996816	24.6043	24.6060	24.9190
$\mathcal{U}(x)$	0	0	0	--	--	--

\* Les Résultats sont de [126]; # Les Résultats sont de [150].

**Tableau 4.5**

Comparaison entre la meilleure, la médiane et la pire solution obtenues par dBA avec les méthodes classiques pour  $\beta = 3$  (Exemple1).

	RBDO-dBA			DLP/PMA <sup>§</sup>	SLA <sup>§</sup>	RDS*	Découplé+ FORM <sup>#</sup>
	Meilleure	Médiane	Pire				
$x_1$	0.8008490	0.8008517	0.8009404	0.9436	0.81	0.8008	0.8846
$x_2$	1.35	1.35	1.3499648	1.35	1.35	1.35	1.35
$x_3$	0.7133922	0.7133927	0.7134107	0.9127	0.7277	0.7134	0.8254
$x_4$	1.5	1.5	1.5	0.9913	1.5	1.5	1.5
$x_5$	0.875	0.875	0.875	0.9026	0.875	0.875	0.921
$x_6$	1.2	1.2	1.2	1.2	1.2	1.2	1.2
$x_7$	0.4	0.4	0.4	0.4	0.4	0.4	0.4
$f_{min}$	28.5526497	28.5526675	28.5529924	28.6528	28.6977	28.5526	29.827
$\mathcal{U}(x)$	0	0	0	--	--	--	--

\* Les Résultats sont de [126]; # Les Résultats sont de [150]; § Les Résultats sont de [125]

Pour analyser la précision de la méthode proposée, les Tableaux 4.4 et 4.5 présentent une comparaison avec la littérature de la meilleure, la médiane et la pire solution optimale de 25 exécutions de dBA avec  $N = 50$  et  $t_{max} = 1000$ , et pour  $\beta = 1.28$  et 3. Comme on peut le voir, les trois solutions (la meilleure, la médiane et la pire) sont très précises, et que dBA a réalisé de meilleurs résultats que les méthodes classiques. Dans le cas de  $\beta = 3$ , les résultats obtenus par notre méthode sont quasi-exacts avec ceux de [126] obtenus avec la méthode de l'espace de conception de fiabilité, et que nous pensons que c'est la meilleure solution connue à ce jour de ce problème dans la littérature.

#### 4.4.2 Exemple 2 : problème mathématique

Considérons le problème de RBDO mathématique suivant :

$$\left\{ \begin{array}{l} \text{Minimiser } f = x_1 + x_2 \\ \text{Tel que} \\ 0.1 \leq x_i \leq 10, \quad i = 1, 2; \quad \sigma_1, \sigma_2 = 0.3 \\ P\left(g_1 = \frac{x_1^2 x_2}{20} - 1 < 0\right) \leq P_{fa} \\ P\left(g_2 = \frac{(x_1 + x_2 - 5)^2}{30} + \frac{(x_1 - x_2 - 12)^2}{120} - 1 < 0\right) \leq P_{fa} \\ P\left(g_3 = \frac{80}{x_1^2 + 8x_2 + 5} - 1 < 0\right) \leq P_{fa} \end{array} \right. \quad (4-21)$$

Ce problème mathématique non linéaire est un exemple très connu, utilisé pour valider les méthodes d'RBDO lorsque les variables ne suivent pas la loi de distribution normale (gaussienne). Nous fixons  $NB = 10$  et  $t_{max} = 100$ . Dans le premier test, nous considérons que les variables sont gaussiennes et  $\beta = 3$ . La Fig. 4.6 représente le comportement de l'essaim des chauves-souris dans l'espace de recherche. Elle représente les positions des chauves-souris à l'état initial ( $t = 0$ ), à 50% du processus d'itération ( $t = 50$ ) et à la fin ( $t = 100$ ). Au fur à mesure que les itérations passent, les chauves-souris ont tendance à se rassembler autour de la solution optimale qui est loin des contraintes déterministes ( $g_i(x) = 0$ ) avec au moins  $\beta$  dans l'espace normal. Le Tableau 4.6 présente une comparaison de la meilleure, la médiane et la pire solution obtenue par dBA avec les méthodes classiques. Comme on peut voir, avec un faible nombre de chauves-souris et d'itérations, dBA converge vers une solution optimale autant que les méthodes classiques.

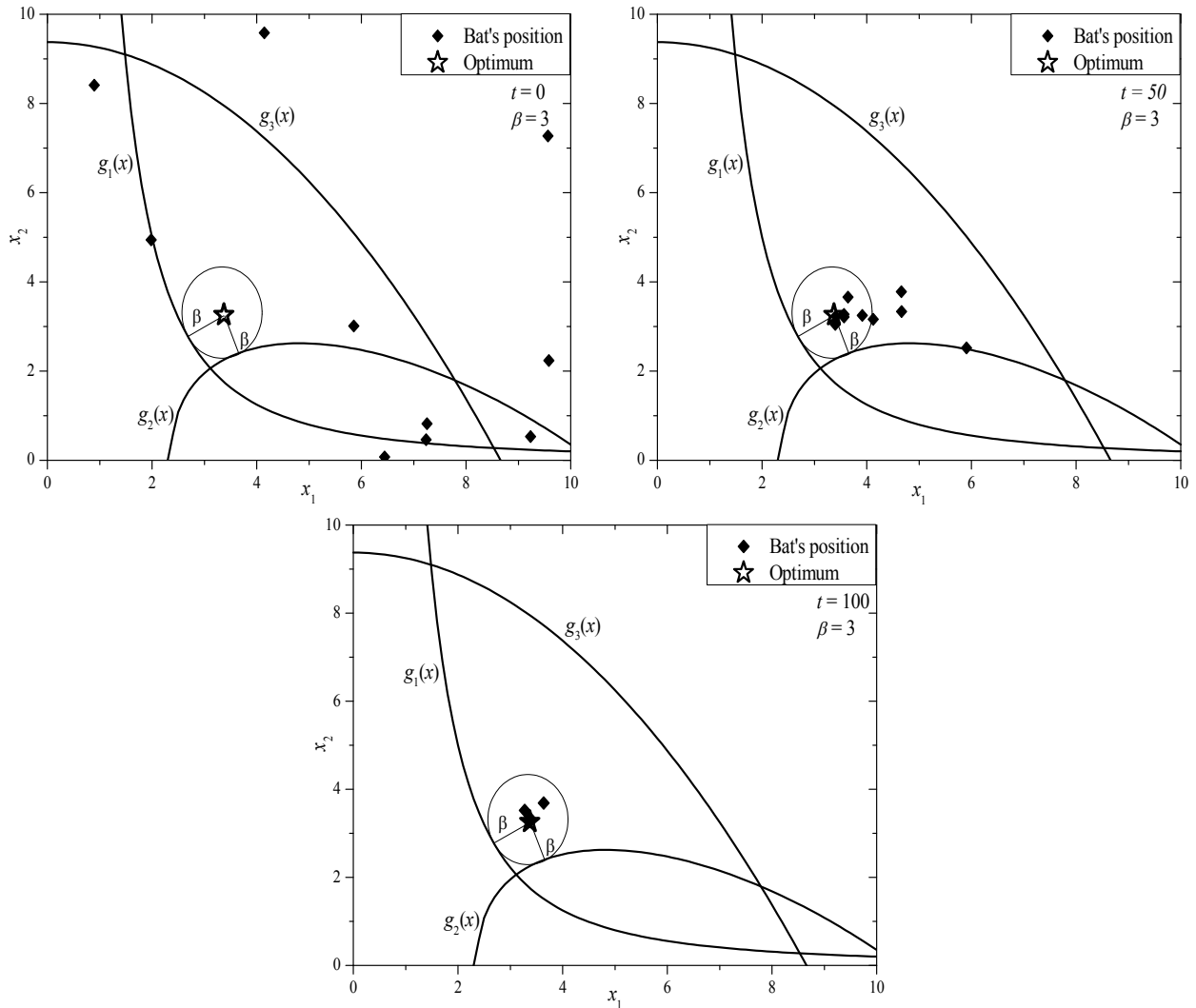


Fig. 4.5 Positions des chauves-souris dans l'espace de recherche au cours de l'optimisation à  $t = 0$ , 50 et 100 (Exemple 2).

Tableau 4.6

Comparaison entre la meilleure, la médiane et la pire solution obtenues par dBA avec les méthodes classiques (Exemple2).

	RBDO-dBA			PBDA <sup>†</sup>	DLP/PMA <sup>§</sup>	SORA <sup>#</sup>	SLA <sup>§</sup>	RDS <sup>*</sup>
	Meilleure	Médiane	Pire					
$x_1$	3.440555	3.438235	3.411155	3.4407	3.4391	3.4409	3.4391	3.4406
$x_2$	3.280018	3.285202	3.348766	3.2895	3.2866	3.2909	3.2864	3.2800
$f_{min}$	6.720573	6.723437	6.759921	6.7266	6.7257	6.7318	6.7255	6.7205

\* Les Résultats sont de[126]; § Les Résultats sont de[125]; † Les Résultats sont de[152]; # Les Résultats sont de[123]

**Tableau 4.7**

Résultats d'optimisation de l'Exemple 2 pour différents indices de fiabilité et lois de probabilité.

PDF	$\beta$	RBDO-dBA			RIA*	PMA*	KKT*	SLA*	SORA*	SAP*
		$x_1$	$x_2$	$f_{min}$						
$x_1$ : Normal $x_2$ : Normal	2	3.2953877	2.8959867	6.1913744	6.1923	6.1923	6.1923	6.1920	6.1923	6.1926
	3	3.4405576	3.2799744	6.7205320	6.7257	6.7251	8.9382	6.7756	6.7251	6.7261
	4	3.6115594	3.6417280	7.2532874	7.2683	7.2683	7.2683	7.2680	7.2683	7.2685
$x_1$ : Gumbel $x_2$ : Gumbel	2	3.2579776	2.7366447	5.9946222	6.0101	6.0101	n.c.	6.0047	6.0101	6.0103
	3	3.3136005	2.9440893	6.2576898	n.c.	6.2904	n.c.	6.2776	6.2904	n.c.
	4	3.3695310	3.1043223	6.4738533	n.c.	6.4340	n.c.	6.3770	6.4341	n.c.
$x_1$ : LogNormal $x_2$ : Normal	3	3.3753203	3.2524384	6.6277587	DLP/PMA <sup>#</sup>		SLA <sup>#</sup>			
					6.8903		6.67			

\*Les Résultats sont de[134]; Les Résultats sont de[125]; n.c.: non convergent

Pour améliorer la précision, nous fixons  $NB = 50$  et  $t_{max} = 1000$ . Pour les 25 essais, toutes les solutions convergentes vers le même optimum avec un écart-type inférieur à  $10^{-10}$ . Les résultats d'une seule exécution avec le réglage précédent sont présentés dans le Tableau 4.7 pour différentes valeurs d'indice de fiabilité et loi de probabilité des variables aléatoires. Contrairement à la RIA, KKT et SAP qui ont échoué à converger quand  $x_1$  et  $x_2$  suivent la distribution de Gumbel, dBA converge vers une solution réalisable, et les résultats sont quasi-exacts avec la littérature.

#### 4.4.3 Exemple 3 : conception d'un réducteur de vitesse

Considérons le problème suivant de conception d'un réducteur de vitesse :

$$\left\{ \begin{array}{l}
 \text{Minimiser } f = 0.7854x_1d_1^2 (3.3333d_2^2 + 14.9334d_2 - 43.0934) \\
 \quad - 1.5079x_1(x_4^2 + x_5^2) + 7.477(x_4^3 + x_5^3) \\
 \quad + 0.7854(x_2x_4^2 + x_3x_5^2) \\
 \text{Tel que:} \\
 P(g_i(d, x, p) < 0) \leq P_{fai}, \quad i = 1, \dots, 10 \\
 g_{11}(d, x, p) \geq 0 \\
 L_{d_i} \leq d_i \leq U_{d_i} \\
 L_{x_i} \leq x_i \leq U_{x_i}
 \end{array} \right. \quad (4-22)$$

où

$$g_1(d, x, p) = 1 - (p_1/x_1 d_1^2 d_2)$$

$$g_2(d, x, p) = 1 - (p_2/x_1 d_1^2 d_2^2)$$

$$g_3(d, x, p) = 1 - (p_3 x_2^3 / x_4^4 d_1 d_2)$$

$$g_4(d, x, p) = 1 - (p_4 x_3^3 / x_5^4 d_1 d_2)$$

$$g_5(d, x, p) = 1 - \frac{0.5 \sqrt{(p_6 x_2 / d_1 d_2)^2 + p_7}}{x_4^3 p_5 p_8}$$

$$g_6(d, x, p) = 1 - \frac{0.5 \sqrt{(p_6 x_3 / d_1 d_2)^2 + p_9}}{x_5^3 p_{10} p_8}$$

$$g_7(d, x, p) = 1 - (0.5 p_{11} d_1 / x_1)$$

$$g_8(d, x, p) = 1 - (x_1 / p_{12} d_1)$$

$$g_9(d, x, p) = 1 - (p_{13} x_4 + p_{15} / 2 x_2)$$

$$g_{10}(d, x, p) = 1 - (p_{14} x_5 + p_{15} / 2 x_3)$$

$$g_{11}(d, x, p) = 1 - (d_1 d_2 / 80)$$

Ce problème a été proposé pour la première fois sous forme déterministe [153], puis il a été transformé à un problème d'RBDO. De nombreux travaux [154-156] qui traitaient cet exemple supposent que toutes les variables de conception sont aléatoires avec une variance fixe, et les contraintes sont toutes probabilistes. Nous préférons utiliser la formulation proposée par [157] en raison des points suivants : 1) deux des variables de conception sont déterministes, tandis que le reste est aléatoire ; 2) les écarts types des variables de conception aléatoires sont dépendantes de la moyenne (elles varient lors de l'optimisation) ; 3) contraintes mixtes, dix sont probabilistes et une déterministe. Ces trois points augmentent la complexité et la non-linéarité du problème. Toutes les variables et les paramètres de conception aléatoires sont censés suivre la distribution normale. Les caractéristiques de ces paramètres figurent dans le Tableau 4.8. Pour plus de détails sur le problème, voir [153, 157].

**Tableau 4.8**

Caractéristiques des variables d'optimisation et paramètres de l'Exemple 3.

Variable	$L_{di}$	$U_{di}$	Variable	Moyenne	E.T.	Variable	Moyenne	E.T.	
$d_1$	0.7	0.8	$p_1$	27.0	2.7	$p_9$	1.58e+08	1.58e+07	
$d_2$	17	28	$p_2$	397.5	39.8	$p_{10}$	850	34	
Variable	$L_{xi}$	$U_{xi}$	C.d.V.	$p_3$	1.93	0.0965	$p_{11}$	5.0	0.25
$x_1$	2.6	4.2	0.05	$p_4$	1.93	0.0965	$p_{12}$	12.0	0.6
$x_2$	7.0	8.3	0.05	$p_5$	1100.0	110.0	$p_{13}$	1.5	0.75
$x_3$	7.0	9.3	0.05	$p_6$	745	74.5	$p_{14}$	1.1	0.11
$x_4$	2.9	3.95	0.02	$p_7$	1.69e+07	1.69e+06	$p_{15}$	1.9	0.19
$x_5$	5.0	6.0	0.02	$p_8$	0.1	0.005			

**Tableau 4.9**

Résultats d'optimisation de l'Exemple 3.

Variable	RBDO-dBA			SORA*
	Meilleure	Médiane	Pire	
$d_1$	0.7	0.7	0.7	0.7
$d_2$	17	17	17	17
$x_1$	3.860190	3.860190	3.860190	3.8619
$x_2$	7	7	7	7
$x_3$	7	7	7	7
$x_4$	2.932511	2.932511	2.932513	2.9326
$x_5$	5	5	5	5
$f_{min}$	2856.547	2856.547	2856.547	2857.27
Moyenne( $f_{min}$ ) =	2856.547		E.T.( $f_{min}$ ) = 9.62e-05	

\* Les résultats sont de [157].

Comme dans les exemples précédents, nous fixons  $NB = 50$  et  $t_{max} = 1000$  pour un maximum de précision et nous exécutons l'algorithme 25 fois. La probabilité de défaillance considérée dans cet exemple est  $P_f = 0.05$  ( $\beta \approx 1.644$ ). Les résultats sont présentés dans le Tableau 4.9. La dernière ligne présente la moyenne et l'écart type de la fonction objectif pour 25 essais. La faible valeur de l'écart type indique que toutes les exécutions ont convergé à la même solution. Comme on peut voir aussi que la différence entre le pire des solutions, la médiane et la meilleure solution est négligeable. La comparaison avec les résultats de [157] qui sont obtenus par une méthode de SORA améliorée montre que dBA converge vers une meilleure solution.



**Tableau 4.10**

Valeurs des contraintes, indice de fiabilité et la probabilité de défaillance pour chaque contrainte de la meilleure solution obtenue par dBA de l'Exemple 3.

Contrainte	Valeur de la contrainte	FORM		SORM				$P_{f-MCS}$
		$\beta$	$P_f$	$\beta_{Breitung}$	$P_{f-Breitung}$	$\beta_{Tvedt}$	$P_{f-Tvedt}$	
$g_1$	0.1603	<b>1.644</b>	<b>0.05000</b>	<b>1.644</b>	<b>0.05000</b>	<b>1.644</b>	<b>0.05000</b>	<b>0.05040</b>
$g_2$	0.2728	3.089	0.00100	3.089	0.00100	3.089	0.00100	0.00092
$g_3$	0.2478	<b>1.644</b>	<b>0.05000</b>	1.651	0.04938	1.653	0.04920	0.04850
$g_4$	0.9110	15.56	0.00000	15.57	0.00000	15.75	0.00000	0.00000
$g_5$	0.2548	2.058	0.01978	2.043	0.02051	2.041	0.02060	0.02605
$g_6$	0.4091	5.127	1.47e-07	5.116	1.56e-07	5.116	1.56E-07	0.00000
$g_7$	0.5467	9.957	0.00000	9.957	0.00000	9.957	0.00000	0.00000
$g_8$	0.7748	15.12	0.00000	15.12	0.00000	15.12	0.00000	0.00000
$g_9$	0.5501	3.310	0.00047	3.311	0.00046	3.311	0.00046	0.00033
$g_{10}$	0.4714	7.145	0.00000	7.146	0.00000	7.146	0.00000	0.00000
$g_{11}$	0.8513	--	--	--	--	--	--	--

Le Tableau 4.10 présente la valeur de violation des contraintes, l'indice de fiabilité et la probabilité de défaillance de chaque contrainte calculée avec des méthodes différentes, à savoir, FORM [11, 12], deux variantes de SORM (Breitung [13] et Tvedt [14]), et la simulation de Monte Carlo (MCS) de la meilleure solution. Les résultats des simulations de Monte Carlo sont obtenus avec  $10^5$  échantillons. D'après les résultats de FORM, deux contraintes sont actives ( $g_1$  et  $g_3$ ) où  $P_f = 0.05$ . En utilisant l'approche de SORM, seule la première contrainte est active. La Simulation de Monte Carlo montre que la première contrainte a moins de 1% de violation. Ceci est dû au fait que l'approche à boucle unique a une certaine similitude avec FORM dans la linéarisation de la fonction de l'état limite. Nous pouvons dire que l'approche à boucle unique a le même degré de précision que FORM. La contrainte qui a la probabilité de défaillance est égale à 0, signifie que la vraie probabilité est inférieure à  $10^{-10}$ . De l'analyse de la probabilité de défaillance des contraintes probabilistes et la violation de la contrainte déterministe, nous pensons que la solution présentée dans le Tableau 9 est la meilleure solution connue à ce jour de cet exemple sous la forme proposée par [157].

#### 4.4.4 Exemple 4 : poutre soudée

Dans cet exemple, nous considérons l'optimisation du coût de fabrication d'une poutre soudée sous contraintes probabilistes [154-156]. Cet exemple comporte quatre variables de conception telle qu'elles sont représentées sur la Fig. 4.6, et cinq contraintes probabilistes qui sont liées à des grandeurs mécaniques, comme la contrainte de flexion, contrainte de cisaillement, le flambement et la flèche. Le problème de RBDO est formulé comme suit :

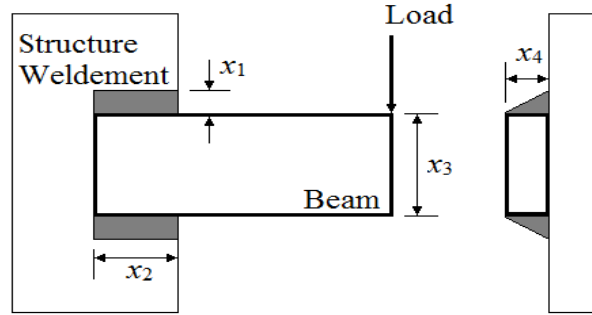


Fig. 4.6. La structure de la poutre soudée.

$$\left\{ \begin{array}{l} \text{Minimiser } f = c_1 x_1^2 x_2 + c_2 x_3 x_4 (p_2 + x_2) \\ \text{Tel que} \\ P(g_i(x, p) < 0) \leq \Phi(-\beta_{ai}), i = 1, \dots, 5 \\ 3.175 \leq x_1 \leq 50.8; \quad 0 \leq x_2 \leq 254 \\ 0 \leq x_3 \leq 254; \quad 0 \leq x_4 \leq 50.8 \\ \beta_{ai} = 3 \Leftrightarrow P_{fai} \cong 0.00135, \quad i = 1, \dots, 5 \\ \sigma_1, \sigma_2 = 0.1693 \\ \sigma_3, \sigma_4 = 0.0107 \end{array} \right. \quad (4-23)$$

où

$$g_1(x, p) = 1 - (\tau(x, p) / p_6)$$

$$g_2(x, p) = 1 - (s(x, p) / p_7)$$

$$g_3(x, p) = 1 - (x_1 / x_4)$$

$$g_4(x, p) = 1 - (\delta(x, p) / p_5)$$

$$g_5(x, p) = (P_c(x, p) / p_1) - 1$$

et

$$\tau(x, p) = \sqrt{t_1^2 + 2t_1^2 t_2 x_2 / 2R + t_2^2}$$

$$t_1 = p_1 / \sqrt{2} x_1 x_2$$

$$t_2 = MR / J$$

$$M = p_1 (p_2 + 0.5x_2)$$

$$R = 0.5\sqrt{x_2^2 + (x_1 + x_3)^2}$$

$$J = \sqrt{2}x_1x_2 \left( (x_2^2 / 12) + (x_1 + x_3)^2 / 4 \right)$$

$$s(x, p) = 6p_1p_2 / x_3^2x_4$$

$$\delta(x, p) = 4p_1p_2^3 / p_3x_3^3x_4$$

$$P_c(x, p) = \frac{4.013x_3x_4^3\sqrt{p_3p_4}}{6p_2^2} \left( 1 - \frac{x_3}{4p_2} \sqrt{\frac{p_3}{p_4}} \right)$$

Les caractéristiques des paramètres aléatoires sont énumérées au Tableau 4.11.

**Tableau 4.11**

Caractéristique des variables et paramètres d'optimisation de l'Exemple 4.

Paramètres	$\mu$	Distribution	C.O.V	$\sigma$
$p_1$ Chargement	26680.00	Log-Normal	0.10	2668.0
$p_2$ Longueur de la poutre	335.56	Normal	0.05	16.778
$p_3$ Module de Young	206850.00	Log-Normal	0.03	6205.5
$p_4$ Module de cisaillement	82740.00	Log-Normal	0.03	2482.2
$p_5$ Flèche admissible	6.35	Normal	0.05	0.3175
$p_6$ Contrainte de cisaillement admissible	93.77	Log-Normal	0.07	6.5639
$p_7$ Contrainte normale admissible	206.85	Log-Normal	0.07	14.4795
$c_1$ Cout du matériau de soudage	6.74E-05	Déterministe	--	--
$c_2$ Cout de l'acier	2.94E-06	Déterministe	--	--

En général, ce problème a été résolu sans tenir compte des incertitudes dans les paramètres du problème [154-156], tels que la charge, le module d'Young, la contrainte admissible, ...etc. Tous les paramètres ont été considérés comme déterministes, et leurs valeurs correspondent aux moyennes présentées dans le Tableau 4.11. Cette hypothèse réduit la complexité du problème et n'interprète pas la réalité. Les résultats de l'optimisation sont résumés dans le Tableau 4.12. La meilleure, la médiane et la pire solution de 25 exécutions de dBA avec  $NB = 50$  et  $t_{max} = 1000$ , sont comparées aux points optimaux obtenus par des méthodes différentes qui existent dans la littérature. Comme on peut le constater, les solutions obtenues par dBA ont une bonne concordance avec la littérature.

**Tableau 4.12**

Résultats d'optimisation de l'Exemple 4 avec variables continues et paramètres déterministes.

Méthode	$f$	$x_1$	$x_2$	$x_3$	$x_4$	Référence
RIA	2.5909	5.7328	200.749	210.597	6.2391	[155]
PMA	2.5886	5.7298	200.467	210.606	6.2384	
Moment	2.5895	5.7299	200.599	210.599	6.2389	
Moment + Kriging	2.5895	5.7299	200.602	210.598	6.2390	
RIA	2.59	5.733	200.7	210.6	6.239	[154]
PMA	2.72	5.750	219.8	210.7	6.260	
RIA+ Enveloppe	3.09	7.006	187.9	182.5	8.485	
PMA+ Enveloppe	3.15	5.183	198.5	234.2	7.313	
RIA	2.591	5.730	200.91	210.60	6.239	[156]
PMA	2.592	5.728	200.97	210.71	6.238	
SLSV	2.592	5.728	200.99	210.72	6.238	
SORA	2.592	5.728	200.96	210.73	6.238	
ESORA	2.593	5.731	200.93	210.64	6.242	
Meilleure	2.591435	5.730402	200.8925	210.5900	6.239425	
Médiane	2.591805	5.728224	200.7882	210.7790	6.237730	RBDO-dBA
Pire	2.591988	5.728566	201.0429	210.5984	6.238961	

Généralement, et d'un point de vue pratique, la personnalisation de la poutre, en particulier l'épaisseur ( $x_4$ ), est très coûteuse. En général, les variables de conception sont sélectionnées à partir d'un ensemble discret où leurs valeurs sont obtenues à partir des produits disponibles dans le marché [138, 158]. Par conséquent, pour tester notre méthode proposée dans le cas de variables d'optimisation discrètes, nous supposons que les variables du problème de la poutre soudée sont discrètes et définies comme suit :

$$x_1 \in \{3, 4, \dots, 50\}, \text{ avec un pas de } 1\text{mm}$$

$$x_2, x_3 \in \{1, 2, \dots, 254\}, \text{ avec un pas de } 1\text{mm}$$

$$x_4 \in \{2, 3, 4, 5, 6, 7, 8, 10, 12, 14, 15, 16, 18, 20, 22, 25\} \text{ mm}$$

Pour  $x_4$ , nous avons choisi l'ensemble des valeurs discrètes qui représentent l'épaisseur des tôles d'acier disponibles dans le marché. Alors que pour  $x_1$ ,  $x_2$  et  $x_3$ , l'ensemble des valeurs discrètes sont choisies selon les limites supérieure et inférieure avec un pas de 1 mm, parce que dans la pratique, il est plus facile de mesurer des millimètres et ne nécessitent pas usage des outils de mesure sophistiqués à haute précision. En outre, nous supposons que les paramètres physiques

( $p = [p_1, \dots, p_7]$ ) du problème sont aléatoires, et leurs fonctions de distribution et les écarts types présentés dans le Tableau 4.11, sont choisis selon un code des modèles probabilistes proposé par le Comité Mixte de la Sécurité Structurale (JCSS)[159].

Le Tableau 4.13 présente une comparaison entre la meilleure, la médiane et les solutions optimales de 25 essais, obtenues avec différents algorithmes (dBA, BA, PS et DE) du problème de la poutre soudée avec des variables de conception discrètes et des paramètres aléatoires. La population a été fixée à 50 et le nombre d'itérations à 1000. Le réglage des paramètres des différents algorithmes sont les mêmes que ceux utilisés dans l'Exemple 1. La comparaison entre les meilleures solutions montre que dBA converge vers un meilleur optimum, tandis que l'analyse de la moyenne et l'écart-type montre que dBA est plus robuste que les autres algorithmes. Pour vérifier la faisabilité de la meilleure solution obtenue avec dBA, le Tableau 4.14 présente les probabilités de défaillance des cinq contraintes évaluées avec différentes méthodes d'approximation. Comme on peut le voir, aucune des contraintes n'a été violée, et seulement la première contrainte ( $g_1$ ) est active. Pour  $g_4$ , la valeur de l'indice de fiabilité est très élevée que la probabilité de défaillance tend vers 0, d'où l'utilisation du symbole  $\infty$ .

**Tableau 4.13**

Résultats d'optimisation de l'Exemple 4 avec variables discrètes et paramètres aléatoires.

Algorithme		$x_1$	$x_2$	$x_3$	$x_4$	$f(x)$	$\nu(x)$	Moyenne( $f$ )	E.T.( $f$ )
dBA	Meilleure	6	233	232	7	3.27626	0		
	Médiane	6	230	235	7	3.28954	0	3.30780	0.02240
	Pire	6	234	238	7	3.35368	0		
BA	Meilleure	9	250	100	22	5.14718	0.69688		
	Médiane	11	100	254	22	7.96130	0	7.60905	1.35765
	Pire	21	100	252	22	10.0623	0		
PSO	Meilleure	6	233	232	7	3.27626	0		
	Médiane	6	225	239	7	3.29934	0	3.87716	0.97042
	Pire	17	104	132	22	5.77374	0		
DE	Meilleure	6	236	231	7	3.28609	0		
	Médiane	6	211	254	7	3.36508	0	3.40391	0.09913
	Pire	6	248	254	7	3.64802	0		

**Tableau 4.14**

Valeurs des contraintes, indice de fiabilité et probabilité de défaillance des contraintes de l'Exemple 4 avec variables discrètes et paramètres aléatoires.

Contrainte	Valeur de la contrainte	FORM		SORM				$P_{f-MCS}$
		$\beta$	$P_f$	$\beta_{Breitung}$	$P_{f-Breitung}$	$\beta_{Tvedt}$	$P_{f-Tvedt}$	
$g_1$	0.3118	3.0031	0.00134	3.0012	0.00134	3.0002	0.00135	0.00113
$g_2$	0.3107	3.0754	0.00105	3.0754	0.00105	3.0754	0.00105	0.00099
$g_3$	0.1429	5.8937	1.89E-09	5.8937	1.89E-09	5.8937	1.89E-09	0
$g_4$	0.9649	$\infty$	0	$\infty$	0	$\infty$	0	0
$g_5$	0.6843	5.1341	1.42E-07	5.1313	1.44E-07	5.1305	1.44E-07	0

## 4.5 Conclusions

Dans ce chapitre, nous avons développé une nouvelle méthode de résolution des problèmes d'optimisation basée sur la fiabilité, en utilisant l'algorithme des chauves-souris directionnelles. Premièrement, nous avons introduit la technique de vérification de la satisfaction des contraintes, par un niveau de comparaison- $\varepsilon$  ( $\varepsilon$ -CHT), pour permettre à l'algorithme dBA de résoudre les problèmes d'optimisation sous contraintes. Ensuite, nous avons introduit une modification pour que l'algorithme prenne en considération les variables discrètes. De plus, nous avons utilisé l'approche à boucle unique basée sur l'espace de conception fiable, pour transformer les contraintes probabilistes du problème d'RBDO en des contraintes déterministes équivalentes.

La nouvelle méthode de RBDO basée sur dBA a été testée sur plusieurs problèmes techniques avec des complexités différentes. Les résultats sont quasi-exacts par rapport aux méthodes classiques. A l'instar des méthodes classiques qui ne sont destinées qu'à un certain type de problèmes d'RBDO, notre algorithme peut prendre en considération un ensemble des variables de conception mixtes, continues et discrètes, déterministes et aléatoires, avec écart-types constants ou dépendant de la moyenne ou suivent des lois de probabilité différentes, ainsi que des contraintes déterministes et probabilistes. Par conséquent, la méthode qu'on a développée dans ce chapitre est plus générale et flexible, et peut répondre aux exigences industrielles d'où son exploitation dans le chapitre suivant relatif à la poutre principale du pont roulant.

# **Chapitre 5**

## **Optimisation basée sur la fiabilité de la poutre principale d'un pont roulant**

*When a train goes through a tunnel and it gets dark, you don't throw away the ticket and jump off. You sit still and trust the engineer. (Corrie Ten Boom 1892-1983)*

## 5.1 Introduction

En général, les ponts roulants représentent le moyen de manutention le plus répandu dans le domaine industriel. Il est rare de trouver un hangar d'usine où d'unité de fabrication sans un pont roulant. Leurs contributions dans différents secteurs économiques (ports, production d'énergie, fonderies, nucléaire, ...etc.) joue un rôle très important dans le développement économique des pays. Du à leur importance, plusieurs codes de calculs ont été introduits [160-165] pour faciliter aux constructeurs de construire des ponts roulants fiables et économiques selon la demande de leurs clients.

Puisqu'elle supporte la charge portée par le pont roulant et le poids du pont lui-même, la poutre principale représente l'organe le plus important de l'ensemble du pont roulant. D'importantes précautions doivent être prises en compte lors de la conception et le dimensionnement de la poutre. Il existe plusieurs types de poutres, poutre en triel, poutre en tube, poutre en profilé et poutre en caisson soudée. A cause de leur haute résistance à la torsion, facilitée de fabrication, poids réduit par rapport aux autres types de poutres et leurs esthétiques, les poutres en caisson soudée sont les plus utilisées par les constructeurs [166].

Dans ce chapitre, nous allons étudier le dimensionnement d'une poutre en caisson soudée d'un pont roulant à bipoutres. Dans une première partie, nous appliquons l'algorithme de chauve-souris directionnelle pour optimiser les dimensions de la poutre selon les codes BS2573 [160] et BS5400 [167]. Dans une deuxième étape, nous allons conduire une analyse de fiabilité de cette poutre en utilisant la méthode proposée dans le Chapitre 3, et dans les dernières parties nous conduisons une optimisation basée sur la fiabilité de la poutre en utilisant la méthode présentée dans le chapitre précédant.



## 5.2 Revue de littérature

Les ponts roulants tels qu'on les connaît sont apparus vers la fin des années 1870 et le début de 1880 avec un système de fonctionnement manuel. Les premiers ponts roulants motorisés ont été construits par des fabricants Américain et Anglais avec un seul moteur pour la translation du pont, et un système complexe des chaînes, des arbres et des embrayages pour le mouvement du chariot et le mécanisme de manutention. En 1890, J. H. Whiting, le fondateur de Whiting Corporation, a mis en opération le premier pont roulant à trois moteurs [168]. Depuis ce temps jusqu'à maintenant, les ponts roulants ont subi beaucoup de progrès dû au développement technologique et la recherche scientifique.

A cause de leur importance industrielle, les ponts roulants, et spécialement la poutre principale, ont été l'objet principal de plusieurs travaux de recherche. Un des principaux sujets de recherche traité dans la littérature est l'optimisation du poids du pont roulant. Rao [169] a formulé le problème d'optimisation du pont roulant avec des restrictions sur la flèche maximale, la contrainte statique, la stabilité et la rigidité globale ainsi que la capacité d'absorption des chocs en cas de collision accidentelle. Deux standards ont été adoptés pour la spécification des critères d'optimisation, à savoir le standard d'association des ingénieurs de fer et d'acier américaine [161] (AISE, Association of Iron and Steel Engineers) et l'institution indienne des normes [162] (ISI, Indian Standard Institution). L'auteur a utilisé la méthode de pénalité intérieure pour la résolution du problème d'optimisation et les résultats montrent qu'il y a une différence de 7.42% de la masse optimale du pont entre les deux standards pour un pont roulant de 105 tonnes et une portée de 14.5 mètres. Cho et Kwak [166] ont conduit une étude comparative de la conception optimale selon 4 standards, à savoir le standard de l'association américaine des fabricants des ponts roulants CMAA N°70 [163], le standard allemand DIN 15018 [164], le standard britannique BS 3579 [170] et le standard industriel japonais JIS B-8821 [165]. Les résultats d'optimisation montrent que les directives du standard CMAA N°70 produisent des ponts roulants plus légers que les autres normes.

Farkas [171] a analysé l'efficacité d'utiliser des aciers à haute résistance dans la fabrication de la poutre principale d'un pont roulant. Les résultats d'optimisation montrent que l'utilisation de ces derniers engendre une réduction du poids de la poutre entre 19 et 28%. Jármai [172] a développé un logiciel d'aide à la décision qui permet de conduire une optimisation multi-objectifs de la poutre principale. Le programme, écrit en FORTRAN, a pour objectif de réduire le poids de la poutre et les

coûts de la fabrication, de la soudure ainsi que de la peinture. Abid et *al.* [173] ont conduit une optimisation paramétrique par éléments finis de la position ainsi que de la géométrie des raidisseurs horizontaux d'un pont roulant d'une capacité de 150 tonnes et de 32 mètres de longueur. Les résultats montrent que l'utilisation des raidisseurs réduit la flèche et la contrainte maximales. Pinca et *al.* [174] ont considéré l'optimisation par éléments finis de la poutre principale avec une hauteur fixe, tandis que les autres paramètres ont subi des changements. Les auteurs ont conclu que le choix du type d'éléments finis influence la certitude des résultats. Zuberi et *al.* [175] ont proposé d'utiliser le code d'optimisation non-linéaire basé sur le gradient réduit généralisé (GRG2) pour l'optimisation de la poutre du pont roulant où les contraintes sont évaluées analytiquement ou par un code d'éléments finis. L'avantage de ce dernier est qu'il nous permet de prendre en considération des paramètres d'optimisation supplémentaires.

Dans [176], les auteurs ont utilisé la méthode de pénalité intérieure pour minimiser le poids de la poutre principale ainsi que l'énergie de fonctionnement du pont roulant, tout en imposant des critères sur la contrainte normale, la contrainte de cisaillement, la flèche verticale et horizontale et aussi la rigidité dynamique (vibration). Pavlovic et *al.* [177] ont résolu analytiquement, en utilisant la méthode des multiplicateurs de Lagrange, le problème d'optimisation de la poutre principale où le rail est placé sur l'âme principale, avec la considération des critères de la stabilité locale et latérale des plaques composantes de la poutre en caissons soudée. Par la suite, en utilisant la même méthode, Savkovic et *al.* [178] ont généralisé le problème en incluant d'autres critères d'optimisation tels que la flèche maximale et le critère de rigidité dynamique. Wang et *al.* [179] ont étudié l'utilisation des poutres à section trapézoïdale dans la conception des ponts roulants. Les résultats d'optimisation ont montré que la poutre à section rectangulaire est plus efficace. Yifie et *al.* [180] ont proposé une méthodologie pour évaluer les économies sur l'énergie de fonctionnement, et l'inclure dans l'optimisation de la structure métallique du pont roulant (la poutre principale) ainsi que le mécanisme de transmission et le système de contrôle électrique.

Lagaros et Papadrakakis [181] ont proposé une méthode d'optimisation des structures basée sur les algorithmes métaheuristiques, les éléments finis et les réseaux de neurones artificielles. Le cas considéré dans cette étude est un pont roulant mono-poutre. Les auteurs ont entraîné un réseau de neurones artificiels via un code d'éléments finis pour prédire la satisfaction ou la violation des critères d'optimisation. Par la suite, trois métaheuristiques à savoir la méthode d'optimisation par

essaim de particules (PSO), l'algorithme d'évolution différentielle (DE) et l'algorithme de recherche d'harmonie (HS), ont été considérés pour optimiser la structure métallique du pont roulant. Dans [182], les auteurs ont introduit l'équation de recherche locale basée sur la mutation à l'algorithme de la colonie de fourmis (ACAM) pour améliorer l'efficacité et prendre en considération les variables discrètes. Ce dernier est appliqué à la résolution du problème d'optimisation de la poutre principale d'un pont roulant suivant la norme chinoise [183]. Les résultats montrent que ACAM est plus efficace que l'algorithme génétique (GA), l'algorithme de la colonie de fourmis standard (ACA) et l'algorithme d'optimisation par essaim de particules (PSO).

Fan et Bi [184] ont étudié l'optimisation basée sur la fiabilité de la poutre principale. Les auteurs ont utilisé l'approche de double boucle, où la boucle extérieure (optimisation) est réalisée par l'algorithme de la colonie des fourmis (ACO), tandis que dans la boucle intérieure (estimation de la fiabilité) la méthode avancée du premier ordre et de second moment (Advanced First-Order Second-Moment, AFOSM [185]) est utilisée pour estimer la probabilité de défaillance de la poutre. Les critères d'optimisation sont imposés selon le standard chinois [183]. Les auteurs n'ont considéré l'incertitude que dans les dimensions de la section de la poutre ainsi que le chargement et le module de Young, or qu'en réalité, d'autres paramètres sont soumis à l'incertitude, tels que la limite d'élasticité par exemple. De plus, les auteurs ont considéré le facteur de sécurité dans la formulation de leurs modèles mécaniques, chose qui a conduit à une poutre plus lourde que celle dimensionnée par l'optimisation déterministe.

Dans d'autres études, Niezgodzinski et Kubiak [186] ont proposé une méthode de régénération de la poutre principale d'un pont roulant. Due aux erreurs de soudage et l'intensité d'usage, la poutre en caisson soudée accumule des déformations plastiques au fil du temps, ce qui engendre une flèche permanente de la poutre. Les auteurs ont proposé, après le redressement de la poutre, de souder des armatures en acier sur la partie inférieure des âmes ainsi que l'aile inférieure. La même conclusion est obtenue par Blum et Haremski [187] qui ont aussi proposé de souder des raidisseurs sur l'aile inférieure et des tôles sur les âmes dans la partie centrale de la poutre pour réduire la flèche. Chakri *et al.* [188] ont analysé la fiabilité de la poutre principale d'un pont roulant industriel, tandis que Mei *et al.* [189] ont formulé un modèle bi-probabiliste pour la prédiction d'apparition des fissures due à la fatigue.

Due à l'importance que représente la poutre principale, plusieurs auteurs se sont intéressés à l'optimisation des dimensions géométriques de cette dernière. Certains ont proposé d'utiliser de nouvelle méthode d'optimisation (algorithmes métaheuristiques, code éléments finis), tandis que d'autres ont imposé des critères d'optimisation additionnels tels que le critère de stabilité des âmes. En revanche, l'analyse de fiabilité et l'optimisation basée sur la fiabilité de la poutre en caisson soudée n'ont pas eu suffisamment de considérations nécessaires, ce qui fait d'eux un domaine frais pour la recherche.

### 5.3 Dimensionnement de la poutre principale

La poutre principale du pont roulant est essentiellement soumise à trois types de chargement. Le *poids mort* ou le *chargement mort* (Dead Load) représente le poids de la poutre et le poids des différents organes du pont roulant comme le chariot, le crochet, la passerelle, les moteurs...etc. Ce chargement est considéré stationnaire et constamment appliqué à la poutre. Le *poids vif* ou le *poids roulant* (Live Load) représente la charge d'exploitation qui est dans notre cas la charge dûe au crochet. Le *Poids d'impact* ou *poids du choc* (Impact Load) intervient dans deux directions, horizontale due à l'accélération et la décélération du pont lors de la translation, et verticale lors de la levée des charges. A cause du mécanisme de charge-décharge répétitive due au fonctionnement du pont, le phénomène de fatigue devient un sérieux problème qu'il faut prendre en considération. Puisqu'il est si difficile d'estimer le nombre de fluctuations entre chargement et déchargement, et afin de déterminer la contrainte admissible en fatigue, le standard britannique BS2573 [160] propose une classification des ponts roulants selon leurs conditions de service et l'état de chargement.

En suivant le cahier des charges du client, le concepteur peut supposer approximativement le nombre maximum de cycles opératoires du pont roulant et ainsi définir la classe d'utilisation comme il est indiqué dans le Tableau 5.1. Neuf classes d'utilisation (U1~9) ont été définies dans le code britannique. L'état de chargement est aussi défini selon l'avis et les besoins du client. Comme on voit dans le Tableau 5.2, l'état de chargement est divisé en quatre classes (Q1~4), *Léger*, *Moyen*, *Lourd* et *Très lourd*. Après que la classe d'utilisation et l'état de chargement sont définis, la classe du pont roulant peut être déduite du Tableau 5.3, et selon cette classe, plusieurs paramètres de conception tels que le facteur d'impact, la contrainte de fatigue admissible, le coefficient de sécurité...etc., peuvent être définis suivant la norme BS2573 [160].

**Tableau 5.1**

Les classes d'utilisation du pont roulant.

Classe d'utilisation	Le nombre maximum des cycles opératoires	Remarques
U1	$3.2 \times 10^4$	Usage peu fréquent
U2	$6.3 \times 10^4$	
U3	$1.25 \times 10^5$	
U4	$2.5 \times 10^5$	Usage moyennement fréquent
U5	$5 \times 10^5$	Usage fréquent
U6	$1 \times 10^6$	Usage très fréquent
U7	$2 \times 10^6$	Usage continu ou près de continu
U8	$4 \times 10^6$	
U9	$>4 \times 10^6$	

Note : le nombre des cycles utilisé pour la sélection des classes d'utilisation est une figure utilisée uniquement dans le but de classification et comme paramètre de conception. Il n'implique pas une durée de vie garantie (Tableau traduit et obtenu de cette source [160])

**Tableau 5.2**

Etat de chargement du pont roulant.

Etat de chargement	Description
Q1 Léger	Pont roulant qui hisse la charge maximale admissible rarement et généralement des charges légères
Q2 Moyen	Pont roulant qui hisse la charge maximale admissible occasionnellement et généralement des charges moyennes
Q3 Lourd	Pont roulant qui hisse la charge maximale admissible fréquemment et généralement des charges lourdes
Q4 Très lourd	Pont roulant qui hisse généralement des charges près de la charge maximale admissible

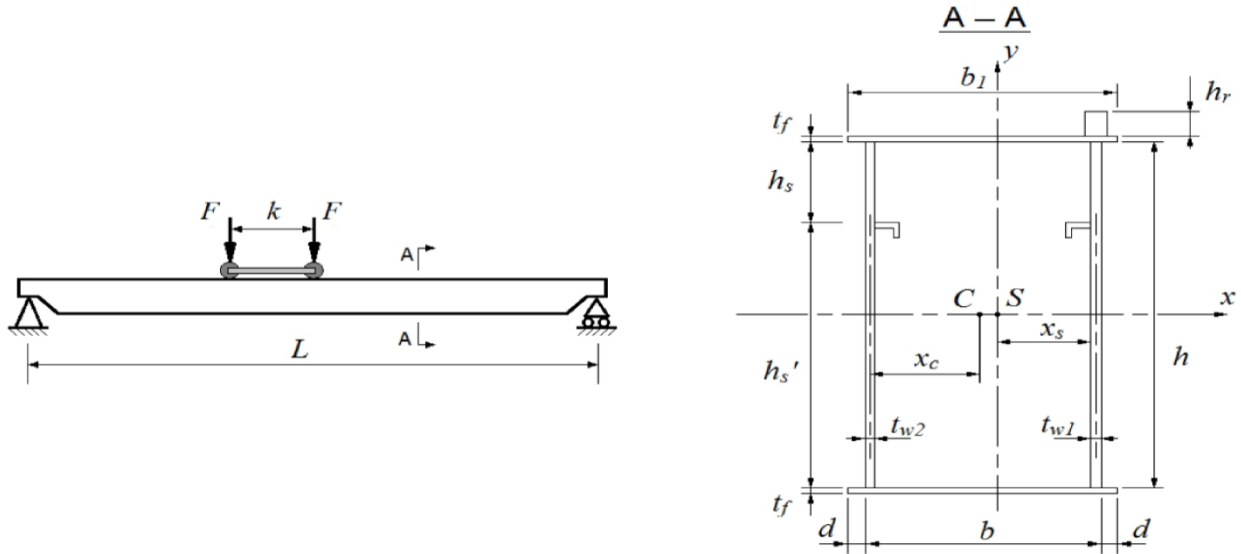
(Tableau traduit et obtenu de cette source [160])

**Tableau 5.3**

Classification des ponts roulants.

Etat de chargement	Classe d'utilisation								
	U1	U2	U3	U4	U5	U6	U7	U8	U9
Q1 Léger	A1	A1	A2	A3	A4	A5	A6	A7	A8
Q2 Moyen	A1	A2	A3	A4	A5	A6	A7	A8	A8
Q3 Lourd	A2	A3	A4	A5	A6	A7	A8	A8	A8
Q4 Très lourd	A3	A4	A5	A6	A7	A8	A8	A8	A8

(Tableau traduit et obtenu de cette source [160])



**Fig. 5.1** Coupe transversale de la poutre principale.

Dans notre étude, nous considérons la poutre en caisson soudée d'un pont roulant à bipoutres comme présentée dans la Fig. 5.1. Selon les travaux de Farkas [171] et Jármai [172] qui sont basés sur les codes BS2573 [160] et BS5400 [167], lors du dimensionnement de la poutre, six (6) critères de conception doivent être satisfaits: 1) L'aile inférieure doit vérifier le critère sur la contrainte admissible en traction; 2) L'aile inférieure doit vérifier le critère sur la contrainte de fatigue; 3) L'aile supérieure doit vérifier le critère sur flambement; 4 et 5) L'âme principale (en dessous du rail) et secondaire doivent vérifier le critère sur flambement; 6) La poutre doit vérifier la condition de rigidité (critère sur la flèche maximale admissible). Cependant, la formulation mathématique des critères de dimensionnement de la poutre en caisson soudée est comme suit :

### 5.3.1 Critère sur contrainte admissible en traction de l'aile inférieure

Le critère sur la contrainte admissible s'écrit comme suit :

$$g_1 = 1 - \left( \frac{M_x}{\alpha_d P_s W_x} + \frac{M_y}{\alpha_d P_s W_y} \right) \geq 0 \quad (5-1)$$

où  $M_x$  et  $M_y$  sont les moments fléchissants et  $W_x$  et  $W_y$  sont les modules de section.  $\alpha_d$  est un facteur de service dépendant de la classe d'utilisation (duty factor) et  $P_s$  est la contrainte admissible qui est définie par:

$$P_s = 0.59 Y_s \quad (5-2)$$

où le coefficient 0.59 représente le facteur de sécurité et  $Y_s$  représente la limite élastique. Ce facteur de sécurité est imposé comme précaution pour couvrir les incertitudes qui existent dans les propriétés physiques et mécaniques du matériau de base ainsi que dans les conditions de travail.

Le moment fléchissant dû aux charges verticales est :

$$M_x = \frac{L^2}{8}(1.05A\rho + p_r + p_s)g + \frac{F}{2L}\left(L - \frac{k}{2}\right)^2 \quad (5-3)$$

où  $L$  est la longueur de la poutre,  $\rho$  est la masse volumique de l'acier,  $g$  est l'accélération gravitationnelle,  $A$  est l'aire de la section transversale de la poutre,  $k$  est la distance entre les deux axes des roues du chariot.  $p_r + p_s$  représentent la masse linéaire de la passerelle et du rail. Le facteur 1.05 exprime la masse du diaphragme transversal.  $F$  est la force exercée par le poids du chariot et le poids du crochet au niveau de chaque roue (voir Fig. 5.1). Elle est estimée comme suit :

$$F = \frac{(\psi_d H + G_t)}{4} \quad (5-4)$$

où  $G_t$  est le poids du chariot, ainsi que  $H$  est la charge maximale admissible que le pont peut soulever. Le coefficient  $\psi_d$  est le facteur d'impact. Ce dernier est introduit comme précaution aux chocs dus au levage des poids.

Lors du démarrage de la translation du pont roulant ou son arrêt, des forces horizontales sont introduites dues à l'accélération et la décélération du pont. Le moment fléchissant induit par ces forces est exprimé par :

$$M_y = 0.15 \left( \frac{L^2}{8}(1.05A\rho + p_r + p_s)g + \frac{G_t}{8L}\left(L - \frac{k}{2}\right)^2 \right) \quad (5-5)$$

où le facteur 0.15 représente l'effet des forces d'inertie selon [171, 172].

Les modules de section, généralement connus sur le nom du module de résistance ou moment de résistance, sont calculés à partir :

$$W_x = \frac{2I_x}{(h+t_f)} \quad (5-6)$$

$$W_y = \frac{I_y}{(b-x_s+d)} \quad (5-7)$$

où  $I_x$  et  $I_y$  représentent les moments d'inertie ou les moments quadratiques de la section transversale de la poutre principale du pont roulant dont l'évaluation est donnée par :

$$I_x = \frac{h^3(t_{w1}+t_{w2})}{12} + \frac{b_1 t_f}{2} (h+t_f)^2 \quad (5-8)$$

$$I_y = h t_{w1} x_s^2 + h t_{w2} (b-x_s)^2 + \frac{b_1^3 t_f}{6} + \frac{b_1 t_f}{2} (b-2x_s)^2 \quad (5-9)$$

où

$$x_s = \frac{1}{2} \left( b - t_{w1} + h b \left( \frac{t_{w1} - t_{w2}}{A} \right) \right) \quad (5-10)$$

Les paramètres composant les trois formules précédentes sont tous des paramètres géométriques dont la signification est illustrée dans la Fig. 5.1.

### 5.3.2 Critère sur contrainte de fatigue admissible de l'aile inférieure

Le critère sur la fatigue de l'aile inférieure s'exprime par :

$$g_2 = 1 - \left( \frac{M_{xf}}{P_{ft} W_x} + \frac{M_y}{P_{ft} W_y} \right) \geq 0 \quad (5-11)$$

où  $M_{xf}$  est le moment fléchissant du à la fatigue:

$$M_{xf} = \frac{L^2}{8} (1.05 A \rho + p_r + p_s) g + \frac{K_p \psi_d H + G_t}{8L} \left( L - \frac{h}{2} \right)^2 \quad (5-12)$$



Le coefficient  $K_p$  dépend de l'état du chargement. Si c'est léger,  $K_p = 0.5$ , si c'est moyen,  $K_p = 0.63$ , tandis que pour un état de chargement lourd  $K_p = 0.8$ , et très lourd  $K_p = 1$ .  $P_{ft}$  est la contrainte de fatigue admissible. Selon le standard britannique BS 2573 [160], l'évaluation de cette dernière dépend de plusieurs paramètres, tels que l'état de chargement, la classe d'utilisation, la classe de fatigue, les contraintes maximale et minimale appliquées au pont et d'autres. Dans le cas de notre étude, la méthode d'estimation de la contrainte de fatigue admissible pour les types des ponts roulants traités dans cette thèse sera expliquée dans la Section 5.4, tandis que pour plus de détails, le lecteur doit se référer au code BS 2573-Section 8, page 46 [160].

### 5.3.3 Critère sur le flambement de l'aile supérieure

Le critère sur le flambement de l'aile supérieure est défini comme suit :

$$g_3 = 1 - \left( \frac{\sigma_{lf}}{P_s K_{lf}} + \left( \frac{\sigma_{bf}}{P_s K_{bf}} \right)^2 \right) \geq 0 \quad (5-13)$$

où

$$\sigma_{lf} = \frac{M_x}{W_x} \quad (5-14)$$

$$\sigma_{bf} = \frac{M_y}{W_y} \quad (5-15)$$

L'évaluation des coefficients  $K_{lf}$  et  $K_{bf}$  dépend de la valeur du coefficient d'élanement  $\lambda_f$  :

$$\lambda_f = \frac{b}{t_f} \sqrt{\frac{Y_s}{355}} \quad (5-16)$$

Cependant

$$K_{bf} = 1.3 - 0.0027\lambda_f \quad (5-17)$$

et

$$\left\{ \begin{array}{l} \text{Si } \lambda_f \leq 24 \Rightarrow K_{1f} = 1 \\ \text{Si } 24 < \lambda_f \leq 47 \Rightarrow K_{1f} = \left( \frac{24}{\lambda_f} \right)^{0.75} \\ \text{Si } 47 < \lambda_f \leq 130 \Rightarrow K_{1f} = \left( \frac{26}{\lambda_f} \right)^{0.85} \\ \text{Si } 130 < \lambda_f \leq 300 \Rightarrow K_{1f} = 0.274 - \frac{\lambda_f}{7000} \end{array} \right. \quad (5-18)$$

### 5.3.4 Critère sur le flambement de l'âme principale

Le critère sur le flambement de l'âme principale qui est sous le rail est défini comme suit:

$$g_4 = 1 - \left( \sqrt{\left( \frac{((h-h_s)/h)\sigma_{lf} + \sigma_{bf}}{P_s K_{1w}} \right)^2 + \left( \frac{\sigma_{cw}}{P_s K_{2w}} \right)^2} + \left( \frac{(h_s/h)\sigma_{lf}}{P_s K_{bw}} \right)^2 + 3 \left( \frac{\tau}{P_s K_{qw}} \right)^2 \right) \geq 0 \quad (5-19)$$

La formule approximative de la contrainte de compression sur l'âme est:

$$\sigma_{cw} = \frac{F}{t_{w1} a_w} \quad (5-20)$$

Selon Farkas [171], la constante  $a_w$  est:

$$a_w = 50 + 2(h_r + t_f - 5) \quad (5-21)$$

La contrainte de cisaillement superposée est estimée par :

$$\tau = \tau_Q + \tau_t = \frac{FS_x}{I_x(t_{w1} + t_{w2})} + \frac{(b - x_c)F}{2A_k t_{w1}} \quad (5-22)$$

où

$$S_x = \frac{bt_f h}{4} \quad (5-23)$$

$$A_k = h \cdot b \quad (5-24)$$

$$h_w = 1,9 \sqrt{\frac{a_w h}{5K_w}} \quad (5-25)$$

$$K_w = \left(3,4 + \frac{2,2h}{5a}\right) \left(0,4 + \frac{a_w}{2a}\right) \quad (5-26)$$

$a$  est la distance entre deux diaphragmes successifs.

Le coefficient d'élanement de la partie supérieure de l'âme principale dans le cas d'une compression locale, est défini comme suit :

$$\lambda_e = \frac{h_w}{t_{w1}} \sqrt{\frac{Y_s}{355}} \quad (5-27)$$

Cependant, le coefficient  $K_{2w}$  est calculé selon les formules suivantes :

$$\left\{ \begin{array}{l} \text{Si } \lambda_e \leq 24 \Rightarrow K_{2w} = \left(\frac{24}{\lambda_e}\right)^{0.5} \\ \text{Si } 43 < \lambda_e \leq 59 \Rightarrow K_{2w} = \left(\frac{28}{\lambda_e}\right)^{0.68} \\ \text{Si } 59 < \lambda_e \leq 90 \Rightarrow K_{2w} = \left(\frac{30}{\lambda_e}\right)^{0.75} \\ \text{Si } 90 < \lambda_e \leq 130 \Rightarrow K_{2w} = \left(\frac{36}{\lambda_e}\right)^{0.9} \\ \text{Si } 130 < \lambda_e \leq 200 \Rightarrow K_{2w} = 0.38 - \frac{\lambda_e}{2000} \end{array} \right. \quad (5-28)$$

Le coefficient d'élanement de la partie supérieure de l'âme principale dans le cas de la flexion est :

$$\lambda_w = \frac{0.2h}{t_{w1}} \sqrt{\frac{Y_s}{355}} \quad (5-29)$$

Les coefficients  $K_{bw}$ ,  $K_{lw}$  et  $K_q$  sont définis comme suit :

$$K_{bw} = 1.3 - 0.0027\lambda_w \quad (5-30)$$

et

$$\left\{ \begin{array}{l} \text{Si } \lambda_w \leq 24 \Rightarrow K_{1w} = 1 \\ \text{Si } 24 < \lambda_w \leq 47 \Rightarrow K_{1w} = \left( \frac{24}{\lambda_w} \right)^{0.75} \\ \text{Si } 47 < \lambda_w \leq 130 \Rightarrow K_{1w} = \left( \frac{26}{\lambda_w} \right)^{0.85} \\ \text{Si } 30 < \lambda_w \leq 300 \Rightarrow K_{1w} = 0.274 - \frac{\lambda_w}{7000} \end{array} \right. \quad (5-31)$$

tandis que

$$\left\{ \begin{array}{l} \text{Si } \lambda_w \leq 40 \Rightarrow K_q = 1 \\ \text{Si } 40 < \lambda_w \leq 112 \Rightarrow K_q = 1 - 0.385 \left( \frac{\lambda_w - 40}{60} \right)^{0.743} \\ \text{Si } 112 < \lambda_w \leq 200 \Rightarrow K_q = 1 - 0.660 \left( \frac{\lambda_w - 40}{160} \right)^{0.505} \\ \text{Si } 200 < \lambda_w \leq 300 \Rightarrow K_q = 0.34 - 0.07 \left( \frac{\lambda_w - 200}{100} \right)^{0.8} \end{array} \right. \quad (5-32)$$

Ces approximations ne sont valides que dans le cas où le coefficient  $\phi = a/h$  est approximativement égal à 2.

### 5.3.5 Critère sur le flambement de l'âme secondaire

L'évaluation du critère imposé sur le flambement de l'âme secondaire se fait de la même façon que pour l'âme principale, seulement que dans ce cas, on néglige la contrainte de compression locale  $\sigma_{cw} = 0$ . Cependant,  $g_5$  est défini comme suit:

$$g_5 = 1 - \left( \left( \frac{((h - h_s) / h) \sigma_{lf} + \sigma_{bf}}{P_s K_{1w}} \right) + \left( \frac{(h_s / h) \sigma_{lf}}{P_s K_{bw}} \right)^2 + 3 \left( \frac{\tau}{P_s K_{qw}} \right)^2 \right) \geq 0 \quad (5-33)$$

Pour l'évaluation de  $g_5$ , on utilise les mêmes équations que pour  $g_4$  (Eqs. 5-20~32) tout en remplaçant  $t_{w1}$  par  $t_{w2}$ .

### 5.3.6 Critère sur la flèche maximale de la poutre

Le critère de rigidité impose une flèche maximale admissible sur la déviation de la poutre due au chargement :

$$g_6 = 1 - \left( \frac{H(L-k)(3L^2 - (L-k)^2)}{192EI_x w_p} \right) \geq 0 \quad (5-34)$$

où  $w_p$  est la flèche maximale admissible et  $E$  est le module de Young.

## 5.4 Optimisation de la poutre principale

Une conception optimale a plusieurs avantages à la fois au client ainsi qu'au constructeur. La réduction du poids de la poutre permet au fabricant de réduire le coût de fabrication en faisant des économies sur la matière première, tandis que pour le client, elle lui permet de réduire le coût de fonctionnement du pont (moins d'énergie électrique nécessaire pour le fonctionnement du pont roulant). La recherche d'une conception optimale de la poutre est un problème d'optimisation sous contraintes. Afin de résoudre ce problème, nous allons utiliser l'algorithme des chauves-souris directionnelles (dBA) combiné avec la  $\varepsilon$ -CHT ( $\varepsilon$ -Constraints Handling Technique) comme il est décrit dans le chapitre précédant (Chapitre 4).

### 5.4.1 Formulation du problème d'optimisation

L'objectif de l'optimisation est de réduire le poids de la poutre principale du pont roulant. On considère que les composants de la poutre en caisson soudée sont fabriqués de la même nuance d'acier. Puisque la longueur est fixée par le cahier de charge du client, la fonction objectif du problème d'optimisation se réduit à la minimisation de l'aire de la section transversale de la poutre. Par conséquent, le problème d'optimisation sous contraintes de la poutre s'écrit comme suit:

$$\left\{ \begin{array}{l} \text{Minimiser } A = h(t_{w1} + t_{w2}) + 2bt_f \\ \text{Tel que: } g_i \geq 0, \quad i = 1, 2, \dots, 6 \\ \quad \quad \quad 3 \leq t_{w1}, t_{w2}, t_f \leq 25mm \\ \quad \quad \quad 500 \leq h \leq 1500mm \\ \quad \quad \quad 100 \leq b \leq 1000mm \end{array} \right. \quad (5-35)$$

où  $A$  est l'aire de la section transversale de la poutre.

**Tableau 5.4**

Paramètres communs des ponts roulants considérés.

Paramètre	Valeur	Paramètre	Valeur
$H$	200 kN	$A$	2.25m
$G_t$	42.25 kN	$p_r + p_s$	190 kg/m
$L$	22.5 m	$E$	210 GPa
$K$	1.9 m	$G$	9.81 m/s <sup>2</sup>
$h_r$	70 mm	$\rho$	7850 kg/m <sup>3</sup>

**Tableau 5.5**

Paramètres et classes des ponts roulants considérés.

Classe du pont roulant	Etat de chargement	Classe d'utilisation	$K_p$	$\alpha_d$	$\psi_d$	$w_p$	$P_{ft}$ (MPa)
A4	Q1 Léger	U5	0.50	1.00	1.1	$L/500$	169+145(R-0.1)
A6	Q2 Moyen	U6	0.63	0.95	1.3	$L/600$	155+135(R-0.1)
A8	Q3 Lourd	U7	0.80	0.90	1.4	$L/700$	142+125(R-0.1)

Pour la fabrication des composants de la poutre en caisson soudée, les constructeurs utilisent généralement des tôles d'acier procurés du marché local. Ces dernières sont caractérisées par des épaisseurs standardisées. De plus, lors du découpage des tôles, il est plus facile de mesurer des millimètres que des micromètres qui nécessitent des instruments de mesure sophistiqués, ce qui augmente le coût de fabrication. Ainsi, le problème d'optimisation de la poutre est un problème d'optimisation discret, dont les valeurs des variables d'optimisation sont définies comme suit :

- $h$ : varie entre 500 et 1500mm avec un pas de 1mm;
- $b$ : varie entre 100 et 1000mm avec un pas de 1mm;
- $t_{w1}, t_{w2}$  et  $t_f \in \{3, 4, 5, 6, 7, 8, 10, 12, 14, 15, 16, 18, 20, 22, 25\text{mm}\}$ .

Dans notre étude, nous considérons le dimensionnement de la poutre principale de trois types de ponts roulants avec la même longueur ( $L = 22.5\text{m}$ ) et une charge maximale ( $H = 20$  tonnes). Les paramètres de conception communs des trois ponts roulants sont illustrés dans le Tableau 5.4. On note que ces paramètres sont tirés de [171], et ont été utilisés dans [172] pour la validation d'un nouveau programme d'optimisation.

Les trois ponts sont de classes différentes A4, A6 et A8 où l'état de chargement de chaque pont et sa classe d'utilisation sont indiqués dans le Tableau 5.5. Selon le code BS 2573 [160], en utilisant ces trois derniers critères, nous pouvons déterminer le facteur d'impact  $\psi_d$ , le spectre de chargement  $K_p$ ,

le facteur de service  $\alpha_d$  ainsi que la flèche admissible  $w_p$ . La détermination de la classe de fatigue de la poutre dépend de la méthode de fabrication de la poutre elle-même. Si on suppose que le caisson soudé est fabriqué avec des tôles jointes par des cordons de soudure continus, la classe de fatigue considérée est la classe D. Par conséquent, la contrainte de fatigue admissible  $P_{fi}$  est définie comme elle est indiquée dans le Tableau 5.5, où  $R$  représente le rapport entre la contrainte maximale et minimale subie par la poutre. Dans ce cas, la contrainte maximale est celle qui est engendrée par un poids de chargement maximal, tandis que la contrainte minimale est celle causée par le poids propre du pont roulant avec crochet vide. Cependant, on définit  $R$  comme suit :

$$R = (\sigma_{\min} / \sigma_{\max}) = (M_{x1} / M_x) \quad (5-36)$$

où

$$M_{x1} = \frac{L^2}{8} (1.05 A \rho + p_r + p_s) g \quad (5-37)$$

$M_{x1}$  représente le moment fléchissant induit par le poids du pont roulant.

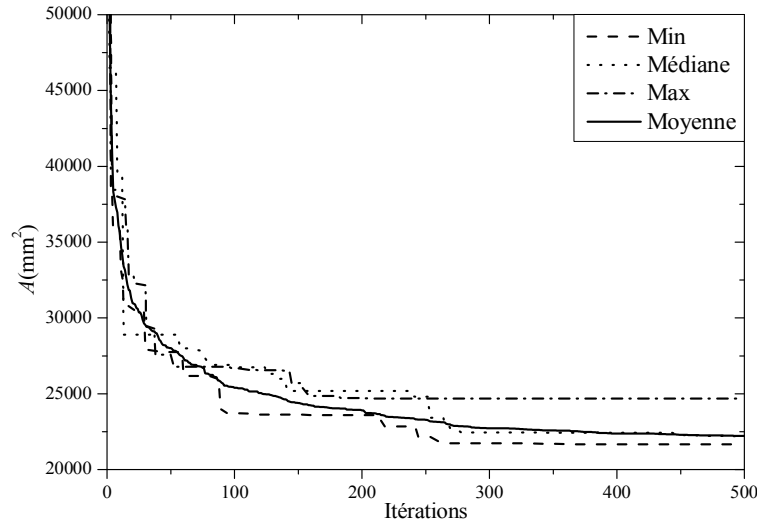
De plus, nous analysons aussi l'effet de l'utilisation des aciers à haute résistance sur la réduction du poids de la poutre. Trois nuances d'acier ont été considérées, le 43, le 50 et le 55 selon le standard britannique, dont leurs équivalents en norme européenne sont le S235, le S355 et le S460. En général, le chiffre après le S représente la limite élastique de l'acier considéré en MPa. Pour les besoins de comparaison, nous considérons que la limite élastique des trois nuances d'acier considérées (S235, S355 et S460) sont comme elles sont indiquées dans [171],  $Y_s = 230, 355$  et  $450$  MPa respectivement.

## 5.4.2 Résultats d'optimisation et discussions

### 5.4.2.1 Analyse de sensibilité et paramétrage de dBA

Dans un premier cas, nous considérons l'optimisation de classe A4 construit avec un acier dont la limite d'élasticité est  $Y_s = 230$  MPa. Nous exécutons l'algorithme des chauves-souris directionnelles 25 fois avec une population de 25 chauves-souris et pour 500 itérations.

La Fig. 5.2 représente l'évolution de la fonction objectif (dans notre cas c'est l'aire de la section de la poutre  $A$ ) au cours des itérations du meilleur essai (essai qui a conduit à la section minimale), du médian et du mauvais essai qui a conduit à une solution dont la section est maximale, ainsi que la



**Fig. 5.2** Evolution de la minimisation de la fonction objectif au cours des itérations.

moyenne des 25 essais. Comme on l'aperçoit dans la Fig. 5.2, à chaque fois qu'on exécute notre programme, l'algorithme converge vers une solution différente. Ceci est une caractéristique connue des algorithmes métaheuristiques causée par le mouvement aléatoire des chauves-souris. Afin de réduire l'effet de ce dernier et augmenter la qualité des résultats et la fiabilité de l'algorithme, nous avons procédé à une série de tests pour trouver la meilleure combinaison entre le nombre des itérations et la population des chauves-souris.

Le Tableau 5.6 représente les résultats statistiques de 25 exécutions de l'algorithme des chauves-souris directionnelles avec différents réglages du nombre des itérations ( $t_{max}$ ) et de la population des chauves-souris ( $N$ ). Les résultats présentés sont la meilleure, la médiane et la pire solution en plus de la moyenne et l'écart type des 25 exécutions. Comme on peut le constater, en augmentant la population des chauves-souris ainsi que le nombre des itérations, la meilleure, la médiane et la mauvaise solution ont tendance à converger vers la même solution. Ceci se traduit par la réduction de la valeur de l'écart-type où dans certains cas, il tend carrément vers zéro (à savoir  $N = 100$  et  $t_{max} = 1500$ ). Cela signifie que lorsque les valeurs de la paire  $(N, t_{max})$  augmentent, la robustesse de l'algorithme et la fiabilité des résultats s'améliorent. En outre, l'analyse de la violation de contrainte (la moyenne de  $g_i$ ) montre que les solutions obtenues par dBA combiné avec la technique  $\varepsilon$ -CHT sont toutes réalisables et faisables et nulle d'entre elles n'a violé les six critères de conception du pont roulant.



**Tableau 5.6**

 Les résultants statistiques de l'optimisation pour différentes paires ( $N$ ,  $t_{max}$ ).

$t_{max}$		$N$				
		10	30	50	75	100
500	Minimum	21666	21666	21666	21666	21666
	Médian	21692	21666	21666	21666	21666
	Maximum	22620	22360	22226	21670	21669
	Moyenne	21931	21803	21690	21666	21666
	E.T.	324.12	233.77	109.49	0.7838	0.5879
	Moyenne ( $g_i$ )	0	0	0	0	0
1000	Minimum	21666	21666	21666	21666	21666
	Médian	22054	21666	21666	21666	21666
	Maximum	22478	22194	22194	21669	21666
	Moyenne	21932	21687	21691	21666	21666
	E.T.	265.29	103.40	103.77	0.8139	0
	Moyenne ( $g_i$ )	0	0	0	0	0
1500	Minimum	21666	21666	21666	21666	21666
	Médian	21668	21666	21666	21666	21666
	Maximum	22450	21670	21669	21668	21666
	Moyenne	21744	21666	21666	21666	21666
	E.T.	205.63	1.0151	0.8139	0.3919	0
	Moyenne ( $g_i$ )	0	0	0	0	0
2000	Minimum	21666	21666	21666	21666	21666
	Médian	21666	21666	21666	21666	21666
	Maximum	22441	21666	21666	21666	21666
	Moyenne	21858	21666	21666	21666	21666
	E.T.	265.77	0	0	0	0
	Moyenne ( $g_i$ )	0	0	0	0	0

E.T. : écart type.

**Tableau 5.7**

 Taux de succès pour différentes paires ( $N$ ,  $t_{max}$ ).

$t_{max}$	$N$				
	10	30	50	75	100
500	16%	52%	76%	81%	85%
1000	43%	72%	89%	98%	96%
1500	52%	86%	98%	97%	100%
2000	64%	93%	99%	99%	100%

En analysant les résultats d'optimisation illustrés dans le Tableau 5.6, nous avons constaté que l'aire optimale de la section de la poutre principale d'un pont roulant de classe A4 est de 21666 mm<sup>2</sup>. En se basant sur cette valeur, nous avons conduit une deuxième analyse pour évaluer le taux de succès de l'algorithme dBA à converger vers cette solution. Avec les mêmes paramétrages du précédent essai, nous avons exécuté dBA 100 fois et nous avons enregistré le nombre des exécutions qui ont convergé vers la solution optimale ( $A = 21666$  mm<sup>2</sup>). En pourcentage, le taux de succès est présenté

dans le Tableau 5.7. Comme on peut le voir, l'accroissement de la valeur de la paire  $(N, t_{max})$  améliore le taux de succès, et pour  $N \geq 100$  et  $t_{max} \geq 1500$ , le taux de succès est de 100%. Cela signifie la probabilité que l'algorithme converge vers l'optimum de 100%. Par conséquent, dans le reste de notre étude d'optimisation, nous fixons la population des chauves-souris à 100 et le nombre des itérations à 1500. Tous les résultats dans ce qui suit sont obtenus avec une seule exécution de l'algorithme de chauves-souris directionnelles.

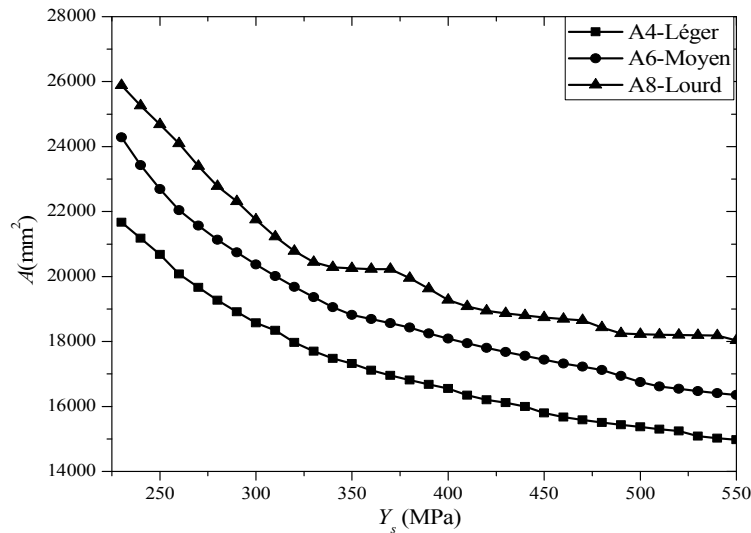
#### 5.4.2.2 Optimisation du pont roulant par dBA

Nous considérons l'optimisation de trois poutres d'un pont roulant avec des états de chargement différents, léger, moyen et lourd et qui sont indiqués dans les Tableaux 5.4 et 5.5. Trois nuances d'acier de construction ont été considérés dont les limites d'élasticités de chaque acier sont  $Y_s = 230$ , 355 et 450MPa. Nous fixons la population de chauves-souris à  $N = 100$  et le nombre maximal d'itérations à  $t_{max} = 1500$ . Le Tableau 5.8 présente une comparaison entre les solutions optimales obtenues avec dBA et les résultats de Farkas [171] obtenus en utilisant la programmation combinatoire discrète Backtrack. Comme on peut le voir, les solution obtenues avec dBA sont beaucoup meilleures que ceux de Farkas [171] avec une réduction sur le poids de la poutre entre 8% et 16% selon l'état de chargement et la nuance d'acier utilisé.

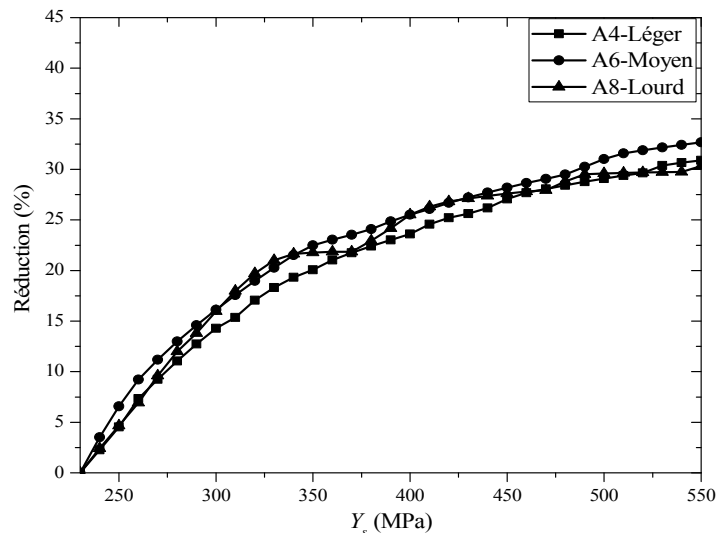
**Tableau 5.8**

Résultats d'optimisation de la poutre principale pour différentes classes et nuances d'acier.

Classe	$Y_s$ (MPa)	$h$ (mm)	$t_{w1}$ (mm)	$t_{w2}$ (mm)	$b$ (mm)	$t_f$ (mm)	$A$ (mm <sup>2</sup> )	différence (%)	Contrainte statique(MPa)	Contrainte de fatigue (MPa)	Flèche (mm)	Réf
A4 Léger	230	1050	6	6	400	14	23800	--	130<136	95<193	29.8<45.0	[171]
		1206	6	5	420	10	21666	08.97	136<136	99<191	23.7<45.0	dBA
	355	950	5	5	375	14	20000	--	159<209	116<191	40.0<45.0	[171]
		914	5	4	448	10	17186	14.07	187<209	134<188	45.0<45.0	dBA
	450	1050	5	5	375	10	18000	--	174<266	125<189	43.2<45.0	[171]
		1014	5	4	417	8	15798	12.23	202<266	144<187	44.3<45.0	dBA
A6 Moyen	230	1150	7	7	375	14	26600	--	129<129	102<175	25.2<37.5	[171]
		1194	6	5	466	12	24318	08.58	126<129	99<174	19.9<37.5	dBA
	355	1050	6	6	325	14	21700	--	165<199	130<173	35.5<37.5	[171]
		1023	6	5	375	10	18753	13.58	196<199	153<171	37.5<37.5	dBA
	450	1000	5	5	325	16	20400	--	170<253	133<172	36.1<37.5	[171]
		1022	5	4	412	10	17438	14.52	194<252	151<170	37.3<37.5	dBA
A8 Lourd	230	1150	7	7	450	14	28700	--	119<122	105<160	21.7<32.1	[171]
		1292	7	5	519	10	25884	9.81	121<122	107<158	17.1<32.1	dBA
	355	1000	6	6	325	18	23700	--	157<188	139<158	31.7<32.1	[171]
		1125	6	5	393	10	20235	14.62	177<189	156<156	29.1<32.1	dBA
	450	1050	5	5	425	14	22400	--	149<239	131<157	29.0<32.1	[171]
		1111	5	4	437	10	18739	16.34	176<239	155<155	29.6<32.1	dBA



**Fig. 5.3** Evolution de l'aire optimale de la section transversale de la poutre principale en fonction de la limite d'élasticité de l'acier utilisé.

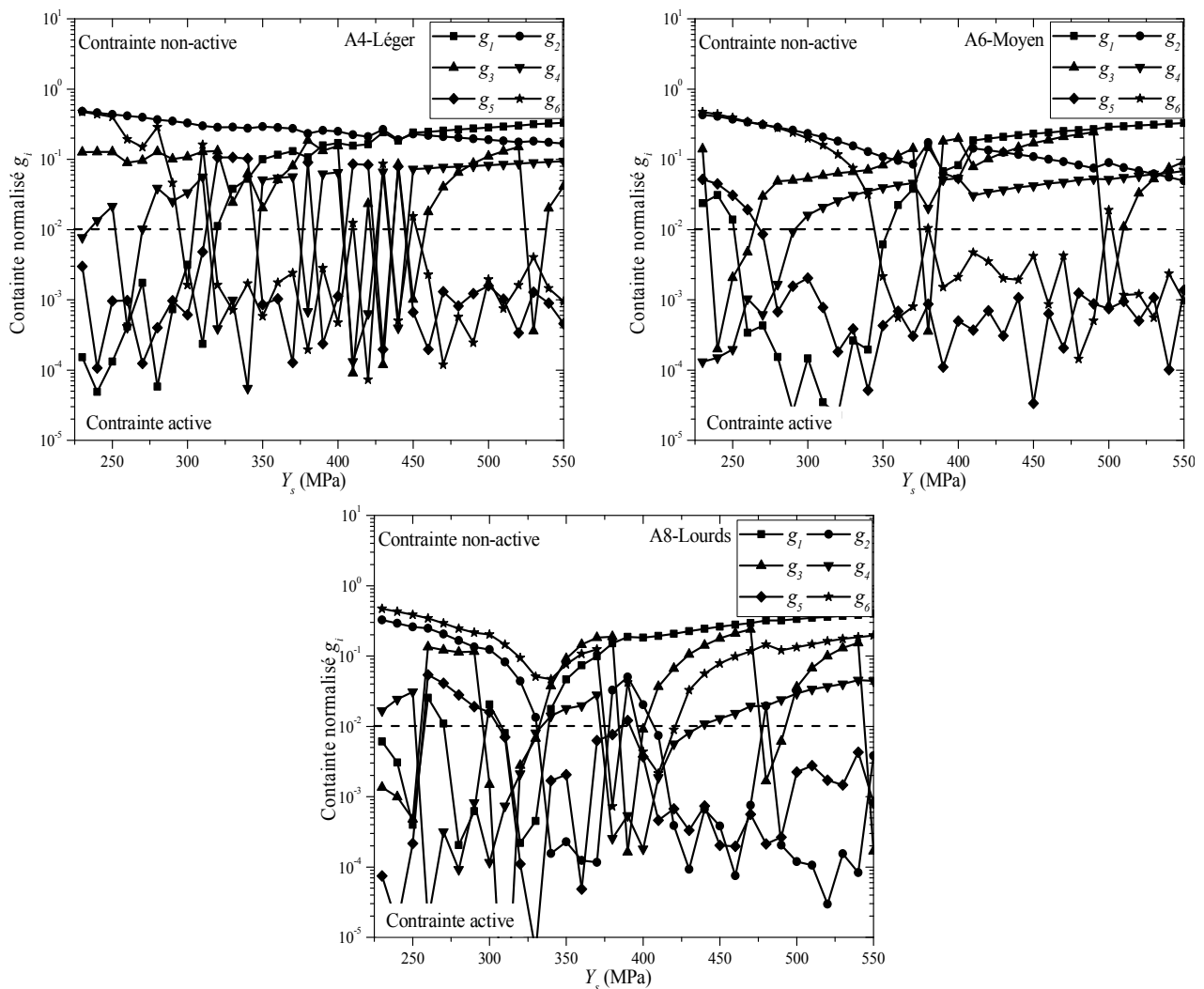


**Fig. 5.4** Pourcentage de réduction en masse de la poutre principale en fonction de la limite d'élasticité de l'acier utilisé.

L'utilisation des aciers à haute résistance peut entraîner une importante réduction de la masse de la poutre, ainsi que des économies de coût de fabrication et de l'énergie d'exploitation comme le Tableau 5.8 le montre. La Fig. 5.3 présente l'aire de section optimale de la poutre-caisson en fonction de la limite d'élasticité de l'acier utilisé (la limite d'élasticité varie entre 230 et 550 MPa). L'augmentation de la valeur de la limite d'élasticité engendre une diminution de l'aire de la section transversale de la poutre principale. Dans le cas d'un pont roulant de classe A8 lourd chargement, on observe une certaine stabilité dans l'évolution de l'aire de la section optimale autour de 350 et 375MPa. Ceci est dû au caractère discret des variables d'optimisation. La Fig. 5.4 présente le

pourcentage de réduction de la masse de la poutre en fonction de la limite d'élasticité de l'acier utilisé. Nous remarquons que l'état de chargement a un effet sur l'économie mais pas autant que la limite d'élasticité.

La Fig. 5.5 présente l'état des contraintes d'optimisation de la conception optimale en fonction de la limite d'élasticité. Trois cas d'états de charge sont considérés. Si la valeur d'une contrainte est faible à 0.01, elle est considérée comme active, tandis que si elle est négative, la contrainte est violée. Dans une première constatation, nous observons qu'aucune des contraintes ou critères d'optimisation est négatif pour tous les points de conception. Ceci signifie que les résultats obtenus par dBA sont tous réalisables et sans aucune violation des contraintes.



**Fig. 5.5** Etat des contraintes d'optimisation normalisées à chaque point optimal.

### **5.4.3 Conclusion partielle**

L'utilisation de l'algorithme des chauves-souris directionnelles associé à la technique  $\varepsilon$ -CHT pour l'analyse de la satisfaction des contraintes s'avère un outil très puissant dans la recherche des conceptions optimales. Le bon calibrage des paramètres de l'algorithme peut conduire à des résultats très fiables et robustes, et une réduction importante dans la masse de la poutre principale. De plus, l'utilisation des aciers à haute résistance conduit à réaliser des importantes économies sur les coûts de fabrication et de fonctionnement.

## **5.5 Analyse de fiabilité de la poutre optimisée**

### **5.5.1 Définition du problème d'analyse de fiabilité et configuration**

En général, l'optimisation structurelle déterministe prend en considération les incertitudes qui existent dans les propriétés du matériau de construction (i.e. limite élastique), ainsi que dans les conditions de travail (incertitude dans le chargement) par le biais du facteur de sécurité qui est tiré des codes de calcul comme l'Eurocode, ASME, BS et autres. Dans notre étude, le facteur de sécurité imposé est de 0.59 (voir Eq. 5-2). Ce dernier qui est sensé couvrir le domaine de conception fiable, n'est pas sensible aux fluctuations et aux incertitudes dans les paramètres de conception, et peut conduire, dans plusieurs cas, à des structures sur ou sous-dimensionnées [134]. Cela signifie que l'optimisation déterministe ne peut pas garantir une conception fiable et économique. Par conséquent, pour l'analyse de fiabilité de la poutre principale du pont roulant, nous considérons que le facteur de sécurité est égal à 1 ( $P_s = Y_s$ ) et les paramètres physiques, mécaniques, géométriques ainsi que ceux liés au fonctionnement du pont, tel que le chargement, sont des variables aléatoires.

La détermination de la nature aléatoire d'une variable ainsi que ces paramètres caractéristiques nécessite la réalisation d'un grand nombre de mesures expérimentales. En se basant sur la littérature et les expériences menées partout dans le monde, le comité mixte de la sûreté des structures (the Joint Committee on Structural Safety - JCSS) a proposé un code sur les modèles probabilistes [159] lié à la fiabilité des structures et qu'on peut télécharger sur [www.jcss.ethz.ch](http://www.jcss.ethz.ch). Selon ce code, les modèles probabilistiques des variables de notre problème ainsi que leurs paramètres correspondants (moyenne et écart type) sont configurés comme ils sont listés dans le Tableau 5.9.

**Tableau 5.9**  
Caractéristiques des variables aléatoires et déterministes.

Variable	Distribution	Moyenne	E.T.(% ou valeur)	Variable	Distribution	Moyenne	E.T.(% ou valeur)
$h$	Normale	Selon le cas	1mm	$\rho$	Log-Normale	7850kg/m <sup>3</sup>	50kg/m <sup>3</sup>
$t_{w1}$	Normale	Selon le cas	5%	$E$	Log-Normale	210GPa	14.70GPa
$t_{w2}$	Normale	Selon le cas	5%	$h_s$	Normale	Selon le cas	1mm
$t_f$	Normale	Selon le cas	5%	$L$	Déterministe	22500mm	--
$b$	Normale	Selon le cas	1mm	$K$	Déterministe	1900mm	--
$d$	Normale	20mm	1mm	$G_t$	Déterministe	42.25kN	--
$h_r$	Normale	70mm	1mm	$p_r+p_s$	Déterministe	190kg/m	--
$Y_s$	Log-Normale	Selon le cas	7%	$G$	Déterministe	9.81m/s <sup>2</sup>	--
$H$	Log-Normale	200KN	10%	$A$	Déterministe	2250mm	--

Le problème d'analyse de fiabilité de la poutre principale d'un pont roulant se constitue de six (6) équations d'états limites (Eqs 5-1, 11, 13, 19, 33 et 34). Si par exemple, la contrainte de compression soumise à l'aile inférieure dépasse la limite d'élasticité, une déformation plastique se produit au sein de l'aile. Selon sa gravité, cette déformation peut conduire à l'arrêt du pont pour correction. La même chose peut arriver si une des autres états limites est dépassée. Par conséquent, notre problème de fiabilité est un système en série qui s'écrit comme suit :

$$g = \min([g_1, g_2, g_3, g_4, g_5, g_6]) \quad (5-38)$$

où la probabilité de défaillance est égale à l'espérance que  $g$  soit inférieure à zéro :

$$P_f = E[g < 0] \quad (5-39)$$

### 5.5.1 Résultats d'analyse de fiabilité et discussions

Pour la détermination de l'indice de fiabilité du pont roulant ainsi que la probabilité de défaillance, nous employons la méthode d'estimation de la fiabilité basée sur l'algorithme des chauves-souris directionnelles décrit dans le Chapitre 3. Le problème d'estimation de la fiabilité de la poutre principale se constitue de 11 variables aléatoires comme il est indiqué dans le Tableau 5.9. En effet, le nombre de chauve-souris est de  $11 \times 50$  ( $N = 550$ ) et le nombre des itérations est de  $t_{max} = 100$ . Nous considérons les neuf conceptions optimales listées dans le Tableau 5.8, obtenues par dBA pour les trois classes des ponts roulants avec trois nuances d'acier.

**Tableau 5.10**

Les résultats statistiques de l'indice de fiabilité pour 25 exécutions de l'algorithme dBA.

Classe	A4-Léger			A6-Moyen			A8-Lourd		
	$Y_s=230$	355	450	230	355	450	230	355	450
Min	3.7612	0.0199	0.1425	4.2066	1.4539	1.4781	3.0887	0.0472	0.0497
Médian	3.7615	0.0199	0.1425	4.2073	1.4541	1.4783	3.0890	0.0472	0.0497
Max	3.7629	0.0200	0.1426	4.3713	1.4549	1.4788	3.0900	0.0472	0.0497
Moyenne	3.7616	0.0199	0.1425	4.2359	1.4541	1.4783	3.0891	0.0472	0.0497
E.T.	4.27E-04	7.24E-06	2.67E-05	6.03E-02	2.22E-04	1.90E-04	3.42E-04	1.28E-06	1.65E-06

E.T.: Ecart Type

Dans un premier test, nous considérons le système de fiabilité décrit par l'Eq. (5-38), nous exécutons l'algorithme des chauves-souris directionnelles 25 fois pour les différentes classes des ponts roulants et des nuances d'acier. Le Tableau 5.10 représente les résultats statistiques de l'indice de fiabilité ( $\beta$ ) obtenus avec dBA. Comme on peut le voir, la comparaison entre les résultats du minimum et du maximum de l'indice de fiabilité obtenue dans une exécution, montre que les résultats de dBA sont très fiables et robustes. Une deuxième remarque montre qu'à part les ponts roulants construits avec un acier dont la limite d'élasticité est de 230MPa, les autres ont un indice de fiabilité très bas, ce qui se traduit par une forte probabilité de défaillance.

Afin de savoir où la défaillance risque de se produire, nous conduisons une estimation de l'indice de fiabilité de chaque équation d'état limite ( $g_1, g_2, g_3, g_4, g_5$  et  $g_6$ ) séparément en utilisant dBA avec la même configuration précédente ( $N$  et  $t_{max}$ ) et une seule exécution. Les histogrammes de la Fig. 5.6 représentent la valeur de l'indice de fiabilité pour chaque équation d'état limite. Nous remarquons que pour les poutres construites avec un acier dont la limite d'élasticité est de 230 MPa, les indices de fiabilité des six équations d'états limites sont élevés ( $\beta > 3$ , ce qui implique que  $P_f < 0.135\%$ ).

En retournant vers la Fig. 5.5, nous observons que les contraintes d'optimisation actives pour  $Y_s=230$ MPa, sont les trois critères imposés sur le flambement de l'aile supérieure et les deux âmes, principale et secondaire ( $g_3, g_4$  et  $g_5$  respectivement) quelle que soit la classe du pont considéré. Ceci se traduit par les faibles valeurs de  $\beta$  enregistrées par rapport aux autres (sans oublier qu'elles sont supérieures à 3). Puisque ces trois critères de flambement ont été imposés sur la base que la contrainte admissible est  $P_s = 0.59Y_s$ , le résultat de l'optimisation déterministe est une poutre surdimensionnée.

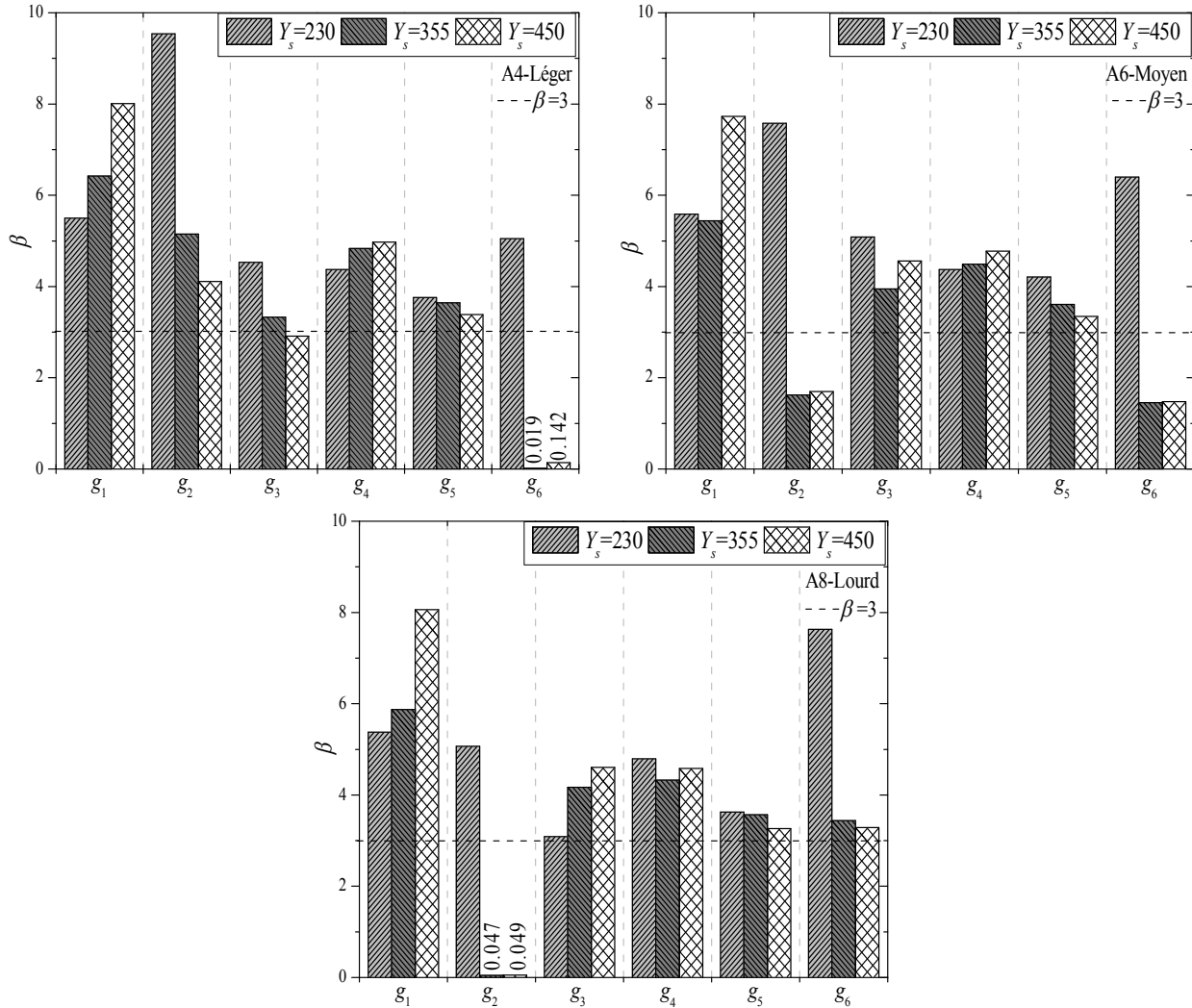
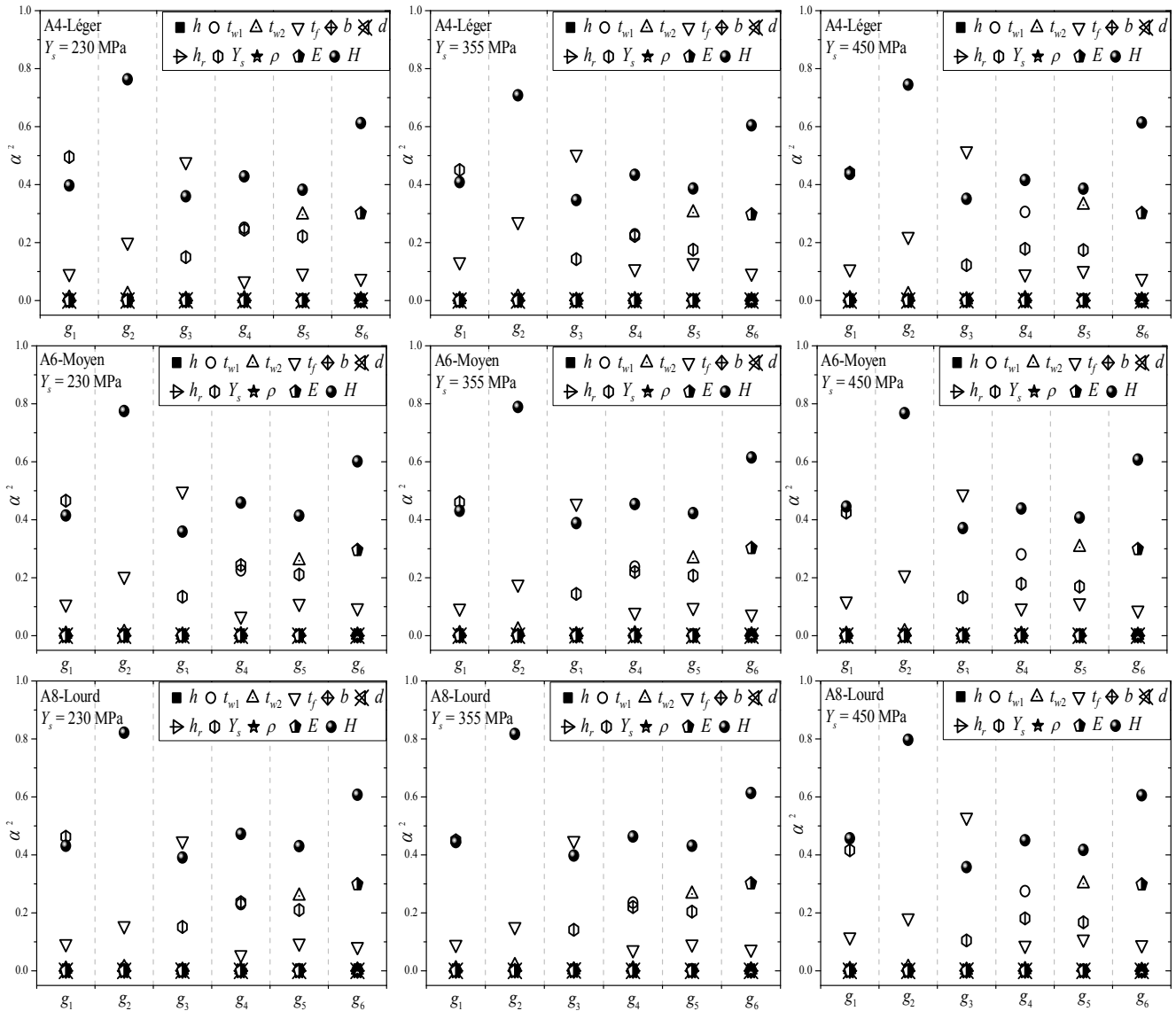


Fig. 5.6 Indice de fiabilité pour chaque équation d'état limite.

Pour le cas des aciers à moyenne et haute résistance ( $Y_s = 355$  et  $450$ MPa respectivement), les équations d'état limite à faibles valeurs d'indice de fiabilité se diffèrent selon la classe du pont roulant considéré. Pour un pont roulant de classe A4-Léger,  $g_6$  a la plus faible valeur de  $\beta$ , pour la classe A6-Moyen, ce sont  $g_2$  et  $g_6$ , tandis que pour la classe A8-Lourd, c'est  $g_2$ . Les deux équations d'état limite  $g_2$  et  $g_6$  qui représentent les critères imposés sur la fatigue et la rigidité (la flèche admissible), respectivement, sont indépendantes de la limite d'élasticité de l'acier utilisé. Par conséquent, les poutres de l'optimisation déterministe ont une forte probabilité de dépasser la flèche admissible, ainsi que l'occurrence des défaillances due à la fatigue.





**Fig. 5.7** Valeurs des cosinus directeurs au carré des variables aléatoires pour chaque équation d'état limite et différentes configurations du pont roulant.

Les cosinus directeurs au carré ( $\alpha^2$ ) du vecteur  $\vec{\beta}$  représentent, suivant une variable, l'influence ou la contribution de cette dernière sur la probabilité de défaillance. Plus la valeur de  $\alpha^2$  d'une variable est élevée, plus son influence est importante. La Fig. 5.7 représente la valeur des cosinus directeurs des variables aléatoires selon chaque équation d'état limite et la configuration du pont roulant. Nous remarquons que la classe du pont roulant et le changement dans la limite élastique de l'acier utilisé ont un effet négligeable sur la variation des cosinus directeurs, par contre, ces derniers sont très sensibles à la nature de l'équation d'état limite. On observe aussi que le chargement maximal que le

pont peut soulever  $H$ , ainsi que l'épaisseur des ailes  $t_f$  ont une contribution importante dans la probabilité de défaillance pour toutes les équations d'état limite.

Le Tableau 5.11 présente une comparaison des résultats d'approximation de la probabilité de défaillance par la méthode de fiabilité du premier ordre (FORM) obtenus par quatre (4) algorithmes différents (HLRF, iHLRF, nHLRF et dBA). On présente la  $P_f$  de chaque équation d'état limite séparément ainsi que celle du système ( $\min(g_i)$ ). Comme on peut le voir, certaines conceptions de la poutre principale obtenues par l'optimisation déterministe ont une probabilité de défaillance très élevée (voir les cases en gris). De plus, nous constatons aussi quelques avantages de la méthode d'estimation de fiabilité basée sur dBA. Dans le cas où la probabilité est très faible, la simulation de Monte Carlo devient très coûteuse et les algorithmes standards (i.e. HLRF) ne convergent pas vers une solution. Notre méthode proposée présente une excellente alternative pour l'estimation de  $P_f$ . A l'instar des méthodes classiques de FORM, dBA a de grandes capacités à déterminer l'indice de fiabilité réel d'un système (voir les chiffres en gras).

Le Tableau 5.12 représente une comparaison des probabilités de défaillances obtenues avec plusieurs approximations de deuxième ordre (SORM), la simulation de Monte Carlo et dBA. On observe qu'à haute probabilité de défaillance, les approximations SORM et FORM (les résultats obtenus par dBA) sont quasi-exactes avec la simulation de Monte Carlo. Quel que soit l'approximation de SORM considérée, elle est sensible à la valeur de l'indice de fiabilité. Comme on a vu au Chapitre 1, les méthodes SORM sont basées sur l'algorithme HLRF pour déterminer le point le plus probable. Cependant, l'algorithme HLRF ne parvient pas à trouver l'optimum, ou rate l'optimum réel et converge vers un optimum local, cette erreur apparaîtra dans l'approximation de la probabilité (voir le Tableau 5.12, Classe A6-Moyen,  $Y_s = 355$  et  $450$  MPa).

### ***5.5.1 Conclusion partielle***

L'analyse de fiabilité montre que l'optimisation déterministe peut conduire à des conceptions surdimensionnées qui sont plus coûteuses, ou mal-dimensionnées dont les risques de défaillance sont très élevés. De plus, les résultats montrent aussi l'efficacité et la robustesse de la méthode d'estimation de la fiabilité par l'algorithme des chauves-souris. Là où les algorithmes classiques échouent, notre méthode basée sur dBA a les capacités à aboutir à des bons résultats, ce qui la rend une excellente méthode alternative.

**Tableau 5.11**

Résultats d'approximation par FORM de la probabilité de défaillance.

Classe	$Y_s$	Méthode	$g_1$	$g_2$	$g_3$	$g_4$	$g_5$	$g_6$	$\min(g_i)$
A4 Léger	230	HLRF	1.9265E-08	N.C.	2.9890E-06	6.2087E-06	8.4579E-05	2.1889E-07	8.4579E-05
		iHLRF	1.9265E-08	N.C.	2.9890E-06	6.2087E-06	8.4579E-05	2.1889E-07	8.4579E-05
		nHLRF	1.9265E-08	N.C.	2.9890E-06	6.2087E-06	8.4579E-05	2.1889E-07	8.4579E-05
		dba	1.9229E-08	7.1882E-22	2.9831E-06	6.1990E-06	8.4532E-05	2.1844E-07	8.4526E-05
	355	HLRF	6.6368E-11	1.3276E-07	4.3820E-04	6.7525E-07	1.3570E-04	4.9205E-01	4.9205E-01
		iHLRF	6.6368E-11	1.3276E-07	4.3820E-04	6.7525E-07	1.3570E-04	4.9205E-01	4.9205E-01
		nHLRF	6.6367E-11	1.3276E-07	4.3820E-04	6.7525E-07	1.3570E-04	4.9205E-01	4.9205E-01
		dba	6.6311E-11	1.3273E-07	4.3776E-04	6.7465E-07	1.3555E-04	4.9205E-01	4.9205E-01
	450	HLRF	5.7447E-16	1.9408E-05	1.7886E-03	3.2800E-07	3.5571E-04	4.4334E-01	4.4334E-01
		iHLRF	5.7447E-16	1.9408E-05	1.7886E-03	3.2800E-07	3.5571E-04	4.4334E-01	4.4334E-01
		nHLRF	5.7447E-16	1.9408E-05	1.7886E-03	3.2800E-07	3.5571E-04	4.4334E-01	4.4334E-01
		dba	5.7421E-16	1.9406E-05	1.7874E-03	3.2735E-07	3.5541E-04	4.4333E-01	4.4334E-01
A6 Moyen	230	HLRF	1.1636E-08	N.C.	1.8158E-07	6.1870E-06	1.2962E-05	7.8253E-11	1.2962E-05
		iHLRF	1.1636E-08	N.C.	1.8158E-07	6.1870E-06	1.2962E-05	7.8253E-11	1.2962E-05
		nHLRF	1.1636E-08	N.C.	1.8158E-07	6.1870E-06	1.2962E-05	7.8253E-11	1.2962E-05
		dba	1.1606E-08	1.7092E-14	1.8131E-07	6.1853E-06	1.2949E-05	7.7994E-11	1.2940E-05
	355	HLRF	2.6715E-08	5.3112E-02	3.9895E-05	3.5465E-06	1.5481E-04	7.2987E-02	5.3112E-02
		iHLRF	2.6715E-08	5.3112E-02	3.9895E-05	3.5465E-06	1.5481E-04	7.2987E-02	5.3112E-02
		nHLRF	2.6715E-08	5.3112E-02	3.9895E-05	3.5465E-06	1.5481E-04	7.2987E-02	5.3112E-02
		dba	2.6698E-08	5.3106E-02	3.9838E-05	3.5410E-06	1.5477E-04	7.2973E-02	<b>7.2971E-02</b>
	450	HLRF	5.3533E-15	4.4781E-02	2.5622E-06	8.9394E-07	4.1018E-04	6.9699E-02	4.4781E-02
		iHLRF	5.3533E-15	4.4781E-02	2.5622E-06	8.9394E-07	4.1018E-04	6.9699E-02	4.4781E-02
		nHLRF	5.3533E-15	4.4781E-02	2.5622E-06	8.9831E-07	4.1018E-04	6.9699E-02	4.4781E-02
		dba	5.3449E-15	4.4779E-02	2.5588E-06	8.9714E-07	4.0955E-04	6.9693E-02	<b>6.9670E-02</b>
A8 Lourd	230	HLRF	3.8566E-08	2.0178E-07	1.0053E-03	8.3064E-07	1.4381E-04	N.C.	1.0053E-03
		iHLRF	3.8566E-08	2.0178E-07	1.0053E-03	8.3064E-07	1.4381E-04	N.C.	1.0053E-03
		nHLRF	3.8566E-08	2.0178E-07	1.0053E-03	8.3064E-07	1.4381E-04	N.C.	1.0053E-03
		dba	3.8525E-08	2.0177E-07	1.0049E-03	8.2913E-07	1.4375E-04	1.1844E-14	1.0049E-03
	355	HLRF	3.8566E-08	2.0178E-07	1.0053E-03	8.3064E-07	1.4381E-04	N.C.	1.0053E-03
		iHLRF	3.8566E-08	2.0178E-07	1.0053E-03	8.3064E-07	1.4381E-04	N.C.	1.0053E-03
		nHLRF	3.8566E-08	2.0178E-07	1.0053E-03	8.3064E-07	1.4381E-04	N.C.	1.0053E-03
		dba	3.8525E-08	2.0177E-07	1.0049E-03	8.2913E-07	1.4375E-04	1.1844E-14	1.0049E-03
	450	HLRF	3.7448E-16	4.8018E-01	2.0286E-06	2.3242E-06	5.4668E-04	5.0726E-04	4.8018E-01
		iHLRF	3.7448E-16	4.8018E-01	2.0286E-06	2.3242E-06	5.4668E-04	5.0726E-04	4.8018E-01
		nHLRF	3.7448E-16	4.8018E-01	2.0286E-06	2.3242E-06	5.4668E-04	5.0726E-04	4.8018E-01
		dba	3.7258E-16	4.8018E-01	2.0219E-06	2.3223E-06	5.4626E-04	5.0696E-04	4.8018E-01

N.C. : Non Convergent

**Tableau 5.12**

Comparaisons entre les probabilités de défaillances obtenues par dBA, SORM et MCS.

Classe	A4-Léger			A6-Moyen			A8-Lourd		
	$Y_s = 230$	355	450	230	355	450	230	355	450
dBA	8.4526E-05	4.9205E-01	4.4334E-01	1.2940E-05	7.2971E-02	6.9670E-02	1.0049E-03	4.8119E-01	4.8018E-01
Breitung	9.0168E-05	4.9208E-01	4.4353E-01	1.3979E-05	5.3253E-02	4.4876E-02	1.0271E-03	4.8123E-01	4.8021E-01
Tvedt	9.2350E-05	4.9481E-01	4.4569E-01	1.4331E-05	5.3386E-02	4.4960E-02	1.0374E-03	4.8245E-01	4.8114E-01
MCS	1.1000E-04	4.9247E-01	4.4506E-01	3.0000E-05	9.3365E-02	8.6310E-02	1.1300E-03	4.8396E-01	4.8351E-01

## 5.6 Optimisation basée sur la fiabilité

### 5.6.1 Formulation du problème d'optimisation basé sur la fiabilité

L'objectif de l'optimisation basée sur la fiabilité ou l'RBDO (Reliability Based-Design Optimization) est de trouver le meilleur compromis entre la fiabilité et la robustesse d'un bien et le coût de fabrication ou de fonctionnement de ce dernier. Dans notre cas, l'objectif est de trouver la section optimale de la poutre principale telle que la probabilité de défaillance ( $P_{fi}$ ) de chaque équation d'état limite ( $g_i$ ) soit inférieure ou égale à une probabilité de défaillance admissible ( $P_{fa}$ ). La formulation mathématique du problème s'écrit comme suit :

$$\left\{ \begin{array}{l} \text{Minimiser } A = x_1(x_2 + x_3) + 2x_4x_5 \\ \text{Tel que: } P(g_i(x, p) < 0) \leq P_{fa}, \quad i = 1, 2, \dots, 6 \\ 500 \leq x_1 \leq 1500 \text{ mm} \\ 3 \leq x_2, x_3, x_4 \leq 25 \text{ mm} \\ 100 \leq x_5 \leq 1000 \text{ mm} \end{array} \right. \quad (5-40)$$

où  $x = [x_1, x_2, x_3, x_4, x_5] = [h, t_{w1}, t_{w2}, t_f, b]$  et  $p$  représente le vecteur des paramètres aléatoires. La description des variables d'optimisation ainsi que les différents paramètres aléatoires et déterministes du problème sont tels qu'ils sont dans le Tableau 5.9. Dans notre étude, nous imposons  $P_{fa} = 0.00135$  ( $\beta = 3$ ), ce qui est équivalent à une fiabilité supérieure à 99.8%.

Le problème d'RBDO de la poutre en caisson soudée est caractérisé par :

- 6 contraintes probabilistes ;
- 5 variables d'optimisation aléatoires à déterminer ;

- Dans les cinq variables d'optimisation, 3 d'entre elles ont un écart type variable qui dépend de la moyenne ;
- Les cinq variables d'optimisation sont discrètes et dont leurs domaines d'appartenance sont tels qu'ils sont définis dans la Section 5.4.1;
- 7 paramètres sont aléatoires et 6 autres sont déterministes.

Pour la résolution d'un tel problème, nous utilisons la méthode RBDO basée sur l'algorithme des chauves-souris présentée dans le Chapitre 4. Les équations d'état limite sont les mêmes que les critères d'optimisation déterministes avec un seul changement, nous imposons un facteur de sécurité égal à 1, donc  $P_s = Y_s$ .

### **5.6.2 Résultats et discussions**

La même configuration des paramètres de l'algorithme des chauves-souris utilisée dans l'optimisation déterministe, est employée pour l'RBDO. La taille de la population des chauves-souris est 100 ainsi que le nombre des itérations est de 1500. Le Tableau 5.13 présente une comparaison entre les résultats d'optimisation basée sur la fiabilité et ceux de l'optimisation déterministe ainsi que ceux de Farkas [171]. On remarque que pour certains cas, et surtout pour  $Y_s = 230\text{MPa}$ , l'optimum obtenu par l'RBDO, quel que soit la classe du pont roulant, est beaucoup meilleur que celui de l'optimisation déterministe avec des économies entre 7 et 11% selon la classe.

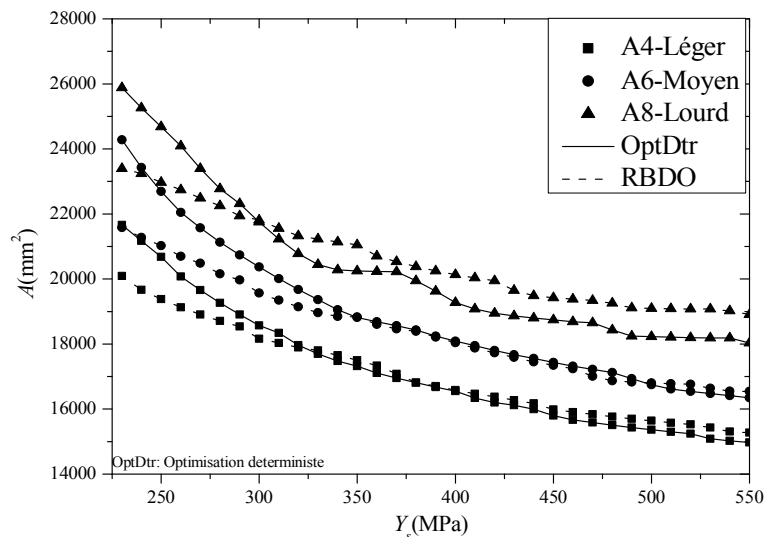
Par contre, lors de l'utilisation des aciers à haute résistance ( $Y_s = 355$  et  $450\text{MPa}$ ), les conceptions optimales obtenues par l'RBDO ont une aire de section supérieure à celles de l'optimisation déterministe. Ceci est dû au fait que les poutres dimensionnées par l'optimisation déterministe ont une forte probabilité de défaillance liée à la fatigue et à la rigidité de la poutre (voir Tableau 5.11 et 12). La Fig. 5.8 présente une comparaison de l'aire optimale obtenue par l'RBDO ainsi que l'optimisation déterministe pour différentes valeurs de la limite d'élasticité ( $Y_s$ ) pour trois classes du pont roulant. Les mêmes observations enregistrées dans les résultats du Tableau 5.13 apparaissent plus clairement dans cette figure.

**Tableau 5.13**

Résultats de l'optimisation basée sur la fiabilité.

Classe	$Y_s$ (MPa)	$h$ (mm)	$t_{w1}$ (mm)	$t_{w2}$ (mm)	$b$ (mm)	$t_f$ (mm)	$A$ (mm <sup>2</sup> )	Réf. et Méthodes
A4-Léger	230	1050	6	6	400	14	23800	[171]
		1206	6	5	420	10	21666	OptDtr
		1090	6	5	405	10	20090	RBDO
	355	950	5	5	375	14	20000	[171]
		914	5	4	448	10	17186	OptDtr
		1122	4	4	422	10	17416	RBDO
450	1050	5	5	375	10	18000	[171]	
	1014	5	4	417	8	15798	OptDtr	
	1241	4	4	433	7	15990	RBDO	
A6-Moyen	230	1150	7	7	375	14	26600	[171]
		1194	6	5	466	12	24318	OptDtr
		1006	6	5	438	12	21578	RBDO
	355	1050	6	6	325	14	21700	[171]
		1023	6	5	375	10	18753	OptDtr
		1226	5	5	406	8	18756	RBDO
450	1000	5	5	325	16	20400	[171]	
	1022	5	4	412	10	17438	OptDtr	
	1131	4	4	415	10	17348	RBDO	
A8-Lourd	230	1150	7	7	450	14	28700	[171]
		1292	7	5	519	10	25884	OptDtr
		1123	6	5	460	12	23393	RBDO
	355	1000	6	6	325	18	23700	[171]
		1125	6	5	393	10	20235	OptDtr
		1138	4	4	587	10	20844	RBDO
450	1050	5	5	425	14	22400	[171]	
	1111	5	4	437	10	18739	OptDtr	
	1396	4	4	516	8	19424	RBDO	

OptDtr: Optimisation déterministe



**Fig. 5.8** Comparaison de l'aire optimale de la section transversale de la poutre obtenue par l'RBDO et l'optimisation déterministe.

**Tableau 5.14**

Comparaison de l'indice de fiabilité pour chaque équation d'état limite.

Classe	$Y_s(\text{MPa})$	$g_1$	$g_2$	$g_3$	$g_4$	$g_5$	$g_6$	
A4-Léger	230	5.4975	9.5393	4.5272	4.3701	3.7611	5.0518	OptDtr
		4.0073	7.8245	3.4333	3.7293	<b>3.0036</b>	<b>3.0135</b>	RBDO
	355	6.4240	5.1464	3.3275	4.8322	3.6412	0.0199	OptDtr
		8.0334	7.3593	5.2806	<b>3.0032</b>	3.4979	<b>3.0115</b>	RBDO
	450	8.0098	4.1144	2.9132	4.9740	3.3851	0.1425	OptDtr
		9.5933	6.4687	3.0848	<b>3.0006</b>	3.2950	<b>3.0152</b>	RBDO
A6-Moyen	230	5.5857	7.5814	5.0873	4.3709	4.2066	6.3989	OptDtr
		3.1847	4.4524	3.3659	3.4239	<b>2.9997</b>	<b>3.0328</b>	RBDO
	355	5.4395	1.6154	3.9450	4.4909	3.6071	1.4539	OptDtr
		6.6716	3.4115	<b>2.9905</b>	<b>3.0022</b>	3.4799	3.7950	RBDO
	450	7.7306	1.6977	4.5596	4.7751	3.3458	1.4780	OptDtr
		8.6425	<b>3.0130</b>	5.3344	<b>3.0049</b>	3.4503	<b>3.0565</b>	RBDO
A8-Lourd	230	5.3737	5.0673	3.0887	4.7908	3.6262	7.6284	OptDtr
		3.7879	<b>3.0054</b>	3.7219	3.2069	<b>3.0076</b>	5.2429	RBDO
	355	5.8717	0.0472	4.1700	4.3259	3.5659	3.4358	OptDtr
		7.9435	<b>3.0907</b>	<b>3.0024</b>	<b>3.0027</b>	4.0197	5.2004	RBDO
	450	8.0623	0.0497	4.6084	4.5801	3.2653	3.2865	OptDtr
		10.146	<b>3.0011</b>	3.6178	<b>3.0001</b>	3.6103	6.7710	RBDO

OptDtr: Optimisation déterministe

Le Tableau 5.14 représente une comparaison de l'indice de fiabilité de chaque équation d'état limite. Les résultats sont obtenus par la méthode de fiabilité basée sur dBA proposée dans le Chapitre 3, avec les mêmes paramètres des algorithmes utilisés dans la Section 5.5. A l'instar de l'optimisation déterministe, les conceptions obtenues par l'RBDO ont toutes un indice de fiabilité supérieur à 3, à l'exception du pont roulant de classe A6-Moyen construit avec un acier dont la limite d'élasticité est de 355MPa. Pour ce dernier, nous avons enregistré un indice de fiabilité  $\beta = 2.9905$ . Ceci est dû aux erreurs de calcul accumulées dû à l'arrondissement et l'approximation des dérivées par les différences finies. Les chiffres en gras représentent les contraintes actives, tandis que les cases à fond gris représentent les contraintes dont l'indice de fiabilité est inférieur à 3.

Le Tableau 5.15 présente la probabilité de défaillance des conceptions obtenues par l'RBDO pour différentes classes et aciers. Comme on le voit, les probabilités de défaillance estimées par FORM de toutes les conceptions sont inférieures ou égales à la probabilité admissible ( $P_{fa}$ ) sauf pour la poutre de classe A6 et d'acier dont  $Y_s = 355\text{MPa}$ . La probabilité de défaillance de cette dernière est légèrement supérieure à  $P_{fa}$ . Comme on l'a expliqué dans le paragraphe précédent, ceci est dû aux erreurs de calculs. La comparaison avec SORM et MCS montre l'existence des écarts dans

l'estimation de  $P_f$ . Ces derniers sont dus au fait que la méthode d'RBDO à boucle unique basée sur le concept de l'espace fiable qu'on a développé dans le Chapitre 4, a un ordre de précision équivalent à celui de FORM.

**Tableau 5.15**

Probabilité de défaillance des conceptions obtenues par l'RBDO.

Classe	A4-Léger			A6-Moyen			A8-Lourd			
	$Y_s = 230$	355	450	230	355	450	230	355	450	
FORM	HLRF	0.00133	0.00134	0.00135	0.00135	0.00139	0.00129	0.00133	0.00100	0.00135
	iHLRF	0.00133	0.00134	0.00135	0.00135	0.00139	0.00129	0.00133	0.00100	0.00135
	nHLRF	0.00133	0.00134	0.00135	0.00135	0.00032	0.00129	0.00133	0.00100	0.00135
	dBA	0.00133	0.00134	0.00135	0.00135	0.00139	0.00133	0.00133	0.00134	0.00135
SORM	Breitung	0.00141	0.00142	0.00142	0.00143	0.00137	0.00130	0.00133	0.00100	0.00135
	Tvedt	0.00145	0.00145	0.00145	0.00146	0.00139	0.00130	0.00133	0.00100	0.00135
MCS		0.00285	0.00285	0.00276	0.00379	0.00324	0.00297	0.00276	0.00260	0.00322

**Tableau 5.16**

Valeurs déterministes des critères d'optimisation pour deux valeurs du facteur de sécurité.

Classe	$Y_s$	FS*	$g_1$	$g_2$	$g_3$	$g_4$	$g_5$	$g_6$
A4-Léger	230	0.59	<b>-0.1544</b>	0.4019	<b>-0.0155</b>	<b>-0.0610</b>	<b>-0.0714</b>	0.3168
		1.00	0.3189	0.4019	0.3079	0.2967	0.2404	0.3168
	355	0.59	0.2385	0.3875	0.2308	<b>-0.0649</b>	0.0070	0.3183
		1.00	0.5507	0.3875	0.4496	0.2444	0.2790	0.3183
	450	0.59	0.3536	0.3447	0.0317	<b>-0.0457</b>	0.0019	0.3163
		1.00	0.6186	0.3447	0.2868	0.2444	0.2670	0.3163
A6-Moyen	230	0.59	<b>-0.2387</b>	0.2728	<b>-0.1031</b>	<b>-0.1588</b>	<b>-0.0667</b>	0.3204
		1.00	0.2692	0.2728	0.3124	0.2854	0.2466	0.3204
	355	0.59	0.1312	0.2105	<b>-0.0106</b>	<b>-0.0718</b>	<b>-0.0017</b>	0.3801
		1.00	0.4874	0.2105	0.2757	0.2511	0.2854	0.3801
	450	0.59	0.3014	0.1896	0.2462	<b>-0.0461</b>	0.0207	0.3221
		1.00	0.5878	0.1896	0.4599	0.2499	0.2818	0.3221
A8-Lourd	230	0.59	<b>-0.1638</b>	0.2037	<b>-0.0362</b>	<b>-0.1265</b>	<b>-0.0631</b>	0.4880
		1.00	0.3133	0.2037	0.3409	0.2724	0.2480	0.4880
	355	0.59	0.2600	0.2159	0.0260	<b>-0.0710</b>	0.0670	0.4877
		1.00	0.5634	0.2159	0.2947	0.2588	0.3263	0.4877
	450	0.59	0.4109	0.2052	0.1021	<b>-0.0471</b>	0.0415	0.5775
		1.00	0.6524	0.2052	0.3382	0.2536	0.2961	0.5775

\*FS: Facteur de sécurité.

Le Tableau 5.16 présente la valeur des équations d'état limite pour chaque conception obtenue par l'RBDO. Nous avons évalué ces dernières en fonction de deux valeurs du facteur de sécurité ( $FS=0.59$  et  $1$ ). Les équations d'état limite ( $g_i$ ) ont été écrites sous forme  $g = 1-S/R$  où  $S$  représente



les sollicitations et  $R$  la résistance du système. Cette forme appelée aussi forme standardisée ou normalisée représente un rapport ou un pourcentage de combien est la valeur de sollicitation appliquée par rapport à la résistance. Si par exemple la valeur de  $g_i = +0.25$ , ceci veut dire que les sollicitations sont inférieures à la résistance du système de 25%, par contre, si la même valeur de  $g_i$  est négative, les sollicitations sont supérieures à la résistance du système de 25 % de plus. Si on prend  $g_4$ , par exemple, qui représente le critère imposé sur le flambement de l'âme principale dont la formulation est basée sur la valeur de la contrainte admissible (Eqs 5-2 et 5-19), on voit que les conceptions d'RBDO violent toutes cette contrainte pour un facteur de sécurité  $FS = 0.59$ . Par contre, pour un facteur de sécurité  $FS = 1$ , on voit que les valeurs de  $g_4$  tournent autour de 0.25, ce qui signifie que les contraintes appliquées sont moins de 25% que la contrainte critique réelle du flambement de l'âme.

### **5.6.3 Conclusion partielle**

Dans cette partie, nous avons réussi à appliquer la méthode d'RBDO basée sur l'algorithme des chauves-souris développée dans le Chapitre 4 pour optimiser la poutre principale d'un pont roulant en se basant sur sa fiabilité. La conception a été considérablement améliorée, que ce soit la rigidité, la fiabilité ou la durée de vie de la poutre, avec une réduction (ou au moins aucune augmentation significative) dans son poids global.

## **5.7 Conclusions**

Dans ce chapitre, nous avons traité l'analyse de fiabilité et l'optimisation basée sur la fiabilité d'une poutre en caisson soudée, d'un pont roulant à bipoutres. Dans une première partie, nous avons utilisé l'algorithme des chauves-souris directionnelles (dBA) pour optimiser le poids de la poutre selon les codes BS 2573 et BS 5400. Les résultats montrent que l'emploi de l'algorithme dBA résulte des conceptions plus efficaces, et l'utilisation des aciers à haute résistance réduit significativement les poids de la poutre. L'analyse de fiabilité a révélé que l'optimisation déterministe basée sur le facteur de sécurité, peut engendrer des poutres surdimensionnées ou mal-dimensionnées (des poutres avec de fortes probabilités de défaillances dues à la fatigue et l'accumulation des déformations plastiques causées par l'excès de la flexion). Pour contourner ce problème, nous avons employé le concept de l'optimisation basée sur la fiabilité. Les résultats montrent que les poutres dimensionnées selon ce concept sont très fiables comparés à ceux issus de l'optimisation déterministe.

Nous notons aussi que l'algorithme des chauves-souris directionnelles présenté dans ce travail ainsi que les méthodes d'estimation de fiabilité et d'optimisation basée sur la fiabilité construites à la base de ce dernier, sont très fiables, robustes et pratiques.

# **Chapitre 6**

## **Proposition d'un système de mouvement autonome avec évitement des obstacles pour un pont roulant basé sur dBA**

*L'intelligence artificielle pourrait mettre fin à la race humaine (Stephen Hawking)*  
*Il faut avoir moins peur de l'intelligence artificielle que de la stupidité naturelle (Joël de Rosnay)*

## 6.1 Introduction

Les ponts roulants existent dans plusieurs industries dans le monde. Ces ponts fonctionnent dans des conditions de travail différentes, y compris certaines qui sont dangereuses pour l'être humain. La grande majorité des ponts roulants sont contrôlés directement par des opérateurs humains. Toutefois, dans certains cas, il est nécessaire d'enlever l'opérateur humain à partir des conditions de fonctionnement dangereuses, ou dû au manque des opérateurs qualifiés pour assurer la sécurité et l'efficacité du travail. Par conséquent, le développement d'un pont roulant autonome est une nécessité.

Le développement des ponts roulants autonomes a fait l'objet de plusieurs travaux de recherche. Terashima et Suzuki [190] ont proposé un système de navigation et de planification de trajectoire en 3D pour un pont roulant, en se basant sur une stratégie dérivée de la résolution de l'équation de la diffusion. Cette dernière traduit la migration des espèces chimiques dans un milieu. L'idée de base est d'assimiler le mouvement du pont roulant à la diffusion des éléments chimiques. Cette méthode est une méthode *off-line* et qui nécessite la connaissance au préalable de l'environnement du travail du pont roulant. Mandelli et Haider [191] ont proposé une méthode pour convertir des anciens ponts roulants d'une cimenterie qui existe aux USA, à des ponts totalement automatisés. Le suivi de ces derniers pour une période de six (6) ans a montré une diminution de 30% des coûts de maintenance.

Akamatsu et *al.* [192] ont proposé une méthode *on-line* pour la planification des trajectoires. Les auteurs ont utilisé la méthode du potentiel qui est basée sur la résolution de l'équation de la diffusion, tout en ajoutant un système composé des détecteurs ultrasoniques pour détecter les petits changements du milieu. Omar et *al.* [193] ont proposé un système de contrôle par logique floue. Ce

dernier imite la pensée d'un opérateur expert des ponts roulants, pour le réglage des paramètres de contrôle. Wecker et *al.* [194] ont développé un système de détection des obstacles en utilisant trois caméras installés au niveau du crochet, pour permettre au pont roulant de se déplacer sans risque de collision. Miyoshi et *al.* [195] ont présenté une méthode de planification des trajectoires en considérant la rotation des charges. Cette technique s'avère très utile lors de la manutention des objets longs tels que les conduits, les profilés, les conteneurs...etc.

Kaneshige et *al.* [196] ont proposé une amélioration de la méthode de contrôle développée par [192] en incluant un système de suppression du balancement des charges pour permettre le transport et la manutention des conteneurs ouverts remplis de liquide. Smoczek et *al.* [197] ont construit un système de reconnaissance d'obstacle en utilisant deux (2) caméras pour créer une reconstruction en 3D de l'environnement, puis ils ont utilisé l'algorithme de recherche A\* (qui se prononce A étoile, ou A star en anglais) pour contourner les obstacles. Yang J.S. et *al.* [198] ont développé un système vision pour l'évitement des obstacles, en utilisant des caméras de surveillance placés au niveau du chariot. Ce système permet de reconnaître la charge et de créer ou de délimiter une zone de sécurité. Si, lors du mouvement du pont, un objet ou obstacle entre dans la zone délimitée, le système envoie un signal d'alarme et s'arrête.

Les algorithmes métaheuristiques ont été considérablement utilisés pour développer des robots mobiles autonomes avec évitement des obstacles comme l'algorithme génétique [199], la méthode d'optimisation par essaim de particules (PSO) [200, 201] et plus récemment l'algorithme de l'herbe envahissante (IWO) [202]. Sivakumar et *al.* [203] ont appliqué l'algorithme A\* (A étoile) pour la planification de la trajectoire de deux (2) grues qui travaillent en coopération. Ai et *al.* [204] ont résolu le même problème en utilisant l'algorithme génétique. Wang X. et *al.* [205] ont utilisé l'algorithme de la colonie des fourmis pour la planification des trajectoires sans risque de collision d'une grue mobile dans un chantier complexe. Malgré leur succès incontestable dans le contrôle des robots et des grues, l'application de ces algorithmes pour l'automatisation du pont roulant reste un domaine de recherche inexploité.

Dans ce chapitre, nous développons une méthode pour rendre le mouvement du pont roulant autonome avec un mécanisme d'évitement des obstacles *on-line*, en se basant sur l'algorithme des chauves-souris directionnelles (dBA).

## 6.2 Stratégie proposée

Premièrement, pour faciliter les simulations, nous supposons que la hauteur du hangar est limitée ce qui réduira les calculs à des simulations en 2D. Nous ne considérons que les mouvements selon la largeur et la longueur. Deuxièmement, nous supposons que le système de contrôle du pont roulant n'a pas des connaissances préliminaires sur la position des obstacles et l'environnement du travail. Cependant, la reconnaissance des obstacles se fait on-line via des capteurs de proximité et sur une périphérie limitée. Nous supposons aussi que seules les positions du départ et d'arrivée sont connues. De plus, pour des raisons purement numériques et pour faciliter les simulations, nous supposons que le chariot se déplace pas à pas dans l'espace de travail, et la trajectoire est l'ensemble constitué de ces points  $\{P_d, P_1, P_2, \dots, P_n, P_a\}$ , où  $P_d$  et  $P_a$  représentent les point de départ et le point d'arrivée respectivement, tandis que les  $P_i$  ( $i=1\sim n$ ) sont les positions du chariot lors de son déplacement.

Naturellement, le chemin optimal et le plus court entre deux points est une droite. Le chariot du pont roulant se déplace de sa position actuelle vers sa nouvelle position en suivant une trajectoire rectiligne. Si on fixe notre repère  $(Oxy)$  dans un coin du hangar, la formule de déplacement du chariot est comme suit :

$$\begin{cases} P_{i+1} = P_i + \text{sign}(P_a - P_i) \begin{pmatrix} \sin(\alpha) \\ \cos(\alpha) \end{pmatrix} R \\ tq : P_a = \begin{pmatrix} x_a \\ y_x \end{pmatrix} \text{ et } P_i = \begin{pmatrix} x_i \\ y_i \end{pmatrix} \\ tg(\alpha) = \begin{pmatrix} x_a - x_i \\ y_a - y_i \end{pmatrix} \end{cases} \quad (6-1)$$

où  $R$  représente le pas.

Lors de son mouvement, le pont roulant est équipé d'un système de détection d'obstacle sur un rayon ( $R_0$ ), qui décrit un périmètre appelé *le périmètre de sécurité*. Pour que le pont roulant se déplace sans risque de collision, le pas  $R$  doit être inférieur à  $R_0$ . Si un obstacle  $P_o(x_o, y_o)$  entre dans le périmètre de sécurité, un signal est envoyé au système de contrôle pour correction de la trajectoire. Le mécanisme de correction de la trajectoire est un problème d'optimisation sous contrainte, où la

fonction objectif est de trouver la prochaine position  $P_{i+1}$  dont la distance vers la cible est minimale, tout en gardant une distance supérieure ou égale à  $R_0$  par rapport à l'obstacle. Cependant, le problème d'optimisation est défini comme suit :

$$\left\{ \begin{array}{l} \text{Minimiser } d_a = \sqrt{(x_a - x_{i+1})^2 + (y_a - y_{i+1})^2} \\ \text{Tel que:} \\ \min(d_{oj}) = \min\left(\sqrt{(x_{oj} - x_{i+1})^2 + (y_{oj} - y_{i+1})^2}\right) \geq R_0 \\ x_i - R \leq x_{i+1} \leq x_i + R \\ y_i - R \leq y_{i+1} \leq y_i + R \end{array} \right. \quad (6-2)$$

L'indice  $i$  représente la position actuelle, tandis que l'indice  $i+1$  représente la position du prochain mouvement. L'indice  $j$  représente l'obstacle.

La résolution d'un tel problème est réalisée par dBA dans un espace de recherche à l'intérieur du périmètre de sécurité appelé *espace de recherche sécurisé*. Nous avons opté pour un espace de recherche rectangulaire de cote égale à  $2R$  (voir Eq. 6-2), pour simplifier le problème d'optimisation et réduire le temps de calcul (i.e. si on avait défini un espace de recherche circulaire de rayon  $R$ , nous aurions besoin d'ajouter une deuxième contrainte d'optimisation, ce qui augmente le temps de calcul). Cependant, pour permettre au pont roulant de se déplacer sans risque de collision, nous avons proposé cet organigramme illustré dans la Fig. 6.1 pour contrôler les mouvements du pont.

Afin de mieux comprendre le mécanisme d'évitement d'obstacle, la Fig. 6.2 représente un schéma hypothétique du système de contrôle proposé dans ce chapitre. Au départ, l'utilisateur fournit les positions de départ et d'arrivée. Le pont roulant commence à se déplacer vers le point d'arrivée (cible), en suivant une trajectoire rectiligne. Si, lors de son déplacement, un obstacle est détecté à la périphérie du périmètre de sécurité, le système de correction de trajectoire est enclenché. Dans cette phase, le système de correction de la trajectoire emploie dBA pour trouver le point et la direction du prochain mouvement. Cette opération est répétée tant qu'elle est nécessaire jusqu'à ce que le pont roulant puisse contourner l'obstacle. Lorsque l'obstacle sort du périmètre de sécurité, le pont roulant continue son mouvement en suivant une trajectoire rectiligne vers la cible.

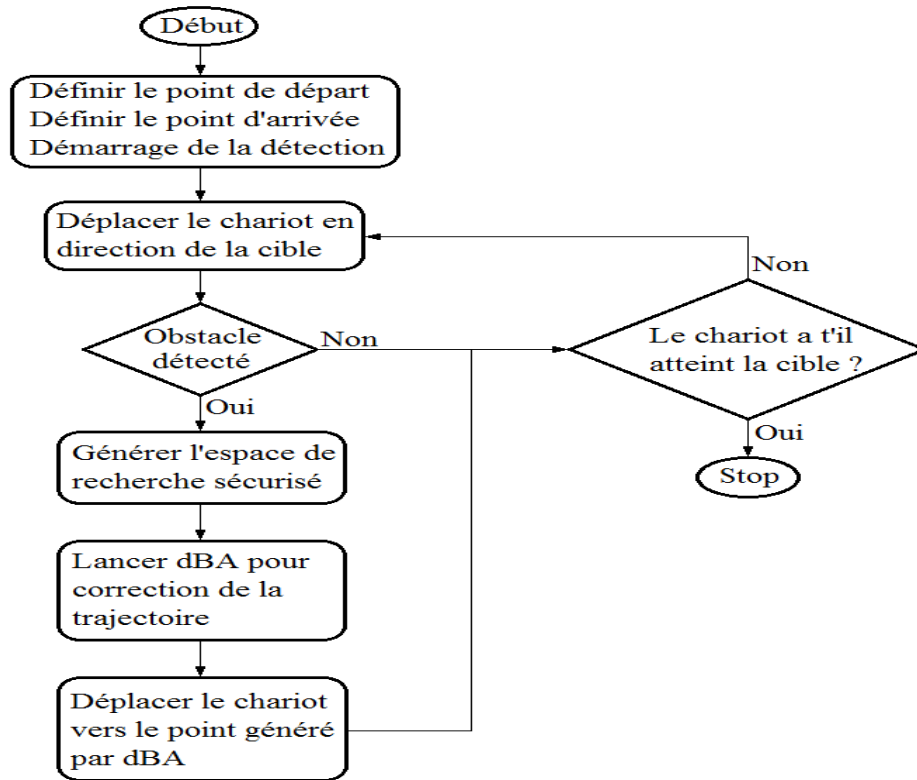


Fig. 6.1 Organigramme du système de contrôle du mouvement proposé.

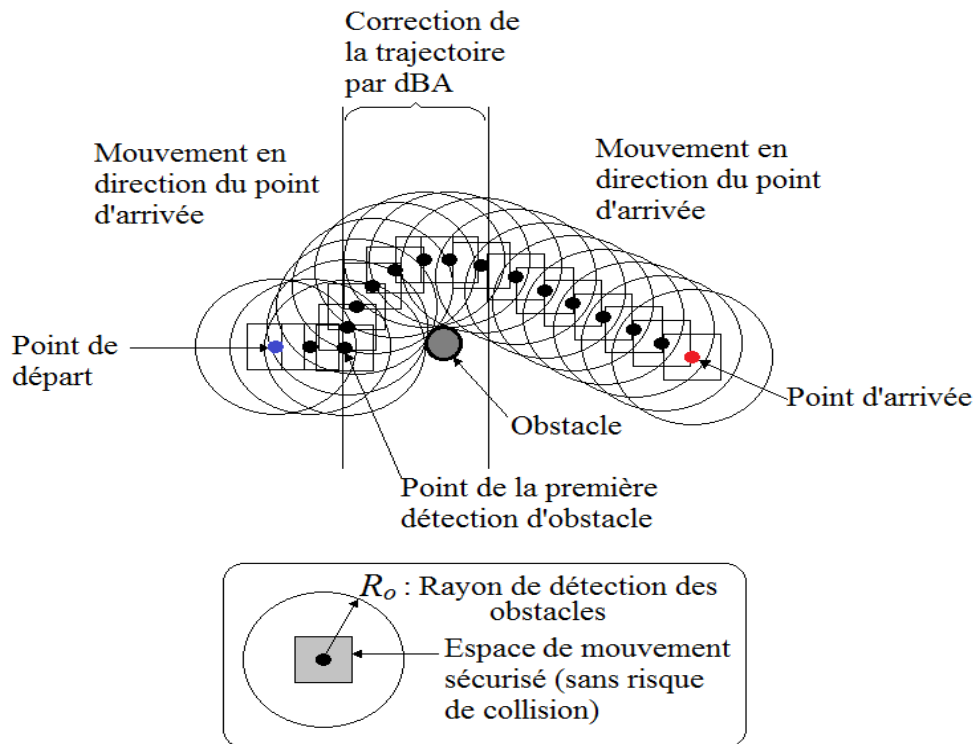


Fig. 6.2 Schéma hypothétique du mécanisme d'évitement d'obstacle.



### 6.3 Simulation, résultats et discussions

Considérons un hangar de largeur ( $X$ ) de 22.5m et de longueur ( $Y$ ) égale à 50m. Pour simuler les mouvements du pont roulant, nous supposons que nous voulons déplacer un objet de sa position actuelle  $P_d$  ( $x_d = 1, y_d = 2$ ) vers un autre coin du hangar, à savoir  $P_a$  ( $x_a = 20, y_a = 45$ ). La distance entre les deux points  $P_d$  et  $P_a$  est 47.01m. Dans la première phase, nous allons mettre un obstacle dans le point  $P_{o1}$  pour examiner le comportement de l'algorithme proposé, puis nous ajoutons un deuxième obstacle  $P_{o2}$  dans le passage du chariot selon la nouvelle trajectoire corrigée. Dans la troisième phase et pour augmenter la complexité du milieu, nous ajoutons 5 nouveaux obstacles posés intentionnellement dans le chemin du pont roulant. La position de chaque obstacle est indiquée dans le Tableau 6.1.

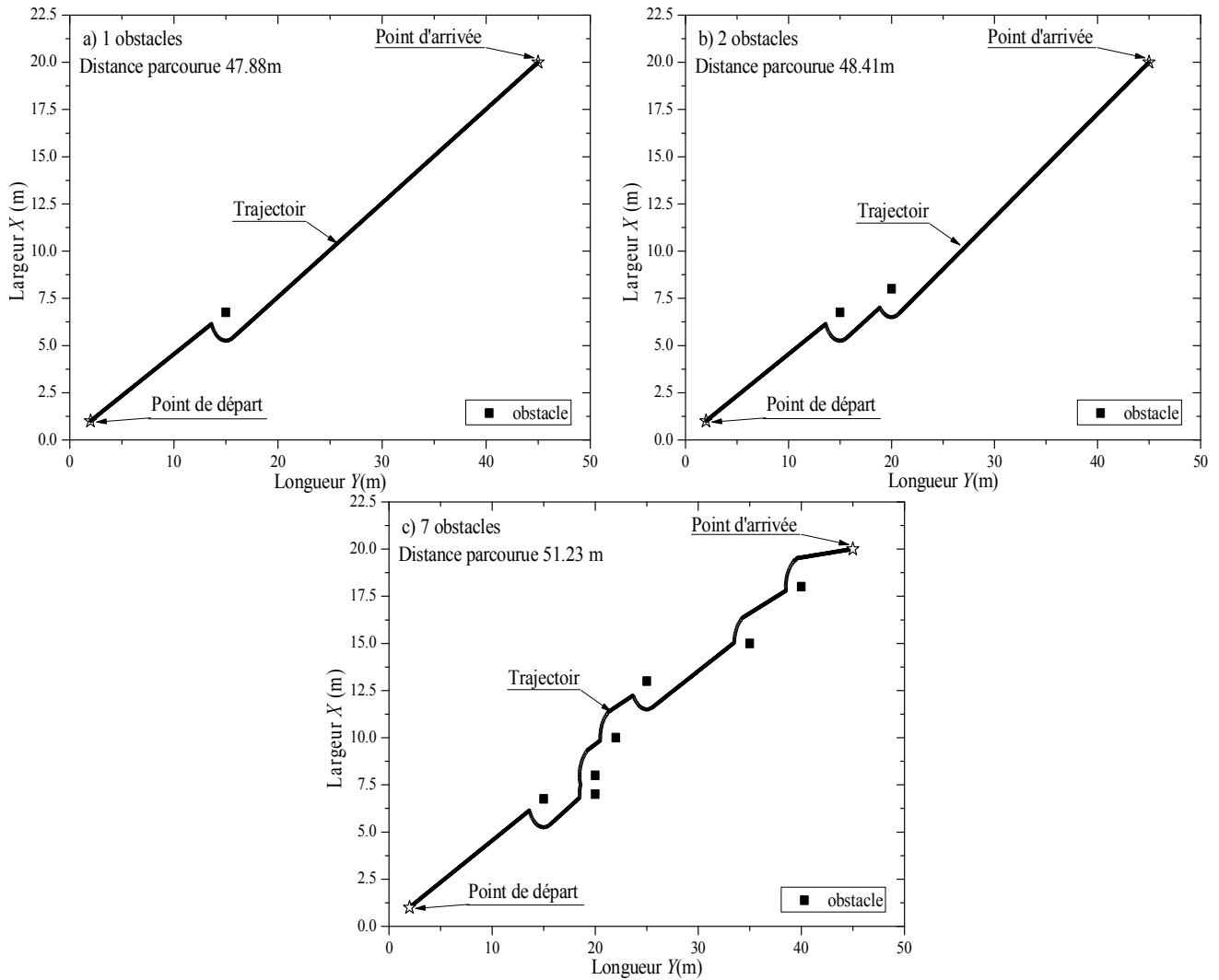
Pour le réglage des paramètres de dBA, nous utilisons les mêmes paramètres proposés dans le Chapitre 2, Section 2.5.1 ( $r_0 = 0.1, r_\infty = 0.7, A_0 = 0.9, A_\infty = 0.6, f_{min} = 0$  et  $f_{max} = 2$ ). Nous considérons une population de dix chauves-souris ( $N = 10$ ) et un nombre d'itérations maximal  $t_{max} = 20$ . De plus, nous considérons que l'objet qu'on va déplacer a un mètre de diamètre (1m), par conséquent, pour permettre au pont roulant de se déplacer sans risque de collision, nous imposons  $R_0 = 1.5m$  et  $R = 0.1m$ .

La Fig. 6.3 (a) représente la trajectoire du pont roulant avec un seul obstacle. On voit bien que l'algorithme proposé pour l'automatisation du pont roulant peut contourner un obstacle sans avoir une connaissance au préalable de l'environnement. La distance parcourue par le pont est supérieure de 87cm à la distance entre  $P_d$  et  $P_a$ . Dans la Fig. 6.3 (b), un deuxième obstacle est ajouté délibérément dans le chemin du pont roulant, on voit bien que l'algorithme le contourne facilement. Dans la Fig. 6.3 (c), une multitude d'obstacles sont imposés dans le chemin du pont roulant, on voit bien que le pont a pu se déplacer sans risque de collision avec ces derniers.

**Tableau 6.1**

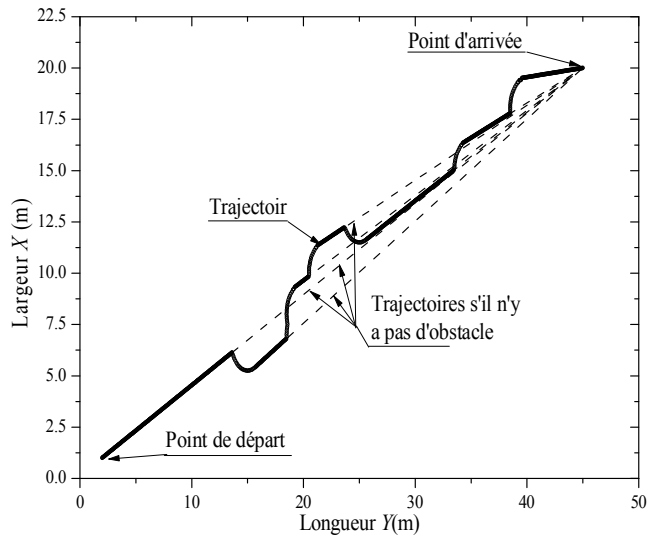
Position des obstacles.

Coordonnées	$P_{o1}$	$P_{o2}$	$P_{o3}$	$P_{o4}$	$P_{o5}$	$P_{o6}$	$P_{o7}$
$x_{oi}$ (m)	6.75	8	7	10	13	15	18
$y_{oi}$ (m)	15	20	20	22	25	35	40

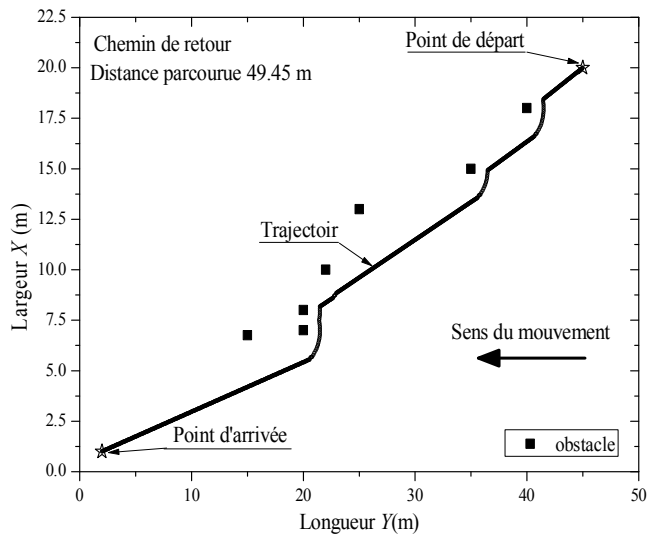


**Fig. 6.3** Trajectoires décrites par le mouvement du pont roulant.

La Fig. 6.4 représente les trajectoires présumées s'il n'y avait pas d'obstacles, ou après le contournement d'un obstacle. On voit bien que chaque fois que le pont roulant contourne un obstacle, il se déplace d'une manière rectiligne vers la cible (le point d'arrivée). La Fig. 6.5 représente le chemin de retour, on voit que ce dernier est différent du chemin d'aller et qu'il est plus court. Ceci est dû au fait que les obstacles ont été intentionnellement posés sur la trajectoire du pont à chaque fois qu'il contourne un obstacle pour inciter le mécanisme d'évitement des obstacles. Il est impossible que les deux trajectoires soient les mêmes, car l'algorithme proposé ne prend pas en considération la topologie complète de l'environnement, seul des détecteurs de proximité sont utilisés pour détecter les obstacles lors du mouvement du pont roulant.



**Fig. 6.4** Trajectoires présumées sans l'existence des obstacles.



**Fig. 6.5** Trajectoire du retour dans le même environnement.

## 6.4 Conclusions

Dans ce chapitre, nous avons utilisé l'algorithme des chauves-souris directionnelles pour développer une technique d'optimisation de la trajectoire d'un pont roulant avec évitement d'obstacle. Cette technique permet de rendre le pont roulant autonome, ainsi nous réduisons les coûts de fonctionnement tout en améliorant l'efficacité du travail. Les résultats montrent que cette technique permet au pont de contourner les obstacles facilement. De plus, du point de vue application, cette

technique est moins coûteuse que d'autres existant dans la littérature, car elle ne nécessite pas une connaissance au préalable de l'environnement de travail (une connaissance au préalable de l'environnement nécessite une cartographie générale de tout le milieu de travail avec mise à jour à chaque fois que ce dernier change, par la suite un équipement sophistiqué doit être installé). Des détecteurs de proximité, que ce soit par ultrason ou par laser ou plus encore des caméras, peuvent être installés selon le budget alloué. De plus, due au faible nombre de la population des chauves-souris et le nombre d'itérations, cette méthode peut être implémentée sur des processeurs de bon marché.

# **Conclusion générale et perspectives**

*Petit à petit je me débarrasse de ma  
méconnaissance et je me regarde devant la  
connaissance infinie*

### ***Conclusion générale***

Dans ce travail, nous avons introduit une nouvelle version améliorée de l'algorithme des chauves-souris, appelé l'algorithme des chauves-souris directionnelles (dBA). L'idée de base est d'introduire le phénomène d'écholocation directionnel pour améliorer les capacités d'exploration et d'exploitation de l'algorithme. Pour ce faire, quatre modifications ont été intégrées afin d'augmenter les performances de BA. Les résultats de simulation et des tests statistiques non-paramétriques ont montré la supériorité de dBA par rapport à une sélection d'algorithmes standards, sophistiqués et certaines variantes de l'algorithme des chauves-souris standard qui existent dans la littérature.

Par la suite, une nouvelle méthode d'estimation de la fiabilité des structures basée sur l'algorithme des chauves-souris directionnelles est présentée. Un nouvel coefficient de pénalité adaptatif dédié à la fiabilité des structures a été proposé pour améliorer la certitude de l'algorithme dans l'estimation de l'indice de fiabilité. Les résultats montrent que cette méthode basée sur dBA est très efficace par rapport à des algorithmes standards, tel que l'algorithme génétique et l'optimisation par essaim de particules (PSO), et la qualité des résultats est semblable aux méthodes classiques, à savoir la simulation de Monte Carlo.

En se basant sur le concept de l'espace fiable et le niveau de comparaison  $\varepsilon$ , dBA a été adapté à réaliser des optimisations fiables. La méthode proposée peut supporter des contraintes déterministes et probabilistes, ainsi que des variables de conception continues et discrètes. Les résultats montrent que la méthode d'RBDO basée sur dBA est fiable, efficace, robuste et plus

générale que les méthodes classiques, qui sont conçues pour ne résoudre que certaines classes de problème.

En s'inspirant des méthodes utilisées pour la conception des robots autonomes, une nouvelle méthode basée sur dBA a été proposée pour automatiser les mouvements du pont roulant. Cette méthode nous a permis de concevoir un pont roulant complètement autonome avec un système d'évitement des obstacles.

L'application de ces méthodes développées dans cette thèse pour résoudre les problèmes d'optimisation, d'analyse de fiabilité et d'optimisation fiabiliste de la poutre principale d'un pont roulant à bipoutre ainsi que l'automatisation de ce dernier, montre que ces méthodes peuvent avoir une application directe dans le domaine industriel.

### *Perspectives*

On a vu que l'usage de l'écholocation directionnelle suivant deux directions a permis d'améliorer les performances de l'algorithme des chauves-souris d'une manière significative. Il est intéressant d'étudier l'écholocation multi-directions et son application à la résolution des problèmes à large dimension. De plus, il peut être très utile d'appliquer dBA pour la résolution des problèmes issus du monde réel.

La méthode d'estimation de la fiabilité par dBA s'avère très fiable, robuste et efficace. Un intéressant axe de recherche est de coupler notre méthode avec des codes de calcul par éléments finis via les surfaces de réponse, ou les réseaux de neurones artificiels, pour résoudre des problèmes plus complexes. Pour l'optimisation fiabiliste basée sur dBA, il est très intéressant d'analyser ces capacités à optimiser de larges structures réelles. Pour la poutre principale du pont roulant, une étude expérimentale sur la résistance des poutres conçues par l'optimisation fiabiliste sera très importante.

## Références

- [1] S.-K. Choi, R. Grandhi, and R. A. Canfield, *Reliability-based structural design*: Springer Science & Business Media, 2006.
- [2] A. Borri and E. Speranzini, "Structural reliability analysis using a standard deterministic finite element code," *Structural Safety*, vol. 19, pp. 361-382, 1997.
- [3] B. D. Youn, Z. Xi, and P. Wang, "Eigenvector dimension reduction (EDR) method for sensitivity-free probability analysis," *Structural and Multidisciplinary Optimization*, vol. 37, pp. 13-28, 2008.
- [4] D. Yang, "Chaos control for numerical instability of first order reliability method," *Communications in Nonlinear Science and Numerical Simulation*, vol. 15, pp. 3131-3141, 2010.
- [5] M. Papadrakakis and N. D. Lagaros, "Reliability-based structural optimization using neural networks and Monte Carlo simulation," *Computer Methods in Applied Mechanics and Engineering*, vol. 191, pp. 3491-3507, 2002.
- [6] C.-Y. Lin, W.-H. Huang, M.-C. Jeng, and J.-L. Doong, "Study of an assembly tolerance allocation model based on Monte Carlo simulation," *Journal of Materials Processing Technology*, vol. 70, pp. 9-16, 1997.
- [7] D. H. Jung and B. C. Lee, "Development of a simple and efficient method for robust optimization," *International Journal for Numerical Methods in Engineering*, vol. 53, pp. 2201-2215, 2002.
- [8] R. H. Myers, D. C. Montgomery, and C. M. Anderson-Cook, *Response surface methodology: process and product optimization using designed experiments* vol. 705: John Wiley & Sons, 2009.
- [9] S. Rahman and H. Xu, "A univariate dimension-reduction method for multi-dimensional integration in stochastic mechanics," *Probabilistic Engineering Mechanics*, vol. 19, pp. 393-408, 2004.
- [10] B. D. Youn, K. K. Choi, and K. Yi, "Performance moment integration (PMI) method for quality assessment in reliability-based robust design optimization," *Mechanics Based Design of Structures and Machines*, vol. 33, pp. 185-213, 2005.
- [11] A. Hasofer and N. C. Lind, "An exact and invariant first order reliability format," *ASCE Journal of Engineering Mechanics Division*, vol. 100, 1974.
- [12] R. Rackwitz and B. Flessler, "Structural reliability under combined random load sequences," *Computers & Structures*, vol. 9, pp. 489-494, 1978.
- [13] K. Breitung, "Asymptotic approximations for multinormal integrals," *Journal of Engineering Mechanics*, vol. 110, pp. 357-366, 1984.
- [14] L. Tvedt, "Distribution of quadratic forms in normal space-application to structural reliability," *Journal of Engineering Mechanics*, vol. 116, pp. 1183-1197, 1990.
- [15] S. Rahman and D. Wei, "A univariate approximation at most probable point for higher-order reliability analysis," *International journal of solids and structures*, vol. 43, pp. 2820-2839, 2006.
- [16] D. Wei and S. Rahman, "A multi-point univariate decomposition method for structural reliability analysis," *International Journal of Pressure Vessels and Piping*, vol. 87, pp. 220-229, 2010.



- [17] D. Yang, G. Li, and G. Cheng, "Convergence analysis of first order reliability method using chaos theory," *Computers & Structures*, vol. 84, pp. 563-571, 2006.
- [18] P.-L. Liu and A. Der Kiureghian, "Optimization algorithms for structural reliability," *Structural Safety*, vol. 9, pp. 161-177, 1991.
- [19] Y. Zhang and A. Der Kiureghian, "Two improved algorithms for reliability analysis," in *Reliability and optimization of structural systems*, ed: Springer, 1995, pp. 297-304.
- [20] T. Santosh, R. Saraf, A. Ghosh, and H. Kushwaha, "Optimum step length selection rule in modified HL-RF method for structural reliability," *International Journal of Pressure Vessels and Piping*, vol. 83, pp. 742-748, 2006.
- [21] S. Santos, L. Matioli, and A. Beck, "New optimization algorithms for structural reliability analysis," *Computer Modeling in Engineering & Sciences (CMES)*, vol. 83, pp. 23-55, 2012.
- [22] B. Fiessler, R. Rackwitz, and H.-J. Neumann, "Quadratic limit states in structural reliability," *Journal of the Engineering Mechanics Division*, vol. 105, pp. 661-676, 1979.
- [23] G. Q. Cai and I. Elishakoff, "Refined second-order reliability analysis," *Structural Safety*, vol. 14, pp. 267-276, 1994.
- [24] H. U. Köylüoğlu and S. R. Nielsen, "New approximations for SORM integrals," *Structural Safety*, vol. 13, pp. 235-246, 1994.
- [25] Y.-G. Zhao and T. Ono, "New approximations for SORM: Part 1," *Journal of Engineering Mechanics*, vol. 125, pp. 79-85, 1999.
- [26] Y.-G. Zhao and T. Ono, "New approximations for SORM: Part 2," *Journal of Engineering Mechanics*, vol. 125, pp. 86-93, 1999.
- [27] R. E. Melchers, "Structural reliability analysis and prediction," *Baffins Lane, Chichester, West Sussex, England: Wiley*, 1999.
- [28] M. Rashki, M. Miri, and M. A. Moghaddam, "A new efficient simulation method to approximate the probability of failure and most probable point," *Structural Safety*, vol. 39, pp. 22-29, 2012.
- [29] A. S. Nowak and K. R. Collins, *Reliability of structures*: CRC Press, 2012.
- [30] Y. Ibrahim, "Observations on applications of importance sampling in structural reliability analysis," *Structural Safety*, vol. 9, pp. 269-281, 1991.
- [31] R. Melchers, "Search-based importance sampling," *Structural Safety*, vol. 9, pp. 117-128, 1990.
- [32] O. Ditlevsen, R. E. Melchers, and H. Gluwer, "General multi-dimensional probability integration by directional simulation," *Computers & Structures*, vol. 36, pp. 355-368, 1990.
- [33] J. Nie and B. R. Ellingwood, "Directional methods for structural reliability analysis," *Structural Safety*, vol. 22, pp. 233-249, 2000.
- [34] D. Huntington and C. Lyrantzis, "Improvements to and limitations of Latin hypercube sampling," *Probabilistic Engineering Mechanics*, vol. 13, pp. 245-253, 1998.
- [35] A. Olsson, G. Sandberg, and O. Dahlblom, "On Latin hypercube sampling for structural reliability analysis," *Structural Safety*, vol. 25, pp. 47-68, 2003.
- [36] S.-K. Au and J. L. Beck, "Estimation of small failure probabilities in high dimensions by subset simulation," *Probabilistic Engineering Mechanics*, vol. 16, pp. 263-277, 2001.
- [37] S. Au and J. Beck, "Subset Simulation and its Application to Seismic Risk Based on Dynamic Analysis," *Journal of Engineering Mechanics*, vol. 129, pp. 901-917, 2003.
- [38] A. M. Freudenthal, J. M. Garretts, and M. Shinozuka, "The Analysis of Structural Safety," *Journal of the Structural Division*, vol. 92, pp. 267-326, 1966.
- [39] L. Davis., *Handbook of genetic algorithms*. New York: Van Nostrand Reinhold, 1991.

- [40] S. Das and P. N. Suganthan, "Differential Evolution: A Survey of the State-of-the-Art," *Evolutionary Computation, IEEE Transactions on*, vol. 15, pp. 4-31, 2011.
- [41] R. C. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," presented at the The sixth international symposium on micro machine and human science 1995.
- [42] R. C. Eberhart and S. Yuhui, "Particle swarm optimization: developments, applications and resources," in *Proceedings of the 2001 Congress on Evolutionary Computation* Seoul, 2001, pp. 81-86 vol. 1.
- [43] M. Dorigo, V. Maziezzo, and A. Coloni, "The ant system: optimization by a colony of cooperating ants," *Systems, Man and Cybernetics B. IEEE Trans.*, vol. 26, pp. 29-41, 1996.
- [44] Z. W. Geem, J. H. Kim, and G. V. Loganathan, "A New Heuristic Optimization Algorithm: Harmony Search," *SIMULATION*, vol. 76, pp. 60-68, February 1, 2001 2001.
- [45] S.-C. Chu, P.-w. Tsai, and J.-S. Pan, "Cat Swarm Optimization," in *PRICAI 2006: Trends in Artificial Intelligence*. vol. 4099, Q. Yang and G. Webb, Eds., ed: Springer Berlin Heidelberg, 2006, pp. 854-858.
- [46] X.-S. Yang and S. Deb, "Cuckoo Search via Lévy flights," in *Nature & Biologically Inspired Computing, 2009. NaBIC 2009. World Congress on*, 2009, pp. 210-214.
- [47] X.-S. Yang, "Firefly algorithm, stochastic test functions and design optimisation," *International Journal of Bio-Inspired Computation*, vol. 2, pp. 78-84, 2010.
- [48] X.-S. Yang, "A New Metaheuristic Bat-Inspired Algorithm," in *Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)*. vol. 284, J. González, D. Pelta, C. Cruz, G. Terrazas, and N. Krasnogor, Eds., ed: Springer Berlin Heidelberg, 2010, pp. 65-74.
- [49] X.-S. Yang and A. H. Gandomi, "Bat algorithm: a novel approach for global engineering optimization," *Engineering Computations*, vol. 29, pp. 464-483, 2012.
- [50] A. Gandomi, X.-S. Yang, A. Alavi, and S. Talatahari, "Bat algorithm for constrained optimization tasks," *Neural Computing and Applications*, vol. 22, pp. 1239-1255, 2013.
- [51] R. A. Jordehi, "Chaotic bat swarm optimisation (CBSO)," *Applied Soft Computing*, vol. 26, pp. 523-530, 2015.
- [52] S. Mirjalili, S. M. Mirjalili, and X.-S. Yang, "Binary bat algorithm," *Neural Computing and Applications*, vol. 25, pp. 663-681, 2013.
- [53] X.-s. He, W.-J. Ding, and X.-S. Yang, "Bat algorithm based on simulated annealing and Gaussian perturbations," *Neural Computing and Applications*, vol. 25, pp. 459-468, 2014.
- [54] J. Xie, Y. Zhou, and H. Chen, "A novel bat algorithm based on differential operator and Levy flights trajectory," *Comput Intell Neurosci*, vol. 2013, ID453812 2013.
- [55] T.-T. Nguyen, J.-S. Pan, T.-K. Dao, M.-Y. Kuo, and M.-F. Horng, "Hybrid Bat Algorithm with Artificial Bee Colony," in *Intelligent Data analysis and its Applications, Volume II*. vol. 298, J.-S. Pan, V. Snasel, E. S. Corchado, A. Abraham, and S.-L. Wang, Eds., ed: Springer International Publishing, 2014, pp. 45-55.
- [56] T.-S. Pan, T.-K. Dao, T.-T. Nguyen, and S.-C. Chu, "Hybrid Particle Swarm Optimization with Bat Algorithm," in *Genetic and Evolutionary Computing*. vol. 329, H. Sun, C.-Y. Yang, C.-W. Lin, J.-S. Pan, V. Snasel, and A. Abraham, Eds., ed: Springer International Publishing, 2015, pp. 37-47.
- [57] T.-K. Dao, J.-S. Pan, T.-T. Nguyen, S.-C. Chu, and C.-S. Shieh, "Compact Bat Algorithm," in *Intelligent Data analysis and its Applications, Volume II*. vol. 298, J.-S. Pan, V. Snasel, E. S. Corchado, A. Abraham, and S.-L. Wang, Eds., ed: Springer International Publishing, 2014, pp. 57-68.

- [58] X. Wang, W. Wang, and Y. Wang, "An Adaptive Bat Algorithm," in *Intelligent Computing Theories and Technology*. vol. 7996, D.-S. Huang, K.-H. Jo, Y.-Q. Zhou, and K. Han, Eds., ed: Springer Berlin Heidelberg, 2013, pp. 216-223.
- [59] W. Wang, Y. Wang, and X. Wang, "Bat Algorithm with Recollection," in *Intelligent Computing Theories and Technology*. vol. 7996, D.-S. Huang, K.-H. Jo, Y.-Q. Zhou, and K. Han, Eds., ed: Springer Berlin Heidelberg, 2013, pp. 207-215.
- [60] Y.-T. Chen, C.-S. Shieh, M.-F. Horng, B.-Y. Liao, J.-S. Pan, and M.-T. Tsai, "A Guidable Bat Algorithm Based on Doppler Effect to Improve Solving Efficiency for Optimization Problems," in *Computational Collective Intelligence. Technologies and Applications*. vol. 8733, D. Hwang, J. Jung, and N.-T. Nguyen, Eds., ed: Springer International Publishing, 2014, pp. 373-383.
- [61] I. J. Fister, D. Fister, and X.-S. Yang, "A hybrid bat algorithm," *ELEKTROTEHNIŠKI VEŠTNIK*, vol. 80, pp. 1-7, 2013.
- [62] I. J. Fister, S. Fong, J. Brest, and I. Fister, "A novel hybrid self-adaptive bat algorithm," *ScientificWorldJournal*, vol. 2014, pp. 709-738, 2014.
- [63] S. Yilmaz and E. U. Kūçūksille, "A new modification approach on bat algorithm for solving optimization problems," *Applied Soft Computing*, vol. 28, pp. 259-275, 2015.
- [64] M. S. Alam and M. W. U. Kabir, "Bat Algorithm with Self-adaptive Mutation: A Comparative Study on Numerical Optimization Problems," *International Journal of Computer Applications*, vol. 100, pp. 7-13, 2014.
- [65] S. Yilmaz, E. U. Kucuksille, and Y. Cengiz, "Modified Bat Algorithm," *Electronics and Electrical Engineering*, vol. 20, pp. 71-78, 2014.
- [66] P. W. Tsai, J. S. Pan, B. Y. Liao, M. J. Tsai, and V. Istanda, "Bat Algorithm Inspired Algorithm for Solving Numerical Optimization Problems," *Applied Mechanics and Materials*, vol. 148-149, pp. 134-137, 2011.
- [67] L. Li and Y. Zhou, "A novel complex-valued bat algorithm," *Neural Computing and Applications*, vol. 25, pp. 1369-1381, 2014.
- [68] O. K. Erol and I. Eksin, "A new optimization method: Big Bang–Big Crunch," *Advances in Engineering Software*, vol. 37, pp. 106-111, 2006.
- [69] E. Rashedi, H. Nezamabadi-pour, and S. Saryazdi, "GSA: A Gravitational Search Algorithm," *Information Sciences*, vol. 179, pp. 2232-2248, 2009.
- [70] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by Simulated Annealing," *Science*, vol. 220, pp. 671-680, 1983.
- [71] A. R. Mehrabian and C. Lucas, "A novel numerical optimization algorithm inspired from weed colonization," *Ecological Informatics*, vol. 1, pp. 355-366, 2006.
- [72] P. Suganthan, N. Hansen, J. liang, K. Deb, Y. Chen, A. Auger, and S. Tiwari, *Problem definitions and evaluation criteria for the CEC'2005 special session on real parameter optimization*: KanGAL Report, 2005005., 2005.
- [73] O. Hasançebi, T. Teke, and O. Pekcan, "A bat-inspired algorithm for structural optimization," *Computers & Structures*, vol. 128, pp. 77-90, 2013.
- [74] A. K. Qin and P. N. Suganthan, "Self-adaptive differential evolution algorithm for numerical optimization," in *The 2005 IEEE Congress on Evolutionary Computation*, Edinburgh, Scotland, 2005, pp. 1785-1791 Vol. 2.
- [75] J. Derrac, S. García, D. Molina, and F. Herrera, "A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms," *Swarm and Evolutionary Computation*, vol. 1, pp. 3-18, 2011.

- [76] S. García, D. Molina, M. Lozano, and F. Herrera, "A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: a case study on the CEC'2005 Special Session on Real Parameter Optimization," *J Heuristics*, vol. 15, pp. 617-644, 2009.
- [77] A. Auger and N. Hansen, "A Restart CMA Evolution Strategy With Increasing Population Size," presented at the The 2005 IEEE Congress on Evolutionary Computation, Edinburgh, Scotland, 2005.
- [78] L. J. Eshelman, "The CHC Adaptive Search Algorithm: How to Have Safe Search When Engaging," in *Foundations of Genetic Algorithms 1991*. vol. 1, G. J. E. Rawlins, Ed., ed San Mateo, California: Morgan Kaufmann, 1991, pp. 265-283.
- [79] L. J. Eshelman and J. D. Schaffer, "Real-Coded Genetic Algorithms and Interval-Schemata," in *Foundations of genetic algorithms*, W. D., Ed., ed San Mateo, California: Morgan Kaufmann, 1993, pp. 187-202.
- [80] C. Fernandes and A. Rosa, "A study of non-random matching and varying population size in genetic algorithm using a royal road function," presented at the The 2001 Congress on Evolutionary Computation, Piscataway, New Jersey, 2001.
- [81] H. Mühlenbein and D. Schlierkamp-Voosen, "Predictive Models for the Breeder Genetic Algorithm I. Continuous Parameter Optimization," *Evolutionary computation*, vol. 1, pp. 25-49, 1993.
- [82] M. Laguna and R. Marti, *Scatter Search: Methodology and Implementations in C*: Kluwer Academic Publishers, 2002.
- [83] F. Herrera, M. Lozano, and D. Molina, "Continuous scatter search: An analysis of the integration of some combination methods and improvement strategies," *European Journal of Operational Research*, vol. 169, pp. 450-476, 2006.
- [84] K. Price, R. Storn, and J. Lampinen, *Differential Evolution A Practical Approach to Global Optimization*: Springer-Verlag, 2005.
- [85] C. Elegbede, "Structural reliability assessment based on particles swarm optimization," *Structural Safety*, vol. 27, pp. 171-186, 2005.
- [86] J. Wang and M. Ghosn, "Linkage-shredding genetic algorithm for reliability assessment of structural systems," *Structural Safety*, vol. 27, pp. 49-72, 2005.
- [87] J. Cheng and Q. S. Li, "Reliability analysis of structures using artificial neural network based genetic algorithms," *Computer Methods in Applied Mechanics and Engineering*, vol. 197, pp. 3742-3750, 2008.
- [88] Y. Hao, G.-W. Meng, Z.-P. Zhou, and F. Li, "Structural Reliability Analysis Based on Imperialist Competitive Algorithm," in *Intelligent Systems Design and Engineering Applications, 2013 Fourth International Conference on*, 2013, pp. 570-574.
- [89] H. Zhao, Z. Ru, X. Chang, and S. Li, "Reliability Analysis Using Chaotic Particle Swarm Optimization," *Quality and Reliability Engineering International*, vol. 31, pp. n/a-n/a, 2014.
- [90] A. Kaveh, M. S. Massoudi, and M. G. Bagha, "Structural reliability analysis using charged system search algorithm," *IJST, Trans Civil Eng.*, vol. 38, pp. 439-448, 2014.
- [91] A. Kaveh, ; and M. I. Ghazaan, "Structural reliability assessment utilizing four metaheuristic algorithms," *Int. J. Optim. Civil Eng.*, vol. 5, pp. 189-209, 2015.
- [92] R. Mallipeddi and P. N. Suganthan, "Ensemble of Constraint Handling Techniques," *Evolutionary Computation, IEEE Transactions on*, vol. 14, pp. 561-579, 2010.
- [93] T. Takahama and S. Sakai, "Constrained optimization by the  $\epsilon$  constrained differential evolution with gradient-based mutation and feasible elites," in *Evolutionary Computation, 2006. CEC 2006. IEEE Congress on*, 2006, pp. 1-8.

- [94] K. Deb, "An efficient constraint handling method for genetic algorithms," *Computer Methods in Applied Mechanics and Engineering*, vol. 186, pp. 311-338, 2000.
- [95] B. Tessema and G. G. Yen, "A Self Adaptive Penalty Function Based Algorithm for Constrained Optimization," in *Evolutionary Computation, 2006. CEC 2006. IEEE Congress on*, 2006, pp. 246-253.
- [96] A. O. Kusakci and M. Can, "Constrained optimization with evolutionary algorithms: A comprehensive review," *Southeast Europe Journal of Soft Computing*, vol. 1, pp. 16-24, 2012.
- [97] Z. Wang, S. Li, and Z. Sang, "A new constraint handling method based on the modified Alopex-based evolutionary algorithm," *Computers & Industrial Engineering*, vol. 73, pp. 41-50, 2014.
- [98] Q. He and L. Wang, "A hybrid particle swarm optimization with a feasibility-based rule for constrained optimization," *Applied Mathematics and Computation*, vol. 186, pp. 1407-1422, 2007.
- [99] C. A. Coello Coello, "Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art," *Computer Methods in Applied Mechanics and Engineering*, vol. 191, pp. 1245-1287, 2002.
- [100] A. Homaifar, C. X. Qi, and S. H. Lai, "Constrained Optimization Via Genetic Algorithms," *SIMULATION*, vol. 62, pp. 242-253, 1994.
- [101] J. A. Joines and C. R. Houck, "On the use of non-stationary penalty functions to solve nonlinear constrained optimization problems with GA's," in *Evolutionary Computation, 1994. IEEE World Congress on Computational Intelligence., Proceedings of the First IEEE Conference on*, 1994, pp. 579-584 vol.2.
- [102] Z. Michalewicz and N. Attia, "Evolutionary optimization of constrained problems," in *Proceeding of the Third Conference on Evolutionary vProgramming*, Singapore, 1994, pp. 98-108.
- [103] A. B. Hadj-Alouane and J. C. Bean, "A Genetic Algorithm for the Multiple-Choice Integer Program," *Operations Research*, vol. 45, pp. 92-101, 1997.
- [104] A. E. Smith and D. M. Tate, "Genetic Optimization Using A Penalty Function," presented at the Proceedings of the 5th International Conference on Genetic Algorithms, 1993.
- [105] A. E. Eiben and J. K. van der Hauw, "Adaptive penalties for evolutionary graph coloring," in *Artificial Evolution*. vol. 1363, J.-K. Hao, E. Lutten, E. Ronald, M. Schoenauer, and D. Snyers, Eds., ed: Springer Berlin Heidelberg, 1998, pp. 95-106.
- [106] Z. Michalewicz and M. Schoenauer, "Evolutionary Algorithms for Constrained Parameter Optimization Problems," *Evolutionary computation*, vol. 4, pp. 1-32, 1996.
- [107] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *Evolutionary Computation, IEEE Transactions on*, vol. 1, pp. 67-82, 1997.
- [108] E. Valian, S. Tavakoli, S. Mohanna, and A. Haghi, "Improved cuckoo search for reliability optimization problems," *Computers & Industrial Engineering*, vol. 64, pp. 459-468, 2013.
- [109] E. Valian, "Solving Reliability Optimization Problems by Cuckoo Search," in *Cuckoo Search and Firefly Algorithm*. vol. 516, X.-S. Yang, Ed., ed: Springer International Publishing, 2014, pp. 195-215.
- [110] D. Zou, L. Gao, J. Wu, S. Li, and Y. Li, "A novel global harmony search algorithm for reliability problems," *Computers & Industrial Engineering*, vol. 58, pp. 307-316, 2010.
- [111] V. Weingarten and P. Seide, "Buckling of thin-walled truncated cones," *NASA Space Vehicle Criteria (Structures), NASA SP-8019, Washington DC*, 1968.

- [112] F. Grooteman, "Adaptive radial-based importance sampling method for structural reliability," *Structural Safety*, vol. 30, pp. 533-542, 2008.
- [113] S. Au and J. L. Beck, "A new adaptive importance sampling scheme for reliability calculations," *Structural Safety*, vol. 21, pp. 135-158, 1999.
- [114] N. Gayton, J. Bourinet, and M. Lemaire, "CQ2RS: a new statistical approach to the response surface method for reliability analysis," *Structural Safety*, vol. 25, pp. 99-121, 2003.
- [115] N. Devictor, "Fiabilité et mécanique: méthodes FORM/SORM et couplages avec des codes d'éléments finis par des surfaces de réponse adaptatives," Phd Thesis, 1996.
- [116] X. S. Nguyen, A. Sellier, F. Duprat, and G. Pons, "Adaptive response surface method based on a double weighted regression technique," *Probabilistic Engineering Mechanics*, vol. 24, pp. 135-143, 2009.
- [117] R. Rackwitz, "Reliability analysis—a review and some perspectives," *Structural Safety*, vol. 23, pp. 365-395, 2001.
- [118] A. Mohsine and A. El Hami, "A robust study of reliability-based optimization methods under eigen-frequency," *Computer Methods in Applied Mechanics and Engineering*, vol. 199, pp. 1006-1018, 2010.
- [119] M. A. Ba-Abbad, E. Nikolaidis, and R. K. Kapania, "New Approach for System Reliability-Based Design Optimization," *AIAA Journal*, vol. 44, pp. 1087-1096, 2006.
- [120] S. E. Gano, J. E. Renaud, H. Agarwal, and A. Tovar, "Reliability-based design using variable-fidelity optimization," *Structure and Infrastructure Engineering*, vol. 2, pp. 247-260, 2006.
- [121] E. Nikolaidis and R. Burdisso, "Reliability based optimization: A safety index approach," *Computers & Structures*, vol. 28, pp. 781-788, 1988.
- [122] J. Tu, K. K. Choi, and Y. H. Park, "A New Study on Reliability-Based Design Optimization," *Journal of Mechanical Design*, vol. 121, pp. 557-564, 1999.
- [123] X. Du and W. Chen, "Sequential Optimization and Reliability Assessment Method for Efficient Probabilistic Design," *Journal of Mechanical Design*, vol. 126, pp. 225-233, 2004.
- [124] J. Liang, Z. P. Mourelatos, and E. Nikolaidis, "A Single-Loop Approach for System Reliability-Based Design Optimization," *Journal of Mechanical Design*, vol. 129, pp. 1215-1224, 2007.
- [125] J. Liang, Z. P. Mourelatos, and J. Tu, "A single-loop method for reliability-based design optimisation," *International Journal of Product Development*, vol. 5, pp. 76-92, 2008.
- [126] S. Shan and G. G. Wang, "Reliable design space and complete single-loop reliability-based design optimization," *Reliability Engineering & System Safety*, vol. 93, pp. 1218-1230, 2008.
- [127] K. S. Lee and Z. W. Geem, "A new structural optimization method based on the harmony search algorithm," *Computers & Structures*, vol. 82, pp. 781-798, 2004.
- [128] H. Adeli and N. Cheng, "Integrated Genetic Algorithm for Optimization of Space Structures," *Journal of Aerospace Engineering*, vol. 6, pp. 315-328, 1993.
- [129] M. Rosenblatt, "Remarks on a multivariate transformation," *The annals of mathematical statistics*, pp. 470-472, 1952.
- [130] Y.-T. Wu and W. Wang, "Efficient probabilistic design by converting reliability constraints to approximately equivalent deterministic constraints," *Journal of Integrated Design and Process Science, SDPS Trans*, vol. 2, pp. 13-21, 1998.
- [131] X. Qu and R. T. Haftka, "Reliability-based design optimization using probabilistic sufficiency factor," *Structural and Multidisciplinary Optimization*, vol. 27, pp. 314-325, 2004.

- [132] G. Cheng, L. Xu, and L. Jiang, "A sequential approximate programming strategy for reliability-based structural optimization," *Computers & Structures*, vol. 84, pp. 1353-1367, 2006.
- [133] H. O. Madsen and P. F. Hansen, "A Comparison of Some Algorithms for Reliability Based Structural Optimization and Sensitivity Analysis," in *Reliability and Optimization of Structural Systems '91*, vol. 76, R. Rackwitz and P. Thoft-Christensen, Eds., ed: Springer Berlin Heidelberg, 1992, pp. 443-451.
- [134] Y. Aoues and A. Chateauneuf, "Benchmark study of numerical methods for reliability-based design optimization," *Structural and Multidisciplinary Optimization*, vol. 41, pp. 277-294, 2010.
- [135] X. Chen, T. Hasselman, and D. Neill, "Reliability based structural design optimization for practical applications," in *38th Structures, Structural Dynamics, and Materials Conference*, ed: American Institute of Aeronautics and Astronautics, 1997.
- [136] C. A. Conceição António, "A hierarchical genetic algorithm for reliability based design of geometrically non-linear composite structures," *Composite Structures*, vol. 54, pp. 37-47, 2001.
- [137] B. A. Tolson, H. R. Maier, A. R. Simpson, and B. J. Lence, "Genetic Algorithms for Reliability-Based Optimization of Water Distribution Systems," *Journal of Water Resources Planning and Management*, vol. 130, pp. 63-72, 2004.
- [138] I. T. Yang and Y.-H. Hsieh, "Reliability-based design optimization with discrete design variables and non-smooth performance functions: AB-PSO algorithm," *Automation in Construction*, vol. 20, pp. 610-619, 2011.
- [139] J. Chen, Y. Tang, R. Ge, Q. An, and X. Guo, "Reliability design optimization of composite structures based on PSO together with FEA," *Chinese Journal of Aeronautics*, vol. 26, pp. 343-349, 2013.
- [140] I. T. Yang and Y.-H. Hsieh, "Reliability-based design optimization with cooperation between support vector machine and particle swarm optimization," *Engineering with Computers*, vol. 29, pp. 151-163, 2013.
- [141] C. K. Dimou and V. K. Koumoussis, "Reliability-Based Optimal Design of Truss Structures Using Particle Swarm Optimization," *Journal of Computing in Civil Engineering*, vol. 23, pp. 100-109, 2009.
- [142] K. Deb, S. Gupta, D. Daum, J. Branke, A. K. Mall, and D. Padmanabhan, "Reliability-Based Optimization Using Evolutionary Algorithms," *Evolutionary Computation, IEEE Transactions on*, vol. 13, pp. 1054-1074, 2009.
- [143] S. Mathakari, P. Gardoni, P. Agarwal, A. Raich, and T. Haukaas, "Reliability-Based Optimal Design of Electrical Transmission Towers Using Multi-Objective Genetic Algorithms," *Computer-Aided Civil and Infrastructure Engineering*, vol. 22, pp. 282-292, 2007.
- [144] Z. Li, G. Tian, G. Cheng, H. Liu, and Z. Cheng, "An integrated cultural particle swarm algorithm for multi-objective reliability-based design optimization," *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, vol. 228, pp. 1185-1196, 2013.
- [145] S. Kawaji and N. Kogiso, "Convergence improvement of reliability-based multiobjective optimization using hybrid MOPSO," presented at the 10<sup>th</sup> World Congress on Structural and Multidisciplinary Optimization, Orlando, Florida, USA, 2013.
- [146] K.-W. Liao and G. Ivan, "A single loop reliability-based design optimization using EPM and MPP-based PSO," *Latin American Journal of Solids and Structures*, vol. 11, pp. 826-847, 2014.

- [147] X. Du and W. Chen, "A Most Probable Point-Based Method for Efficient Uncertainty Analysis," *Journal of Design and Manufacturing Automation*, vol. 4, pp. 47-66, 2001.
- [148] B. D. Youn, K. K. Choi, and Y. H. Park, "Hybrid Analysis Method for Reliability-Based Design Optimization," *Journal of Mechanical Design*, vol. 125, pp. 221-232, 2003.
- [149] L. Gu, R. J. Yang, C. H. Tho, M. Makowskit, O. Faruquet, and Y. L. Y.Li, "Optimisation and robustness for crashworthiness of side impact," *International Journal of Vehicle Design*, vol. 26, pp. 348-360, 2001.
- [150] T. Zou and S. Mahadevan, "A direct decoupling approach for efficient reliability-based design optimization," *Structural and Multidisciplinary Optimization*, vol. 31, pp. 190-200, 2006.
- [151] R. C. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Proceedings of the sixth international symposium on micro machine and human science*, 1995, pp. 39-43.
- [152] F. Li, T. Wu, M. Hu, and J. Dong, "An accurate penalty-based approach for reliability-based design optimization," *Research in Engineering Design*, vol. 21, pp. 87-98, 2010.
- [153] J. Golinski, "Optimal synthesis problems solved by means of nonlinear programming and random methods," *Journal of Mechanisms*, vol. 5, pp. 287-309, 1970.
- [154] J. J. Lee and B. C. Lee, "Efficient evaluation of probabilistic constraints using an envelope function," *Engineering Optimization*, vol. 37, pp. 185-200, 2005.
- [155] B. Hyeon Ju and B. Chai Lee, "Reliability-based design optimization using a moment method and a kriging metamodel," *Engineering Optimization*, vol. 40, pp. 421-438, 2008.
- [156] T. M. Cho and B. C. Lee, "Reliability-based design optimization using convex linearization and sequential optimization and reliability assessment method," *Structural Safety*, vol. 33, pp. 42-50, 2011.
- [157] X. Yin and W. Chen, "Enhanced sequential optimization and reliability assessment method for probabilistic optimization with varying design variance," *Structure and Infrastructure Engineering*, vol. 2, pp. 261-275, 2006.
- [158] M. McDonald and S. Mahadevan, "Reliability-Based Optimization With Discrete and Continuous Decision and Random Variables," *Journal of Mechanical Design*, vol. 130, ID: 061401, 2008.
- [159] JCSS, "Probabilistic model code," ed. Denmark: Joint Committee on Structural Safety, 2000.
- [160] B. 2573:1, "BS 2573: Part 1, Rules for design of cranes, specification for classification, stress calculation and design criteria for structures," vol. BS 2573:1, ed. London: British Standards Institution, 1983.
- [161] A. o. I. a. S. Engineers, "Specification for electric overhead traveling cranes for use in steel mill service," vol. AISE Standard N°6, ed. Pittsburgh: Association of Iron and Steel Engineers, 1969.
- [162] I. S. Institution, "Code of practice for design, manufacture, erection and testing (structural portion) of cranes and hoists," vol. I.S.807, ed. India: Indian Standards Institution, 1963.
- [163] CMAA, "Specifications for electric overhead traveling cranes," vol. CMAA N°70, ed. USA: Crane Manufacturers Association of America, 1975.
- [164] D. 15018, "Cranes; Principles for steel structures; Stress Analysis," vol. DIN 15018, ed. Germany: Deutsche Norm, 1974.
- [165] B-8821, "Specification for the design of crane structure," vol. Japanese Industrial standard B-8821, ed. Tokyo, Japan: Japanese Standards Association, 1976.
- [166] S. W. Cho and B. M. Kwak, "Optimal Design of Electric Overhead Crane Girders," *Journal of Mechanisms, Transmissions, and Automation in Design*, vol. 106, pp. 203-208, 1984.



- [167] B. 5400:3, "BS 5400: Part 3, Code of practice for design of steel bridges," vol. BS 5400:3, ed. London: British Standards Institution, 1982.
- [168] H. G. Greiner, *Crane Handbook, 3rd Edition*. Harvey, Illinois: Whiting Corporation, 1967.
- [169] S. S. Rao, "Optimum Design of Bridge Girders for Electric Overhead Traveling Cranes," *Journal of Engineering for Industry*, vol. 100, pp. 375-382, 1978.
- [170] B. 3579, "Specifications for heavy duty electric overhead traveling and special cranes for use in steel works," vol. BS 3579, ed. London: British Standards Institution, 1963.
- [171] J. Farkas, "Economy of higher-strength steels in overhead travelling cranes with double-box girders," *Journal of Constructional Steel Research*, vol. 6, pp. 285-301, 1986.
- [172] K. Jarmai, "Decision support system on IBM PC for design of economic steel structures applied to crane girders," *Thin-Walled Structures*, vol. 10, pp. 143-159, 1990.
- [173] M. Abid, M. Akmal, and S. Parvez, "Optimization of Box Type Girder of Overhead Crane," in *Global Design to Gain a Competitive Edge*, X.-T. Yan, W. Ion, and B. Eynard, Eds., ed: Springer London, 2008, pp. 609-618.
- [174] B. Pinca, O. Tirian, V. Socalici, and D. Ardeleadn, "Dimensional optimization for the strength structure of a traveling crane," *WSEAS Transactions on Applied and theoretical Mechanics*, vol. 4, pp. 147-156, 2009.
- [175] R. H. Zuberi, L. Kai, and Z. Zhengxing, "Design optimization of EOT crane bridge," presented at the International Conference on Engineering Optimization - EngOpt 2008, Rio de Janeiro, Brazil, 2008.
- [176] G. F. Tian, S. Z. Zhang, and S. H. Sun, "The Optimization Design of Overhead Traveling Crane's Box Girder," in *Advanced Materials Research*, 2013, pp. 3-8.
- [177] G. Pavlovic, M. Savkovic, M. Gasic, R. Bulatovic, and N. Zdravkovic, "Optimization of the box section of the main girder of the double beam bridge crane according to the criteria of lateral stability and local stability of plates," *Machine Design*, vol. 4, pp. 197-204, 2012.
- [178] M. Savković, M. Gašić, D. Čatić, R. Nikolić, and G. Pavlović, "Optimization of the box section of the main girder of the bridge crane with the rail placed above the web plate," *Structural and Multidisciplinary Optimization*, vol. 47, pp. 273-288, 2013.
- [179] Q. X. Wang, X. F. Zhang, and C. B. Sun, "Lightweight Design and Research of Box Girder for Double Girder Bridge Crane," in *Applied Mechanics and Materials*, 2012, pp. 478-481.
- [180] T. Yifei, Y. Wei, Y. Zhen, L. Dongbo, and L. Xiangdong, "Research on multidisciplinary optimization design of bridge crane," *Mathematical Problems in Engineering*, vol. 2013, ID: 763545, 10 pages, 2013.
- [181] N. Lagaros and M. Papadrakakis, "Applied soft computing for optimum design of structures," *Structural and Multidisciplinary Optimization*, vol. 45, pp. 787-799, 2012.
- [182] X. Qu, G. Xu, X. Fan, and X. Bi, "Intelligent optimization methods for the design of an overhead travelling crane," *Chinese Journal of Mechanical Engineering*, vol. 28, pp. 187-196, 2015.
- [183] C. S. Press, "Design rules for cranes," vol. GB/T3811-2008, ed. Beijing: China Standards Press, 2008.
- [184] X. Fan and X. Bi, "Reliability-based design optimization for crane metallic structure using ACO and AFOSM based on China standards," *Mathematical Problems in Engineering*, vol. 2015, pp. Article ID 828930, 12 pages, 2015.
- [185] Y. T. Wu, H. R. Millwater, and T. A. Cruse, "Advanced probabilistic structural analysis method for implicit performance functions," *AIAA Journal*, vol. 28, pp. 1663-1669, 1990.
- [186] T. Niezgodziński and T. Kubiak, "The problem of stability of web sheets in box-girders of overhead cranes," *Thin-Walled Structures*, vol. 43, pp. 1913-1925, 2005.

- [187] A. Blum and K. Haremski, "An influence of technological contraflexure of crane bridge on its durability," *Transport Problem*, vol. 5, pp. 5-14, 2010.
- [188] A. Chakri, R. Khelif, and M. Benouaret, "Conception fiabiliste d'un système mécanique: Cas d'un pont roulant," presented at the Advanced Materials: Processing, Properties, Reliability and Diagnostics, ACMA 2012 Fez, Morocco, 2012.
- [189] X. Mei, D. Dong, and Y. Teng, "Bi-probability fatigue life prediction for bridge crane structures," *Advanced Materials Research*, vol. 485-484, pp. 736-740, 2012.
- [190] K. Terashima and M. Suzuki, "Path Planning and Navigation of Overhead Traveling Crane with Three-Dimensional Transport Based on a Diffusion Equation Strategy," in *Advances in Manufacturing*, ed: Springer, 1999, pp. 335-347.
- [191] V. Mandelli and A. Haider, "Converting existing overhead cranes to a fully automatic operation," in *Cement Industry Technical Conference, 1999. Conference Record. 1999 IEEE-IAS/PCA*, 1999, pp. 357-375.
- [192] T. Akamatsu, A. Kaneshige, and K. Terashima, "Real time path planning based on the potential method for an autonomous mobile overhead traveling crane," in *Industrial Electronics, 2004 IEEE International Symposium on*, 2004, pp. 699-704.
- [193] F. Omar, F. Karray, O. Basir, and L. Yu, "Autonomous overhead crane system using a fuzzy logic controller," *Journal of Vibration and Control*, vol. 10, pp. 1255-1270, 2004.
- [194] T. Wecker, H. Aschemann, and E. Hofer, "Sensor-based collision avoidance for rope-suspended autonomous material flow systems," in *World Congress*, 2005, pp. 2030-2030.
- [195] T. Miyoshi, S. Kawakami, and K. Terashima, "Path planning and obstacle avoidance considering rotary motion of load for overhead cranes," *Journal of Mechanical Systems for Transportation and Logistics*, vol. 1, pp. 134-145, 2008.
- [196] A. Kaneshige, T. Miyoshi, and K. Terashima, "The development of an autonomous mobile overhead crane system for the liquid tank transfer," in *Advanced Intelligent Mechatronics, 2009. AIM 2009. IEEE/ASME International Conference on*, 2009, pp. 630-635.
- [197] J. Smoczek, J. Szpytko, and P. Hyla, "Non-collision path planning of a payload in crane operating space," in *Solid State Phenomena*, 2013, pp. 559-564.
- [198] J. Yang, W. Chien, M. Huang, and M. Tsai, "Application of Machine Vision to Collision Avoidance Control of the Overhead Crane," presented at the International Conference on Electrical, Automation and Mechanical Engineering Phuket, Thailand, 2015.
- [199] Y. Hu and S. X. Yang, "A knowledge based genetic algorithm for path planning of a mobile robot," in *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, 2004, pp. 4350-4355.
- [200] L. Wang, Y. Liu, H. Deng, and Y. Xu, "Obstacle-avoidance path planning for soccer robots using particle swarm optimization," in *Robotics and Biomimetics, 2006. ROBIO'06. IEEE International Conference on*, 2006, pp. 1233-1238.
- [201] L. Lu and D. Gong, "Robot path planning in unknown environments using particle swarm optimization," in *Natural Computation, 2008. ICNC'08. Fourth International Conference on*, 2008, pp. 422-426.
- [202] P. K. Mohanty and D. R. Parhi, "A new efficient optimal path planner for mobile robot based on Invasive Weed Optimization algorithm," *Frontiers of Mechanical Engineering*, vol. 9, pp. 317-330, 2014.
- [203] P. Sivakumar, K. Varghese, and N. R. Babu, "Automated path planning of cooperative crane lifts using heuristic search," *Journal of Computing in Civil Engineering*, vol. 17, pp. 197-207, 2003.

- [204] M. A. D. Ali, N. R. Babu, and K. Varghese, "Collision free path planning of cooperative crane manipulators using genetic algorithm," *Journal of Computing in Civil Engineering*, vol. 19, pp. 182-193, 2005.
- [205] X. Wang, Y. Zhang, D. Wu, and S. D. Gao, "Collision-free path planning for mobile cranes based on ant colony algorithm," *Key Engineering Materials*, vol. 467-469, pp. 1108-1115, 2011.