



République Algérienne Démocratique et Populaire  
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique  
Université A/Mira de Bejaia  
Faculté des Sciences et Sciences de l'Ingénieur  
Département Informatique  
Ecole Doctorale **Réseaux et Systèmes Distribués ReSyD**

---

*Mémoire de Magistère*

en Informatique

**Option**

Réseaux et Systèmes Distribués

**Thème**

---

**ETUDE COMPARATIVE DES ALGORITHMES DE VOTE  
DANS LES SYSTEMES TOLERANTS AUX FAUTES**

---

**Présenté par**  
Nadia BATTAT

**Devant le Jury composé de :**

Président :	Pr. D. AISSANI	Université Abderrahmane Mira de Béjaïa, Algérie
Rapporteur :	Pr. A. BELMAHDI	Université Abderrahmane Mira de Béjaïa, Algérie
Examineur :	Pr. N. BADACHE	Université de Bab Ezzouar (USTHB), Algérie
Examineur :	Pr. M.S. RADJEF	Université Abderrahmane Mira de Béjaïa, Algérie

**Promotion 2005-2006**

## Dédicaces

*A mes parents  
A toute ma famille  
A la mémoire de mon neveu JASSEM*

## Remerciements

Je tiens à remercier avant tout mon encadreur Ali Belmahdi, professeur à l'*université Abderrahmane Mira de Bejaia*, pour m'avoir proposé ce sujet très intéressant, pour ses conseils et orientations.

Je remercie les membres de Jury qui ont accepté de juger mon travail : Pr. D. AISSANI- Université Abderrahmane Mira de Béjaïa, Pr. N. BADACHE- Université de Bab Ezzouar (USTHB), Pr. M.S. RADJEF Université Abderrahmane Mira de Béjaïa.

Je remercie tous ceux qui ont contribué à la réussite de l'école doctorale ReSyD, en particulier Mr Tari, chef de département informatique à Bejaia, pour les efforts qu'il a fournis pour que notre formation puisse se dérouler.

Je remercie mes amis pour la patience et le dévouement dont ils ont fait preuve. Je remercie aussi mes collègues pour leurs caractères et ses aides.

Je remercie tout particulièrement ma famille, pour leur soutien inconditionnel tout au long de ces longues années d'études.

Enfin, je remercie tous ceux qui m'ont aidé ou avaient l'intention de le faire.

## Résumé

Ce mémoire porte sur une étude comparative d'algorithmes de vote ou d'élection qui sont employés pour arbitrer entre les résultats de modules redondants dans les systèmes tolérants aux fautes. Le voter de majorité inexacte, le voter de médiane et le voter de moyenne pondérée ont été employés dans beaucoup d'applications pour masquer les erreurs aux utilisateurs. Cependant, le premier nécessite la spécification du seuil de voter et les autres sont incapables de produire une sortie bénigne dans le cas de désaccord complet entre ses entrées. De plus, aucun de ces trois voters n'est capable de prendre en considération les incertitudes associées aux entrées. Un nouveau schéma (voter flou) basé sur la théorie des ensembles flous est utilisé pour résoudre les problèmes posés par les autres voters. Le voter flou est expérimentalement évalué du point de vue de sûreté, disponibilité et fiabilité, et comparé avec les voters de majorité inexacte, de médiane et de moyenne pondérée. Les résultats de simulation suggèrent que le voter flou soit approprié pour des applications dans lesquelles une petite dégradation de sa sûreté est acceptable. Dans ce mémoire, nous avons proposé une amélioration du voter flou par l'utilisation des algorithmes génétiques hiérarchisés. Ces algorithmes permettent d'optimiser les fonctions d'appartenance floues et les règles de décision intervenant dans la conception d'un voter flou. Les résultats de simulation montrent que le voter flou optimisé peut avoir une meilleure sûreté, une meilleure disponibilité et une meilleure fiabilité que les autres voters en présence de petites et de grandes erreurs.

**Mots clés :** tolérance aux fautes, masquage d'erreurs, seuil de voter, algorithmes génétiques hiérarchisés,

## Abstract

This thesis deals with a comparative study of voting algorithms which are used to arbitrate between the results of redundant modules in fault-tolerant systems. Inexact majority, median and weighted average voters have been used in many applications to mask the errors to the users. Although the first require an application-specific 'voter threshold' value to be specified, whereas median and weighted average voters are unable to produce a benign output when no agreement exists between the voter inputs. Neither voter type is able to cope with uncertainties associated with the voter inputs. A novel voting scheme (fuzzy voter) based on fuzzy set theory is used to solve the problems arising from the others voters. The fuzzy voter is experimentally evaluated from the point of view safety, availability and reliability, and compared with the inexact majority voter, median voter and weighted average voter. The simulation results suggest that the fuzzy voter is adapted for the applications in which a small degradation of its safeties is acceptable at the cost of a great increase in its availabilities. In this thesis, we proposed an amelioration of fuzzy voter by the hierarchical genetic algorithms. These algorithms use to optimize the fuzzy membership functions and to minimize the rules of decision intervening in the design of fuzzy voter. The simulation results show that the optimized fuzzy voter can have higher safety, higher availability and higher reliability than the others voters in the presence of small and large errors.

**Keywords:** Fault tolerance, mask the errors, voter threshold, hierarchical genetic algorithms

---

# TABLE DES MATIERES

<b>TABLE DES MATIERES</b>	<b>i</b>
<b>LISTE DES FIGURES</b>	<b>v</b>
<b>LISTE DES TABLEAUX</b>	<b>vi</b>
<b>LISTE DES ALGORITHMES</b>	<b>vii</b>
<b>INTRODUCTION GENERALE</b>	<b>1</b>
<b>1. INTRODUCTION A LA TOLERANCE AUX FAUTES</b>	<b>4</b>
1.1 Introduction.....	4
1.2 Définitions.....	4
1.2.1 Faute.....	4
1.2.2 Erreur.....	5
1.2.3 Défaillance.....	5
1.3 Classification des fautes et des défaillances.....	5
1.3.1 Classification des fautes.....	5
1.3.2 Classification des défaillances.....	7
1.4 Prévention de fautes.....	8
1.4.1 Évitement de fautes.....	8
1.4.2 Élimination de fautes.....	8
1.5 La Tolérance aux fautes.....	9
1.5.1 Niveaux de tolérance aux fautes.....	9
1.5.2 Traitement des fautes et des erreurs.....	9
1.5.2.1 Traitement des fautes.....	10
1.5.2.2 Traitement des erreurs.....	10
1.5.3 Tolérance aux fautes de logiciel.....	11
1.5.3.1 Techniques de tolérance aux fautes de logiciel a une seule version.....	11
1.5.3.2 Techniques de tolérance aux fautes de logiciel multi versions.....	15
1.5.4 Tolérance aux fautes de matériel.....	18
1.5.4.1 Techniques de redondance statique (ou passive).....	19
1.5.4.2 Techniques de redondance dynamique (ou active).....	20
1.5.4.2 Techniques de redondance hybride.....	20
1.5.5 Algorithmes de sélection de sortie.....	21
1.6 Conclusion.....	22
<b>2. LES ALGORITHMES DE VOTE DANS LES SYSTEMES TOLERANTS AUX FAUTES</b>	<b>23</b>
2.1 Introduction.....	23
2.2 Fonctionnalité générale d'un voter.....	24
2.3 Classification des algorithmes de vote.....	25
2.3.1 Le nombre de voters.....	25

2.3.2	La méthode d'implémentation.....	25
2.3.3	Le type d'agreement.....	26
2.3.4	La cardinalité d'espace de résultat.....	26
2.3.5	La nature de l'environnement de travail.....	26
2.3.6	La fonctionnalité du voter.....	27
2.4	Classification des voters basés sur leur fonctionnalité.....	27
2.4.1	Algorithmes de vote génériques.....	28
2.4.1.1	Résultat de sélection.....	28
2.4.1.2	Voter d'amalgamation de résultats.....	31
2.4.2	Algorithmes de vote hybrides.....	33
2.4.2.1	Algorithmes de vote supplémentaire probabiliste et information heuristique sur des variantes.....	34
2.4.2.2	Algorithmes de vote incorporant la prévision et le smoothing.....	34
2.4.2.3	Algorithmes de vote supplémentaire avec l'information diagnostiquée sur des variantes.....	36
2.4.3	Voters basés sur des mesures.....	36
2.4.3.1	Le voter de négociation en étapes.....	36
2.4.3.2	Le voter expédient.....	36
2.4.3.3	Le voter de 2 :3-manières.....	37
2.5	Conclusion .....	37
<b>2.</b>	<b>ETUDE COMPARATIVE D'ALGORITHMES DE VOTE</b>	<b>38</b>
3.1	Introduction.....	38
3.2	Comparaison des performances des algorithmes de vote.....	39
3.2.1	Structure de test de Harnais.....	39
3.2.2	Hypothèses.....	40
3.2.3	Paramètres d'environnement.....	41
3.2.4	Mesure des performances.....	41
3.3	Résultats de comparaison.....	42
3.3.1	Perturbation d'une seule entrée (erreur unique).....	42
3.3.1.1	Expérience n° 1.....	43
3.3.2	Perturbation de plusieurs entrées (erreurs multiples).....	43
3.3.2.1	Expérience n° 1.....	43
3.3.2.2	Expérience n° 2.....	51
3.3.2.3	Expérience n° 3.....	55
3.3.2.4	Expérience n° 4.....	59
3.3.2.5	Expérience n° 5.....	61
3.4	Conclusion.....	62
<b>3.</b>	<b>AMELIORATION DE L'APPROCHE BASEE SUR LA LOGIQUE FLOUE</b>	<b>63</b>
4.1	Introduction.....	63
4.2	Le voter flou.....	64
4.2.1	Fuzzification.....	64
4.2.2	Définition de l'agréabilité floue de chaque entrée.....	65
4.2.3	Définition de l'ensemble des règles de décision.....	65
4.2.4	Défuzzification de l'agréabilité floue.....	66
4.2.5	Calcul de la sortie pondérée du voter.....	67

4.3 Amélioration du voter flou.....	68
4.3.1 Algorithmes génétiques conventionnels.....	68
4.3.1.1 Terminologie.....	69
4.3.1.1 Principe de fonctionnement des algorithmes génétiques conventionnels...	70
4.3.2 Algorithmes génétiques hiérarchisés.....	73
4.3.2.1 Formulation et codage du chromosome hiérarchie.....	74
4.3.2.2 Activation des gènes de paramètres.....	74
4.3.2.3 Opérateurs génétiques.....	75
4.3.3 Optimisation du voter flou par les algorithmes génétiques hiérarchisés.....	75
4.3.3.1 Structure des chromosomes.....	76
4.3.3.2 Initialisation des chromosomes.....	77
4.3.3.3 Procédure réparatrice.....	78
4.3.3.4 Fonction d'adaptation.....	78
4.3.3.5 Réglage des paramètres de l'algorithme génétique hiérarchisé.....	79
4.3.3.6 Le cycle génétique de l'algorithme génétique hiérarchisé.....	81
4.4 Résultats de Simulation.....	83
4.4.1 Expérience n° 1.....	83
4.4.2 Expérience n° 2.....	85
4.4.2.1 Amélioration de la sûreté.....	86
4.4.2.1 Amélioration de la disponibilité et la fiabilité.....	87
4.5 Conclusion.....	89
<b>CONCLUSION GENERALE ET PERSPECTIVES</b>	<b>90</b>
<b>BIBLIOGRAPHIE</b>	<b>92</b>