

UNIVERSITE SAAD DAHLED DE BLIDA

Faculté des Sciences de l'Ingénieur

Département d'Electronique

MEMOIRE DE MAGISTER

Spécialité : Contrôle

UTILISATION D'UN CONTROLEUR HYBRIDE

FLOU - H.G.A. POUR LE CONTRÔLE D'UN REACTEUR

CHIMIQUE DU TYPE C.S.T.R.

Par

Abdelhalim BENOURED

Devant le jury composé de :

A. GUESSOUM	Professeur, U. de Blida	Président
D. BOUKHETALA	Maître de conférence, ENP	Examineur
M. HADJ SADOK	Chargé de cours, U. de Blida	Examineur
H. SALHI	Maître de conférence, U. de Blida	Rapporteur

Blida, Juillet 2007

RESUME

Dans ce travail nous expliquerons comment les algorithmes génétiques standards (A.G.S) et hiérarchiques (A.G.H.) peuvent effectuer une recherche automatique de la base de connaissance d'un contrôleur flou. Pour mettre en œuvre cette technique, on l'appliquera à un système dynamique non linéaire qui consiste à stabiliser la température d'un réacteur chimique du type C.S.T.R. Dans ce présent mémoire nous les utiliserons pour optimiser les règles floues puis l'optimisation simultanée des fonctions d'appartenance et des règles floues. De même nous montrerons que les algorithmes génétiques constituent un bon alternatif pour fournir la meilleure base de connaissance possible, et cela sans avoir besoin d'un expert dans le domaine ou, lorsqu'on n'a pas suffisamment de données du système à contrôler.

ملخص

في هذا العمل سوف نبين كيفية استعمال الخوارزميات الجينية والنموذجية و التدرجية من أجل البحث الآلي على القاعدة المعرفية للمراقب الغامض. و من أجل تحقيق هذه التقنية سنوظفها لتثبيت درجة الحرارة داخل مفاعل كيميائي نو تفاعل غير عكسي. في هذه المذكرة نستعمل أيضا الخوارزميات الجينية لتقليص عدد القواعد الغامضة ثم الدالات الانتماية و عدد القواعد الغامضة معا. نبين كذلك أن الخوارزميات الجينية تستطيع أن تكون بديلا عن خبير في الميدان وهذا بإعطائنا أحسن قاعدة معرفية أو عندما لا يوجد معلومات كافية لمراقبة النظام .

Abstract

In this work we will explain how the standard genetic algorithms (S.G.A) and hierarchical (H.G.A.) can carry out an automatic research of the base of knowledge of a fuzzy controller. To implement this technique, will use a nonlinear dynamic system which consists in stabilizing the temperature of a chemical engine of the type CSTR. In this present report will also use we them to optimize the fuzzy rules then the simultaneous optimization of the functions of membership and the fuzzy rules .In the same way we will show as the genetic algorithms constitute a good alternate to provide the best of base knowledge and that without needing an expert in the field or when we does not have sufficient data of the system to control

REMERCIEMENTS

Nous remercions tout d'abord, DIEU le Tout Puissant qui par sa générosité, nous a donné la volonté et la patience durant toutes nos longues d'années de préparation. Nous le remercions aussi de nous avoir permis d'arriver à bout de ce mémoire.

Je tiens beaucoup à remercier notre aimable Promoteur Mr SALHI Hassen pour ses précieux conseils et orientations, ainsi que son assistance, qui m'a permis d'aboutir à la fin de ce mémoire.

Que soit également remercié Mr HADJ SADOK M'Hamed et Mr BENZINEB Omar pour leurs collaborations et encouragements sans oublier Monsieur Abdelmalek du département de chimie Industrielle.

J'exprime ma plus profonde gratitude à Monsieur le Professeur GESSOUM Abderrezak pour m'avoir accueilli au sein de son laboratoire de recherche. Je le remercie d'avoir bien voulu accepter la présidence de ce jury.

Je remercie également toute l'équipe de chercheurs du Laboratoire d'Automatique pour leur présence amicale et le soutien apporté.

Enfin, on ne saurait oublier de remercier les Membres de ce jury qui ont eu l'amabilité d'examiner notre travail, ainsi à tous ceux qui ont contribué de près ou de loin à ce modeste mémoire. Les derniers remerciements vont à ma Chère Mère et à ma Famille car sans eux, rien de ce qui est entre vos mains aujourd'hui n'aurait été réalisé

TABLE DES MATIERES

RESUME	1
REMERCIEMENTS.	2
TABLE DES MATIERES	3
LISTE DES ILLUSTRATIONS, GRAPHIQUES ET TABLEAUX	5
INTRODUCTION	8
1. MODELISATION MATHEMATIQUE DU C.S.T.R	12
1.1 Introduction.	12
1.2 Caractéristique des réacteurs chimiques	12
1.3 Classification des réacteurs chimiques	12
1.4 Caractéristique du réacteur C.S.T.R.	13
1.5 Bilan de matière et d'énergie dans les réacteurs chimiques	14
1.6 Modélisation	15
1.7 Conclusion	20
2. LES ALGORITHMES GENETIQUES	21
2.1 Introduction.	21
2.2 Principe des algorithmes génétiques.	21
2.3 Les opérateurs génétiques.	24
2.4 Mise en oeuvre d'un algorithme génétique.	29
2.5 Algorithmes génétiques hiérarchiques.	30
2.6 Algorithme génétique en parallèle.	35
2.7 Conclusion.	37
3. OPTIMISATION DES PERFORMANCES D'UN CONTROLEUR FLOU PAR AG	38
3.1 Introduction.	38
3.2 Principe de la logique floue	39
3.3 Généralités sur les sous ensembles flous.	39
3.4 Structure générale d'un contrôleur flou.	41
3.5 Application de l'A.G. au contrôleur flou.	44
3.6 Conclusion.	55
4. SIMULATIONS ET VALIDATIONS DES RESULTATS.	56
4.1 Introduction.	56
4.2 Objectif.	56
4.3 Simulations des résultats.	57
4.4 Résultats d'un contrôleur flou classique.	76
4.5 Conclusion.	80

CONCLUSION	81
APPENDICES	84
A. Liste des abréviations	84
REFERENCES	85

LISTE DES ILLUSTRATIONS, GRAPHIQUES ET TABLEAUX

Figure 1.1	Schéma du réacteur chimique parfaitement agité	13
Figure 1.2	Variation des paramètres de réactions dans un CSTR	14
Figure 1.3	Stratégie de simulation d'un processus	15
Figure 1.4	Coupe du réacteur avec les différents échanges de chaleur	18
Figure 1.5	Évolution des paramètres C_a , T et T_c du réacteur quand $Q_c=0$	20
Figure 2.1	Organigramme de fonctionnement d'un A.G.S	22
Figure 2.2	Méthode de croisement en un point	26
Figure 2.3	Exemple de croisement à deux points.	27
Figure 2.4	Exemple de croisement uniforme	27
Figure 2.5	Principe de fonctionnement de la mutation binaire	28
Figure 2.6	Mécanisme de la transcription T.A.F. du promoteur	32
Figure 2.7	Structure hiérarchique multi niveaux du chromosome.	32
Figure 2.8.a	Cas d'un chromosome à un niveau de contrôle.	33
Figure 2.8.b	Cas d'un chromosome à deux niveaux de contrôle.	33
Figure 2.9	Organigramme d'évolution d'un H.G.A	34
Figure 2.10	Structure d'un AG global	35
Figure 2.11	Migration en boucle	36
Figure 2.12	Migration en voisinage.	36
Figure 2.13	Migration libre	36
Figure 2.14	Exemple de topologie de diffusion (4,4)	37
Figure 3.1	Structure générale d'un système d'inférence flou	42
Figure 3.2	Structure générale d'un contrôleur flou en chaîne direct ...	42
Figure 3.3	Structure d'un contrôleur flou proportionnel à deux entrées piloté par un AG	44
Figure 3.4	Structure d'un contrôleur flou PD piloté par un AG.	45
Figures 3.5	Répartition des fonctions d'appartenance de : a : l'erreur de concentration b : l'erreur de température c : du débit réfrigérant.	46

Figure 3.6	Schéma des pics et écartements des triangles et trapèzes	47
Figure 3.7	Configuration d'un chromosome du type AGS	48
Figure 3.8	Méthode de codage du chromosome du type HGA1	49
Figure 3.9	Organigramme du cycle génétique du type HGA1	50
Figure 3.10	Codage des fonctions d'appartenance	52
Figure 3.11	Codage et configuration du chromosome du type HGA2	53
Figure 3.12	Organigramme du cycle génétique du type HGA2	54
Figure 3.13	Cas où l'univers de discours présente des régions indéfinies.	55
Figure 3.14	Méthode de recouvrement des fonctions inactives F1, F5 et F6	55
Figure 4.1	Choix respective des fonctions d'appartenance de : erc, ert , Qc par l'AGS	58
Figure 4.2	Signaux de sorties de Ca, T, et débit.	58
Figure 4.3	Courbe de suivie de la qualité des générations	59
Figure.4.4	Nature du bruit de température dans Tc	60
Figure 4.5	Choix respective des fonctions d'appartenance de : erc, ert , Qc par l'A.G.S	60
Figure 4.6	Evolution des signaux de sorties de Ca, T, débit et erc	61
Figure 4.7	Courbes de suivie de la qualité des générations	61
Figure 4.8	Déduction des fonctions d'appartenance de : erc, ert , Qc par l'AGS	62
Figure 4.9	Signaux de sorties de Ca,T, débit et erc.	62
Figure 4.10	Courbes de suivie de qualité des générations	63
Figure 4.11	Choix des fonctions d'appartenance de erc, derc, dQc déduite par l'AGS	64
Figure 4.12	Signaux de sorties de Ca,T, débit et erc	64
Figure 4.13	Courbes de suivie de qualité des générations	65
Figure 4.14	Signaux de sorties de Ca,T,débit et erc	66
Figure 4.15	Courbes de suivie de la qualité des générations	66
Figure 4.16	Choix respective des fonction d'appartenance de erc, derc , dQc	67
Figure 4.17	Signaux de sorties de Ca,T, débit et erc	68
Figure 4.18	Courbes de suivie de qualité des générations ...	69
Figure. 4.19	Nature du bruit de température dans le réfrigérant	70
Figure 4.20	Signaux de sorties de Ca,T, Débit et erc	70
Figure 4.21	Courbes de suivie de qualité des générations	71
Figure 4.22	Répartition des fonctions d'appartenance de erc, derc et dQc déduite par l'HGA2 sans bruit	72

Figure 4.23	Signaux de sorties de Ca,T, débit et erc	73
Figure 4.24	Courbes de suivie de qualité des générations	74
Figure 4.25	Répartition des fonctions d'appartenance de erc, derc et dQc déduite par l'H.G.A.2 en présence du bruit	75
Figure 4.26	Signaux de sorties de Ca,T, débit et erc en présence du bruit	75
Figure 4.27	Courbes de suivie de qualité des générations	76
Figure 4.28	Répartition équirépartie des fonctions d'appartenance a : Erreur de la concentration $-0.06 \leq \text{erc} \leq 0.06 \text{ mol/l}$ b : Dérivée de l'erreur de concentration $-0.02 \leq \text{derc} \leq 0.02 \text{ mol/l}$ c : Dérivée du débit $-8 \leq \text{dQc} \leq 8 \text{ l/min}$	77
Figure 4.29	Signaux de sorties de Ca,T, débit et erc	78
Figure 4.30	Répartition équirépartie des 7 fonctions d'appartenance	78
Figure 4.31	Signaux de sorties de Ca, T, débit et erc (7 Fonctions d'appartenance)	79
Tableau 1.1	Classification des réacteurs chimiques	13
Tableau 1.2	Constantes thermodynamiques chimiques du CSTR	19
Tableau 3.1	Classification des paramètres d'un système d'inférence floue	43
Tableau 3.2	Tableau de décision de la sortie	47
Tableau 3.3	Table de décision de la dérivée de sortie dQc	49
Tableau 4.1	Conclusion des règles floues déduites par l'AGS	59
Tableau. 4.2	Conclusion des règles floues optimisées par le HGA1	69
Tableau 4.3	Tableau résumant les règles floues optimisées par le HGA1 en présence du bruit	71
Tableau 4.4	Conclusion des règles floues optimisées par le HGA2	74
Tableau 4.5	Table des règles floues optimisées par le HGA2	76
Tableau 4.6	Conclusion des 25 règles floues classiques	77
Tableau 4.7	Conclusion des 49 règles floues classiques	79
Tableau 4.8	Tableau récapitulatif des résultats obtenus	80

INTRODUCTION

De nos jours, les systèmes de régulation automatique pour la commande des processus sont intégrés dans de nombreuses applications, tant dans le domaine scientifique que technologique. Les systèmes devenant de plus en plus complexes ne cessent de faire appel à des compétences de plus en plus exigeantes, qui font que les performances des régulateurs utilisés deviennent de plus en plus améliorées mais demandant un personnel encore plus qualifié. Les méthodes de réglage conventionnelles comme la commande optimale, la commande adaptative ou la commande robuste, se basent sur une connaissance plus ou moins précise du modèle mathématique du système à réguler. Lorsque le système est fortement non linéaire, imprécis ou très complexe, il est parfois impossible de définir un modèle mathématique satisfaisant. Dans ce cas, les régulateurs conventionnels sont difficilement utilisables.

Dans ce mémoire, on abordera le problème de configuration dynamique des paramètres de fonctionnement du régulateur où on considérera que les paramètres du système sont fixes ou variables. Le régulateur analyse l'évolution de la sortie du système et éventuellement celle de certaines de ses variables internes pour déduire la commande nécessaire du système.

Comme on peut le remarquer dans la figure a., le régulateur est placé en cascade, et la question qui se pose est : comment peut-on, dans ces conditions, définir la structure du régulateur qui assure l'ajustement dynamique des paramètres, de telle sorte que le fonctionnement global se trouve amélioré ?

En ce moment, on utilise de plus en plus des régulateurs qui font appel à l'intelligence artificielle pour résoudre ce genre de problème et qui regroupent principalement la logique floue [36], les réseaux de neurones [37][38] et les algorithmes génétiques [1]. Ils constituent une voie prometteuse pour aborder ce problème particulier de

régulation. Ces méthodes font intervenir des mécanismes qui ont été observés et étudiés dans des domaines de recherche initialement très éloignés de l'informatique : linguistique et raisonnement humain pour la logique floue, physiologie humaine et animale pour les réseaux de neurones ou encore génétique et sélection naturelle pour les algorithmes évolutionnistes.

Afin de commander un système, et si on utilisera un contrôleur, basé sur des concepts qui font intervenir des connaissances acquises par un expert humain, ceci peut se faire avec la logique floue, car elle a tous les outils nécessaires pour donner le réseau de règles avec une grande fiabilité et cela à partir d'une base de connaissance acquise par l'expert humain.

L'intérêt principal de cette méthode est qu'elle ne nécessite pas de connaître le fonctionnement approfondi du système, mais simplement la façon de le commander. Les connaissances sont exploitées sous une forme linguistique par l'intermédiaire de règles comme : « si condition **alors** action ».

Lorsque les règles de la commande ne sont pas connues avec précision, la conception du contrôleur flou se complique et peut être assimilée à un problème d'optimisation qui consiste à : Comment choisir les paramètres du contrôleur afin d'assurer un fonctionnement optimal selon certains critères ? De nombreuses techniques d'optimisation ont été décrites dans la littérature et pour rester dans le domaine de l'intelligence artificiel, nous montrerons comment utiliser les algorithmes génétiques pour optimiser automatiquement les paramètres de la structure d'un contrôleur flou afin de commander un réacteur chimique du type C.S.T.R. (Continuous Stirred Tank Reactor)

Un algorithme génétique est une méthode d'exploration de l'espace des solutions possibles d'un problème qui utilise des mécanismes similaires à ceux de la sélection naturelle. A chaque itération, l'algorithme génétique évalue un ensemble limité de solutions, et retient les meilleures d'entre elles (phase de sélection). Ces solutions sont combinées (phase de croisement) et éventuellement légèrement modifiées (phase de mutation) avant d'être à nouveau évaluées. A l'issue de ce processus itératif, on aboutit sous certaines conditions à une solution acceptable du problème.

Le premier chapitre comportera un bref aperçu sur la modélisation du réacteur

chimique C.S.T.R choisi par la présente application, et qui facilitera la compréhension du mécanisme de fonctionnement afin de bien diagnostiquer les résultats de sorties quand il est commandé par le contrôleur flou. De même il permettra de donner un outil supplémentaire pour valider les techniques utilisées.

Dans le deuxième chapitre on expliquera de façon succincte les algorithmes génétiques, et qu'ils peuvent être un moyen d'optimisation. On expliquera son développement puis l'intérêt des mécanismes des opérateurs génétiques : sélection des individus, croisement et mutation. De même on présentera plusieurs méthodes de représentation des informations (codage binaire, codage réel) à l'aide d'un un algorithme génétique standard ou hiérarchique

Le troisième chapitre est consacré à la description de la méthode de synthèse d'un contrôleur flou par un algorithme génétique. Au début sont énoncés les concepts fondamentaux de la logique floue, puis de voir comment on peut exprimer selon un formalisme simple des informations très diverses (données incertaines ou imprécises, . . .) en connaissances écrites sous une forme linguistique, puis on abordera la structure générale d'un contrôleur flou et les différents sous ensembles qui le constituent. Ensuite on décrira plusieurs méthodes de configuration du chromosome pour différentes stratégies de contrôle (deux boucles de retour et une boucle de retour), ainsi que les paramètres sur lequel repose en grande partie l'efficacité de la phase d'optimisation.

On présentera aussi une méthode de codage mixte (code binaire, code réel) qui permet de décrire tous les paramètres régissant le fonctionnement d'un contrôleur flou du type P ou PD .Ensuite on exposera la stratégie d'optimisation des règles, puis la stratégie d'optimisation simultanée des règles et des fonctions d'appartenance en utilisant les algorithmes génétiques.

Enfin dans le dernier chapitre, il sera consacré aux résultats et aux interprétations des différentes simulations obtenues par les différentes stratégies d'optimisations : l'AG Standard et l'AG Hiérarchique, qui assurent le réglage dynamique des paramètres du contrôleur flou de manière à déduire le débit du réfrigérant nécessaire pour stabiliser la température à l'intérieur du réacteur chimique C.S.T.R. afin d'obtenir la concentration

désirée. De même, des simulations seront présentées lorsque le système est soumis à des perturbations de température au niveau du réfrigérant pour tester la robustesse du contrôleur hybride.

CHAPITRE 1

MODELISATION MATHEMATIQUE DU REACTEUR C.S.T.R.

1.1. Introduction

On appelle réacteur chimique tout appareil permettant de réaliser une réaction chimique qui permet de transformer des espèces moléculaires en d'autres espèces moléculaires dans le but de produire une ou plusieurs espèces chimiques ou d'éliminer d'un mélange un ou plusieurs composés [15].

Généralement un réacteur présente des caractéristiques de fonctionnement qui conditionnent les installations placées en amont (préparation des charges de réactifs, choix de la température et la pression) et les dispositifs placés en aval (pour la séparation des produits), c'est un élément intermédiaire qui participe énormément à l'amélioration du coût d'investissement et de la consommation en matière d'énergie. En ce sens il est véritablement « le cœur du procédé » qui requiert toute l'attention de l'ingénieur [16].

1.2. Caractéristique des réacteurs chimiques

Un réacteur chimique exige des caractéristiques très diverses. Toutefois tous les réacteurs doivent répondre généralement aux conditions principales suivantes [17] :

- assurer un rendement élevé.
- fournir un taux de transformation le plus grand possible.
- consommer le moins d'énergie pour le transport et l'agitation des réactifs.
- être suffisamment simple et pas cher.
- utiliser au mieux la chaleur des réacteurs exothermiques.
- avoir un fonctionnement sûr et permettre un contrôle automatique du procédé.

1.3. Classification des réacteurs chimiques.

On distingue plusieurs types de réacteurs qui dépendent essentiellement des principaux critères suivants [18]:

- La circulation du mélange réactionnel.
- L'évolution dans le temps.
- L'état thermique.
- Le degré de mélange des substances en réaction à l'intérieur du réacteur.

Le tableau 1.1 résume la classification et les types de réacteurs les plus utilisés

Critère	Type de réacteur
Circulation du mélange réactionnel	- Réacteur fermé - Réacteur semi- fermé - Réacteur ouvert
Évolution dans le temps	En régime transitoire En régime permanent
État thermique	Réacteur adiabatique. Réacteur isothermique.
Degré de mélange des substances en réaction	Réacteur parfaitement agité Réacteur à écoulement piston

Tableau 1.1 : Classification des réacteurs chimiques

Le réacteur de notre application est un réacteur continu parfaitement agité connu sous la dénomination anglo-saxonne : Continuous Stirred Tank Reactor (CSTR). Une description assez succincte sera présentée pour faciliter sa modélisation est nécessaire.

1.4.Caractéristique du réacteur C.S.T.R.

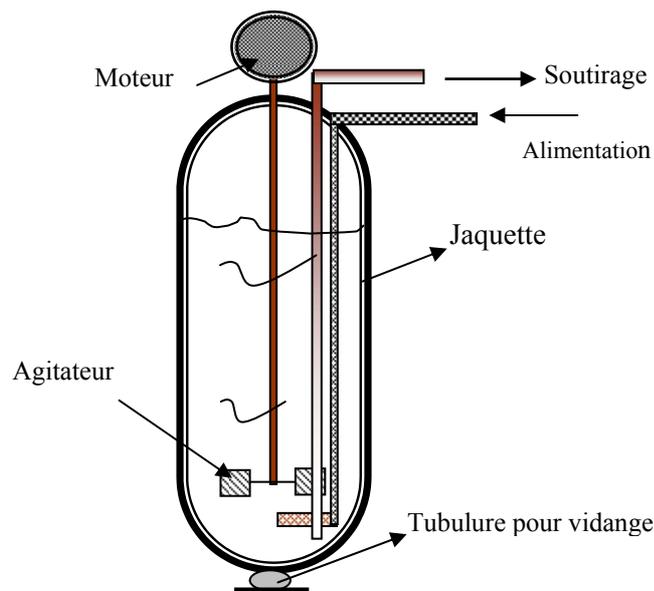


Figure 1.1 : Schéma du réacteur chimique parfaitement agité

Ce type de réacteur étant continu, ouvert et parfaitement agité, il appartient donc à la famille des réacteurs homogènes. L'ensemble du volume des réactifs à l'intérieur du réacteur, possède une concentration identique avec une température uniforme. Comme le montre la figure 1.1, il est alimenté par un réactif frais, et le produit de la réaction est soutiré de manière continue.

Ce réacteur présente une réaction exothermique, car la conversion du réactif de A vers B s'accompagne d'un dégagement de chaleur. Celle-ci augmente progressivement jusqu'à ce que la température d'activation relative à la réaction chimique désirée ($A \longrightarrow B$) soit atteinte. Mais dès que la température dépasse un certain seuil T_a , le réactif résultant B désiré va se décomposer pour donner un troisième réactif C indésirable. Une autre conséquence plus grave est qu'elle peut engendrer l'explosion de l'installation dû à l'emballement thermique.

D'après la figure 1.2, on peut remarquer que la température et le taux de conversion des réactifs sont uniformément croissants pour un CSTR. De même, la vitesse de conversion passe par un maximum, puis elle tend vers zéro tel que le taux de conversion s'approche de 100%. Delà, on voit l'importance du point maximum de la vitesse de conversion sur le rendement de la réaction, et la nécessité de le prendre en considération, et cela en agissant sur la température d'une manière très prudente pour que le système fonctionne dans les conditions voulues. Un refroidissement performant doit être intégré au bon fonctionnement de l'installation.

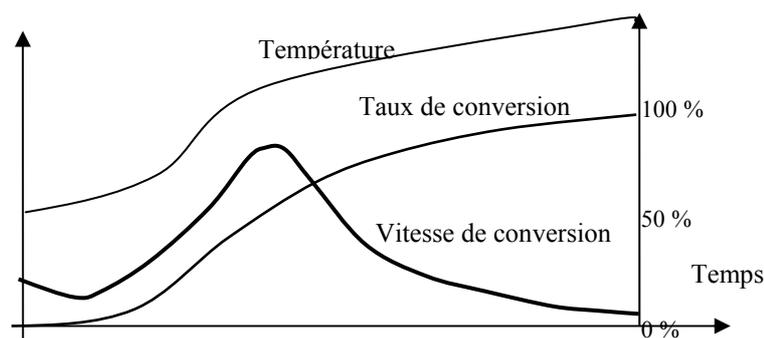


Figure 1.2 : Variation des paramètres de réactions dans un CSTR

En général, un bilan de matière ou d'énergie doit être exprimé dans une zone tel que le volume réactionnel (siège d'une transformation chimique), effectue un échange

de matière et d'énergie avec le milieu extérieur.

1.5.1 Bilan de la matière

Pour établir un bilan de la matière dans un réacteur, il faut qu'il existe une variation des paramètres opératoires. Dans le cas d'une cuve parfaitement agitée la conservation de la matière exige qu'à chaque instant [16] :

$$\left(\text{Flux de } A_i \text{ entrant} \right) + \left(\text{débit de production} \right) = \left(\text{flux de } A_i \text{ sortant} \right) + \left(\text{débit d'accumulation de } A_i \right)$$

1.5.2 Bilan énergétique

Dans le cas d'un réacteur ouvert en région quelconque, et en appliquant le premier principe de la thermodynamique (conservation d'énergie) la forme générale se réduit à une balance d'enthalpie de la forme :

$$\left(\text{Flux d'énergie interne} \right)_{\text{entrant}} + \left(\text{Chaleur ajoutée au} \right)_{\text{système par réaction}} = \left(\text{Flux d'énergie interne} \right)_{\text{sortant}} + \left(\text{énergie accumulée} \right)_{\text{dans le réacteur}}$$

Un réacteur est dit instable lorsqu'une légère augmentation de la température produira un accroissement de la vitesse de production de la chaleur plus importante que la vitesse d'élimination de la chaleur [19].

1.6 Modélisation.

La modélisation et la simulation des procédés sont devenues dans de nombreux secteurs industriels, un moyen efficace et souvent indispensable dans la représentation et l'analyse d'un processus et ceci dans le but de concevoir un équipement ou une installation devant répondre aux exigences de sécurité et de productivité optimales. Parmi les avantages de la simulation, on peut citer, la minimisation de l'effort expérimental souvent coûteux et parfois inadéquat. D'autre part, elle permet de prédire le mauvais fonctionnement et les risques les plus divers. On peut résumer la stratégie de simulation comme le schématise la figure 1.3 en plusieurs étapes importantes [20].

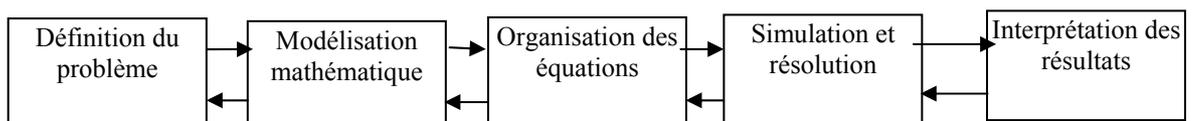


Figure 1.3 : Stratégie de simulation d'un processus

1.6.1 Modèle Mathématique. [18]

Comme la réaction est du premier ordre et irréversible $A \longrightarrow B$, et en prenant le cas où la conversion d'un réactif A est unique en milieu indilatable et si x est son taux de conversion, et en utilisant la loi Arrhenius, la vitesse de conversion r de la réaction sera :

$$r = K \cdot C_a \quad (1.1)$$

comme ,

$$K = K_0 \cdot \exp^{-(E./RT)} \quad \text{et} \quad C_a = C_{a0} \cdot (1-x)$$

Par conséquence ,

$$r = K_0 \cdot \exp^{-(E./RT)} \cdot C_a \quad (1.2)$$

où

C_a : concentration du réactif.

K_0 : facteur de fréquence constant.

C_{a0} : concentration initiale du réactif a l'instant $t = 0$.

K : constante de vitesse.

T : température en °Kelvin.

E : énergie d'activation.

R : constante des gaz parfaits.

a) En faisant intervenir le bilan de la matière qui se traduit par la conservation du débit molaire c'est à dire,

$$F_{\text{entrant}} + F_{\text{produit}} = F_{\text{sortant}} + F_{\text{accumulé}} \quad (1.3)$$

Alors en remplaçant chaque débit molaire par son expression dans 1.3 on obtient :

$$Q \cdot C_{a0}(t) + \gamma \cdot r \cdot V = Q \cdot C_a(t) + d\eta_a/dt \quad (1.4)$$

γ : coefficient stchiométrique égale à -1

η_a : molarité .

Q : débit volumique du réactif.

V : volume du réactif.

F : débit molaire en (mole/min).

Puisque $\eta_a = C_a \cdot V$ et en utilisant 1.1 ,1.2 et 1.4 alors ,

$$Q \cdot C_{a0}(t) - K \cdot C_a(t) \cdot V = Q \cdot C_a(t) + dC_a(t) \cdot V/dt \quad (1.5)$$

Comme le volume de la cuve est constant, l'équation 1.5 devient :

$$Q(C_{a0} - C_a(t)) - K_0 \left(\exp - \frac{E}{R.T} \right) \cdot C_a(t) \cdot V = V \cdot \frac{dC_a(t)}{dt} \quad (1.6)$$

D'où

$$\frac{dC_a(t)}{dt} = \frac{Q}{V} (C_{a0} - C_a(t)) - K_0 \cdot C_a(t) \cdot \left(\exp - \frac{E}{R.T(t)} \right). \quad (1.7)$$

b) Le bilan d'énergie se traduit maintenant par la conservation de l'énergie au niveau du réacteur, l'équation est de la forme :

$$F_e \cdot \Delta H_e + \gamma \cdot r \cdot V \cdot \Delta H_r = F_s \cdot \Delta H_s + q + dU/dt \quad (1.8)$$

Avec :

F_e et F_s : débits volumiques des réactifs entrant et sortant égales à Q

ΔH_e : flux d'enthalpie entrant

γ : coefficient stœchiométrique égale à -1

ΔH_r : flux d'enthalpie de réaction

ΔH_s : flux d'enthalpie sortant.

V : volume du réactif.

q : quantité d'énergie reçue par le réfrigérant (ou l'extérieur)

dU/dt : accumulation d'énergie

En remplaçant chaque terme par son expression dans 1.8, on aura :

$$Q \cdot \rho \cdot C_p (T_0 - T_r) - K \cdot C_a(t) \cdot V \cdot \Delta H_r = Q \cdot \rho \cdot C_p (T_s(t) - T_r) + q + d(m \cdot C_p \cdot T(t))/dt \quad (1.9)$$

$$Q \cdot \rho \cdot C_p \cdot (T_0 - T_r) - C_a(t) \cdot \Delta H_r \cdot V \cdot K_0 \cdot \exp^{-\frac{E}{R \cdot T_s}} = Q \cdot \rho \cdot C_p (T_s(t) - T_r) + q + \rho \cdot V \cdot C_p \cdot dT(t)/dt$$

$$\frac{dT(t)}{dt} = \frac{Q}{V} (T_0 - T(t)) - \frac{\Delta H_r}{\rho \cdot C_p} \cdot C_a(t) \cdot K_0 \cdot \left(\exp - \frac{E}{R \cdot T_s(t)} \right) - \frac{q}{\rho \cdot V \cdot C_p} \quad (1.10)$$

Q : débit du réactif.

ρ : masse volumique du réactif .

C_p : chaleur spécifique du réactif.

$T_r = 0^\circ K$: température de référence.

T_0 : température initiale du réactif .

$T_s(t) = T(t)$: température de sortie du réactif .

c) En faisant maintenant intervenir le bilan énergétique au niveau de la jaquette et du milieu réactionnel, qui se traduit simplement par un échange de chaleur entre le réactif et le réfrigérant (voir figure 1.4) :

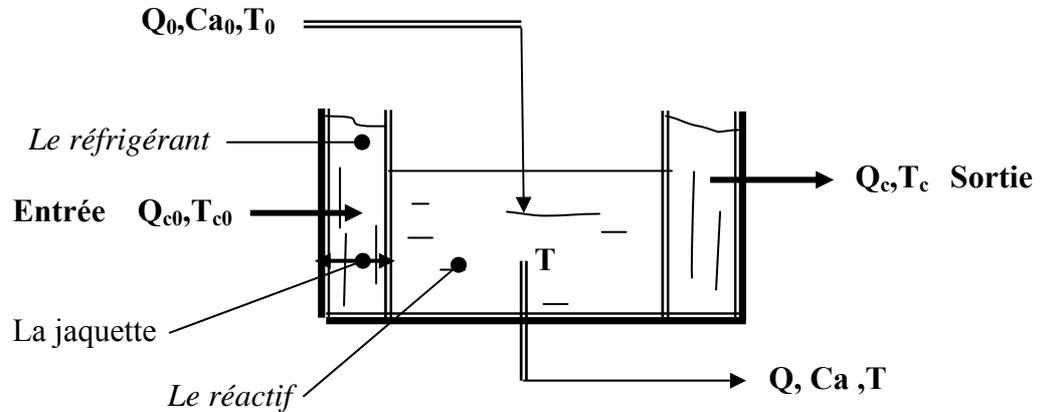


Figure 1.4 : Coupe du réacteur avec les différents échanges de chaleur

Puisque le réacteur est parfaitement agité alors les concentrations et les températures sont uniformément réparties dans le réacteur, et en supposant que les débits dans le réacteur sont constants alors $Q_0 = Q$ et $Q_{c0} = Q_c$.

De même, l'équation de transfert de chaleur entre le réfrigérant et le milieu réactionnel est :

$$q = h.A.(T - T_c) \quad (1.11)$$

Avec :
 h : coefficient de transfert de chaleur.
 A : surface d'échange.
 T : température du milieu réactionnel.
 T_c : température du réfrigérant.

or le bilan d'énergie au niveau de l'échangeur (réfrigérant et réactif)(voir figure 1.4), est :

$$F_e . \Delta H_e + q = F_s . \Delta H_s + dU/dt \quad (1.12)$$

Avec :

F_e et F_s : débits volumiques du réfrigérant entrant et sortant égales à Q_c

Et en remplaçant chaque terme dans l'expression 1.12, on obtient :

$$Q_c . [\rho_c . C_{pc} (T_{c0} - T_r)] + q = Q_c . [\rho_c . C_{pc} (T_c(t) - T_r)] + \rho_c . V_c . C_{pc} dT_c(t)/dt \quad (1.13)$$

q : quantité d'énergie reçue par le réactif.

Q_c : débit volumique du réfrigérant.

T_c : température de sortie du réfrigérant.

T_{c0} : température d'entrée du réfrigérant.

V_c : volume du réfrigérant.

En utilisant l'équation 1.11 avec $h_a = h.A$, l'équation 1.13 devient :

$$\frac{dT_c(t)}{dt} = \frac{Q_c}{V_c} (T_{c0} - T_c(t)) + \frac{h_a}{\rho_c \cdot V_c \cdot C_{pc}} (T(t) - T_c(t)) \quad (1.14)$$

Enfin le model mathématique du réacteur CSTR est donné par trois équations différentielles non linéaires du premier ordre 1.7, 1.10 et 1.14 :

$$\left\{ \begin{array}{l} \frac{dC_a(t)}{dt} = \frac{Q}{V} (C_{a0} - C_a(t)) - K_0 \cdot C_a(t) \cdot \left(\exp - \frac{E}{R \cdot T(t)} \right) \\ \frac{dT(t)}{dt} = \frac{Q}{V} (T_0 - T(t)) - \frac{\Delta H_r}{\rho \cdot C_p} \cdot C_a(t) \cdot K_0 \cdot \left(\exp - \frac{E}{R \cdot T} \right) - \frac{h_a}{\rho \cdot V \cdot C_p} (T(t) - T_c(t)) \\ \frac{dT_c(t)}{dt} = \frac{Q_c}{V_c} (T_{c0} - T_c(t)) + \frac{h_a}{\rho_c \cdot V_c \cdot C_{pc}} (T(t) - T_c(t)) \end{array} \right. \quad (1.15)$$

Les valeurs des paramètres du système considéré sont résumées dans le tableau 1.2

Paramètre	Description	Valeur Nominale
Q	Débit du réactif	100 l/min
V	Volume du réacteur	100 l
K0	Constante de vitesse	$7.2 \cdot 10^{10} \text{ min}^{-1}$
E/R	Énergie d'activation	1.10^4 °K
T0	Température d'alimentation	350 °K
Tc0	Température d'alimentation du réfrigérant	350 °K
ΔH_r	Chaleur de réaction	-2.10^5 cal/mol
C_p, C_{p0}	Constante de chaleur spécifique	1 cal/g.°K
ρ_c, ρ	Densité des liquides	1.10^3 g/l
ha	Coefficient de transfert	$7.10^5 \text{ cal/min. °K}$

Tableau 1.2 : Constantes thermodynamiques chimiques du CSTR.

À l'équilibre et pour une concentration nominale voulue de $Ca = 0.1 \text{ mol/l}$. En utilisant l'équation 1.15, on peut déduire les paramètres du régime permanent du réacteur chimique qui sont : $T = 438.54 \text{ °K}$, $T_c = 425,48 \text{ °K}$ et $Q_c = 121,1 \text{ l/min}$.

Dans le cas où le dispositif de refroidissement n'est pas actionné ($Q_c = 0 \text{ l/min}$), on voit d'après la figure 1.5, que la réaction est exothermique, et elle présente une croissance rapide de la température T du réactif autour de la concentration nominale $Ca=0.1 \text{ mol/l}$ du réacteur. Il en résulte un réactif indésirable.

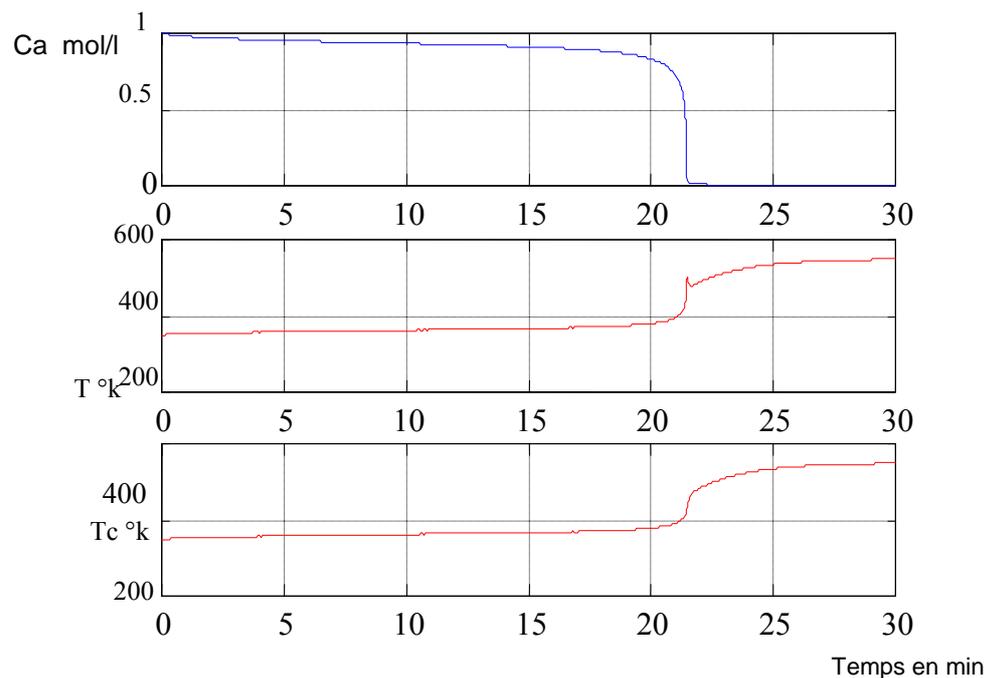


Figure 1.5 : Évolution des paramètres Ca , T et T_c du réacteur quand $Q_c=0$

1.7 Conclusion

Dans ce chapitre, on a commencé par montrer l'aspect dynamique de la réaction exothermique, en se basant sur le principe de la conservation de la matière et de l'énergie, cette étude a permis aussi de conclure qu'il y a un couplage entre les différents états (température et concentration). On peut remarquer aussi que la température est l'élément majeur qui déstabilise le réacteur. Pour arriver à des performances acceptables, il y a lieu de prévoir un contrôleur adéquat qui tient compte de ces couplages et qui commandera le débit au niveau de la jaquette afin d'obtenir le produit voulu. Ceci fera l'objet du troisième chapitre.

CHAPITRE 2

LES ALGORITHMES GENETIQUES

2.1. Introduction.

Depuis des millénaires, l'homme cherche à améliorer ses récoltes agricoles et ses races d'animaux en utilisant des opérations génétiques telles que les croisements et les mutations. Elles sont le fruit de beaucoup d'essais de transpositions de gènes naturels et cela en manipulant à l'intérieur de la cellule vivante afin d'obtenir un produit plus performant, plus résistant et à moindre coût. Cette démarche a poussé les chercheurs à développer des théories pour les utiliser dans des domaines autre que la biologie, mais tout en s'inspirant des caractéristiques biologiques [1]. Ces observations sont les fondements de la théorie des algorithmes génétiques (en abrégé A.G.) afin de résoudre une gamme de problèmes allant des questions mathématiques pures aux applications concrètes de la vie courante tel que l'optimisation multicritère, la régulation du trafic aérien, le contrôle des procédés industrielles, l'optimisation des réseaux de neurones, les systèmes d'aide à la décision [2] [3] [6]. Les algorithmes génétiques sont donc des méthodes d'exploration de l'ensemble des solutions d'un problème tout en utilisant les mêmes mécanismes que ceux intervenant dans la sélection naturelle. Ils sont donc utilisés principalement dans les domaines de l'optimisation et de l'apprentissage.

Dans la présente étude on d'optimisera la base de connaissance (constituée de fonctions d'appartenance et de règles floues) d'un contrôleur flou, en utilisant le concept des algorithmes génétiques.

2.2. Principe des algorithmes génétiques

Les algorithmes génétiques sont des algorithmes inspirés des mécanismes de la sélection naturelle de la génétique. Au cours de leur évolution, ils utilisent à la fois, le principe de la survie des individus les mieux adaptés et la propagation du patrimoine génétique. Ils ont été développés par David Goldberg [4] pour résoudre des problèmes combinatoires jugés complexes.

En biologie on manipule les gènes des chromosomes, il en va de même dans le domaine des A.G où les problèmes et les solutions seront obtenus sous forme d'un code qu'il faut décoder en un langage exploitable. L'encodage peut prendre différentes formes. L'une des plus usuelles sont les chaînes de bits. Ces dernières sont comparables aux chromosomes des systèmes biologiques, alors que les caractères sont les bits qui les composent et sont comparables aux gènes. L'ensemble des chaînes forme une structure connue sous le nom de génotype en terme biologique.

L'évolution d'une espèce d'une génération à la suivante, est à la fois une succession de recombinaisons obtenue par des opérations de croisement de chromosomes appariés, et de mutations, caractérisées par l'apparition d'un nouveau chromosome inexistant chez les individus parents et qui se traduit par le changement du code génétique de l'individu enfant. Ce changement dans la morphologie des espèces s'effectue dans le sens d'une meilleure adaptation au milieu naturel.

La disparition de certaines espèces est expliquée par la loi de la survie, selon laquelle, seuls les individus les mieux adaptés auront une longévité suffisante pour générer une descendance : c'est le phénomène de la sélection naturelle. Ainsi un algorithme génétique standard (AGS) peut être construit suivant l'organigramme de la figure 2.1.

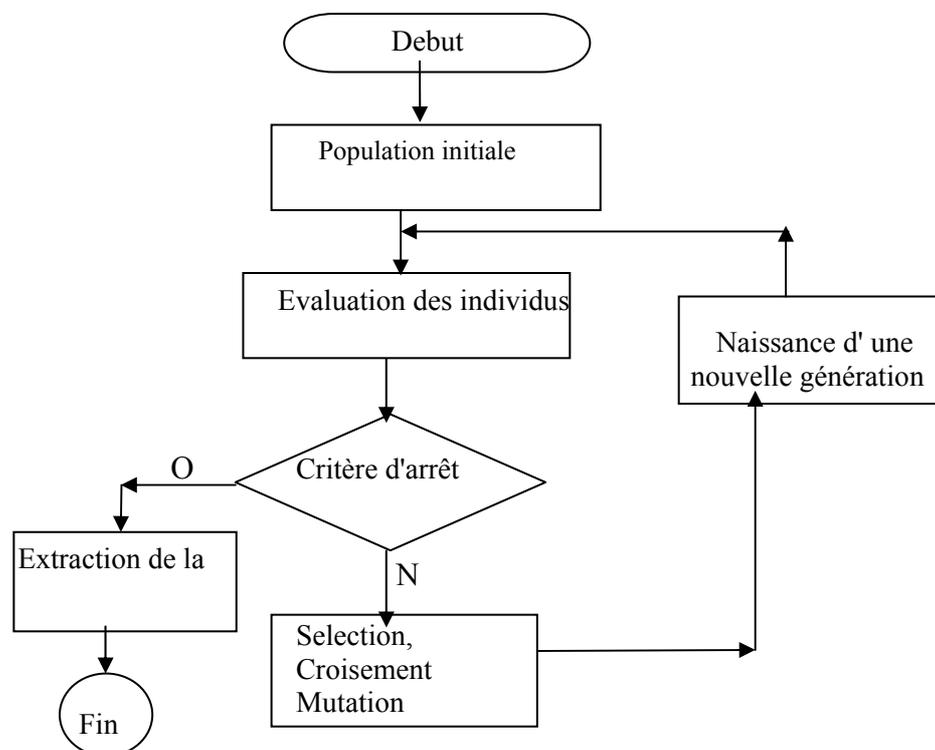


Figure 2.1 : Organigramme de fonctionnement d'un AGS.

Il est construit de manière tout à fait comparable aux mécanismes génétiques classiques. On commence par générer une population initiale de taille N qui constitue N solutions possibles. Ensuite une sélection des meilleurs chromosomes choisis par une fonction d'adéquation (ou évaluation) propre au problème à résoudre est entamée, puis on effectue le croisement des chromosomes sélectionnés, qui consiste à échanger un certain nombre de bits entre deux parents, puis on termine par une opération de mutation au niveau de chaque chromosome. On répète ces étapes jusqu'à l'obtention d'une solution acceptable ou désirée.

2.2.1. Terminologie

Avant de décrire le principe des algorithmes génétiques, il est nécessaire de présenter le vocabulaire utilisé tout au long de ce travail.

Chromosome → **chaîne** : Dans les systèmes naturels, les chromosomes sont les porteurs de l'information génétique nécessaire à la construction et au fonctionnement d'un organisme. Dans les algorithmes génétiques, les chaînes, ou chromosomes sont analogues aux chromosomes des systèmes biologiques. Ils sont les éléments à partir desquels sont élaborées les solutions.

Génotype → **structure** : Dans les systèmes naturels, l'ensemble du matériel génétique est appelé génotype. Dans les algorithmes génétiques, l'ensemble des chaînes est appelé structure.

Phénotype → **ensemble de paramètres, solution, point** : Dans les systèmes naturels, l'organisme est formé par l'interaction de l'ensemble du matériel génétique avec son environnement appelé phénotype. Dans les algorithmes génétiques, les structures décodées forment un ensemble de paramètres donnés, ou une solution ou un point dans l'espace des solutions.

Gène → **trait, détecteur** : Dans les systèmes naturels, les chromosomes sont constitués par des gènes. Dans les algorithmes génétiques, on dit que les chaînes se composent de traits ou détecteurs.

Allèle → **valeur de caractéristique** : Dans les systèmes naturels, l'allèle est une composante du gène. Les allèles sont les différentes valeurs que peuvent prendre les gènes. Dans les algorithmes génétiques, l'allèle est également appelé valeur caractéristique.

Locus → position dans la chaîne : Le locus est la position du gène dans le chromosome. Ce terme est appelé également position dans la chaîne dans les algorithmes génétiques.

Individu, organisme → **individu, chromosome** : Un organisme ou un individu biologique est une forme qui est le produit de l'activité des gènes. Dans le cadre d'un AG traditionnel, l'individu est réduit à un chromosome, on l'appelle indifféremment individu ou chromosome.

Population → **génération** : Dans les systèmes naturels, la population est un groupe d'individus. Dans les algorithmes génétiques, la population est l'ensemble des individus ou des chromosomes.

2.3. Les opérateurs génétiques [5] [6]

Les algorithmes génétiques sont appliqués à une population initiale de chromosomes de façon à produire des populations successives et de qualité meilleure grâce aux diverses étapes citées précédemment, connues sous le nom des opérateurs génétiques. Maintenant les principales opérations seront présentées.

2.3.1. La sélection.

Comme son nom l'indique, la sélection vise à élire une population enfant à partir d'une population parent. Les meilleurs individus peuvent être choisis plusieurs fois dans la prochaine génération, alors que les moins aptes auront moins de chance de l'être; cette dernière se fera aléatoirement en fonction de la qualité d'adéquation liée au problème posé. Ce choix est réalisé par tirage au sort parmi les N individus, en tenant compte d'une probabilité de sélection affectée à chacun d'eux.

Un individu a d'autant plus de chance d'être sélectionné que sa fonction d'évaluation f_e (terme anglo saxon : fitness) prend une valeur importante. Pratiquement, la probabilité p_i d'évolution d'un individu c_i est définie par :

$$p_i = \frac{f_e(c_i)}{\sum_{j=1}^N f_e(c_j)} \quad (2.1)$$

Où : $\sum_{j=1}^N f_e(c_j)$: représente la somme de toutes les valeurs des fonctions d'évaluations de chaque chromosome c_i de la population

Elle joue un rôle important dans l'accélération de la convergence de l'algorithme. Il existe plusieurs types de sélection, parmi eux, on peut citer :

2.3.1.1. La sélection par roulette (Wheel sélection).

Cette méthode a été initiée par J. Holland lui même en 1975. Elle procède en une sélection proportionnelle au niveau de la fonction d'évaluation f_e qui consiste à associer à chaque individu une probabilité de sélection p_i définie par l'équation 2.1 .On tire alors un nombre aléatoire puis on regarde quel est le chromosome sélectionné, ainsi les meilleurs individus seront plus adressés que les plus mauvais. Lors du tirage au sort, certains individus peuvent être retenus plusieurs fois alors que d'autres sont tenus à l'écart. Ceci s'effectue par le calcul d'une probabilité de sélection cumulée q_i telle que :

$$q_i = \sum_{j=1}^i p_j \quad (2.2)$$

On génère un nombre réel r aléatoirement sur l'intervalle $[0,1]$. Cette valeur est générée plusieurs fois en fonction de la taille de la population. L'individu C_i est sélectionné lorsque $q_{i-1} < r < q_i$. Mais si la qualité d'adéquation d'un individu est très supérieure à la moyenne des f_e , il constituera la majorité des individus de la population suivante, ce qui peut conduire à une perte de la diversification dans la population qui elle même, peut conduire à une convergence prématurée ou une diminution rapide de l'espace de recherche.

2.3.1.2. La sélection élitiste.

Dans cette technique, seule la moitié supérieure de la population N comprenant les meilleurs individus sont sélectionnés pour participer à la naissance de la génération suivante. On lance une roue de sélection qui représente $N/2$, autant de fois qu'il y a de population. Néanmoins, on remarque qu'il y a une forte pression de sélection qui peut conduire à une convergence prématurée vers une solution non désirée et comme tout individu peut transmettre à sa descendance des gènes qui, une fois combinés avec d'autres peuvent se révéler intéressants, ceci peut induire à une diminution du caractère de diversification dans la population ou une dérive génétique.

2.3.1.3. La sélection par tournoi.

Cette procédure a été suggérée par Wetzel en 1983. Dans cette technique deux individus sont choisis au hasard et combattent pour accéder à la génération suivante .Le chromosome ayant la meilleure qualité d'adéquation est déclaré vainqueur avec une probabilité égale à un. Cette étape est répétée jusqu'à ce que la population soit terminée .Il

est tout à fait possible que certains individus participent plusieurs fois au tournoi et s'ils gagnent, ils auront le droit d'être copiés plusieurs fois.

2.3.2. Le Croisement

C'est l'instrument majeur des A.G. Il permet l'apparition de nouveaux individus selon plusieurs procédures biologiques .Il permet aussi d'enrichir la diversité de la population en manipulant la structure du chromosome [7] [8] se traduisant par un échange d'informations.

Bien qu'il soit aléatoire, cet échange offre à l'algorithme génétique une part de leur puissance, quelques fois de bons gènes d'un parent viennent remplacer les mauvais, pour générer des fils mieux adaptés aux parents. Cette transmission est appelée héritage. Elle va conduire à des progénitures encore mieux adaptées. Le phénomène de croisement est une propriété naturelle de l'acido désoxy ribonucléique (A.D.N.) qui joue un rôle important dans la structure des AG. Dans un algorithme génétique, on définit ainsi un deuxième paramètre de fonctionnement : la probabilité de croisement p_c , qui permet de choisir entre une conservation ou une modification des chromosomes.

On distingue plusieurs techniques de croisement qui s'appliquent sur le chromosome.

2.3.2.1. Croisement en un point

On choisit aléatoirement un nombre k compris entre 1 et N où N représente la longueur du chromosome. Cette opération s'effectue entre deux chromosomes en changeant la partie située entre k et N . Ceci donnera naissance à deux autres chromosomes représentant les enfants .Ce procédé est expliqué dans la figure 2.2.

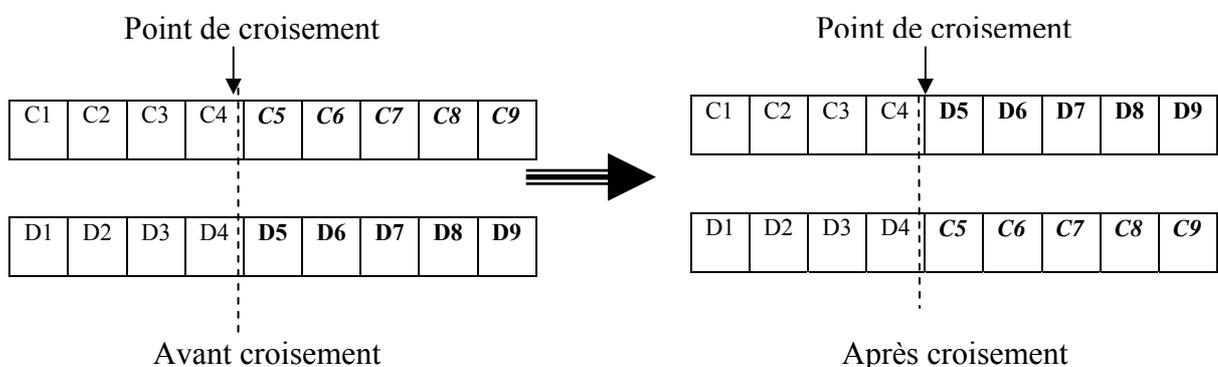


Figure 2.2 : Méthode de croisement en un point

2.3.2.2. Croisement multipoint

Dans ce cas, les deux chromosomes parents sont couplés en plusieurs points de coupures. La figure 2.3, montre un exemple de croisement à deux points

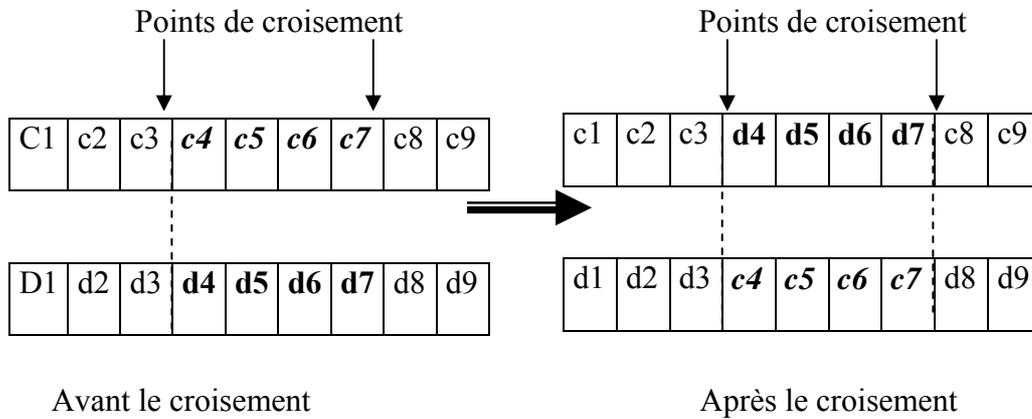
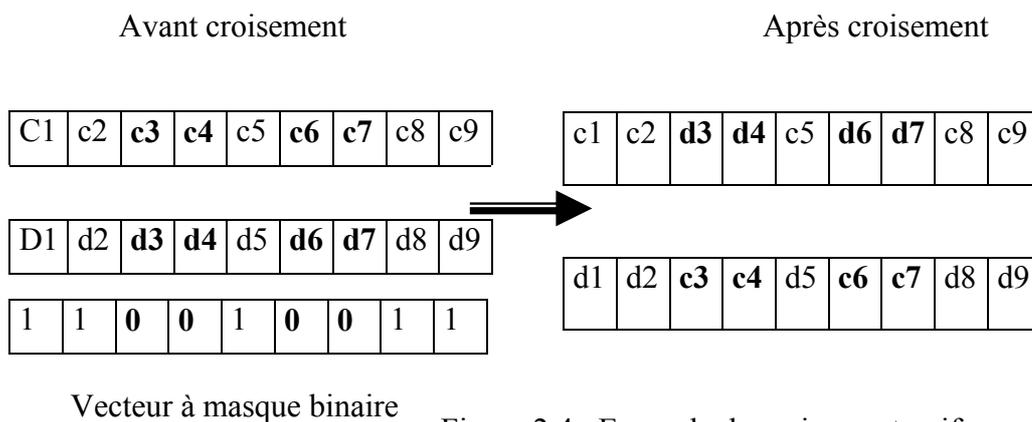


Figure 2.3 : Exemple de croisement à deux points.

2.3.2.3. Croisement uniforme

Ce type de croisement est accompli par l'intermédiaire d'un vecteur binaire généré aléatoirement appelé masque. Ce dernier permet de définir les points de croisements. Sa taille est identique à celle du chromosome et le croisement a lieu à la position des bits zéros du masque, comme l'illustre la figure 2.4.



Vecteur à masque binaire

Figure 2.4 : Exemple de croisement uniforme

On remarque que ce type de croisement est complètement différent des techniques précédentes.

Le croisement est la clé pour la puissance des AG. Il est directement lié à l'aptitude d'une population d'individus pour explorer son espace de recherche et de combiner entre

2.3.3.2. La mutation réelle aléatoire.

La mutation réelle ne se différencie de la mutation binaire que par la nature de l'élément qu'elle altère. Dans ce cas, ce n'est plus un bit binaire qu'il faut inverser mais une variable réelle qu'il faut changer.

2.4. Mise en oeuvre d'un algorithme génétique

Pour démarrer un A.G. il faut définir certains paramètres tel que :

- La taille de la population.
- les probabilités de croisement et de mutation.
- le nombre de générations.
- le type d'opérateurs génétiques.

C'est un problème de réglage qui doit être effectué pour chaque type de problème. Ceci constitue une part importante du travail de l'expérimentateur. Dans la littérature, ces paramètres changent d'une application à une autre et constituent un domaine de recherche qui est encore en exploration.

2.4.1. Taille de la population

L'A.G. nécessite la détermination du nombre d'individus qui constitue la population. Le problème qui se pose est : comment fixer la taille de la population ? Une population trop petite évolue probablement vers un optimum local peut être intéressant, mais avec une vitesse de convergence rapide. Une population convenable doit être choisie de façon à réaliser un compromis entre le temps de calcul et la qualité du résultat. Plusieurs chercheurs se sont penchés sur le problème qui consiste à déterminer une taille optimale. Certains l'ont fixé entre 20 et 100 de manière empirique, d'autres essayent de l'ajuster en respectant un taux d'erreur, où la varier au cours de l'évolution de l'algorithme [10].

2.4.2. Nombre de générations

C'est un chiffre que l'expérimentateur doit fixer. Il est préférable qu'il soit assez grand afin de visualiser l'espace de recherche. Certains utilisent un nombre entre 50 et 1000, d'autres jusqu'à 10000. Il dépend essentiellement du problème à solutionner, et

l'essentiel est de trouver des solutions acceptables par le cahier de charge en un nombre réduit de générations.

2.4.3. Probabilité des opérateurs génétiques.

Le choix de la probabilité de croisement P_c et de la probabilité de mutation P_m est un problème d'optimisation non linéaire complexe difficile à résoudre, car ces probabilités dépendent de la nature de la fonction objective, qui elle-même dépend du problème à résoudre. Cependant une formule que les auteurs considèrent est, que la probabilité P_m dépend de la taille du chromosome tel que : $1/N_p < P_m < 10/N_p$ où N_p est la longueur du chromosome.

Pour la plupart des problèmes standard, DeJong [12] conseille d'utiliser des paramètres fixés à : $N_p = 100$, $P_c = 0,6$ et $P_m = 10^{-3}$.

Quant à J.D. Schaffer [13], il propose une formule telle que la probabilité de mutation P_m , est en fonction de la taille de la population N_p et la longueur du chromosome m .

$$P_m \approx \frac{1.75}{N_p \sqrt{m}} \quad (2.3)$$

Il y a aussi d'autres chercheurs qui ont proposé un réglage dynamique de ces paramètres en prenant durant les premières itérations, des probabilités de croisement et de mutation assez élevées, ce qui permet d'explorer efficacement l'espace des solutions et par la suite, des valeurs plus faibles sont choisies, ce qui tend à stabiliser la population aux alentours des bonnes solutions. Pour ce faire, Hesser et Manner. [14] présentent une formule en faisant intervenir le numéro t de la génération traitée par l'algorithme génétique

$$p_m = p_{m0} \cdot e^{-\gamma t/2} \quad (2.4)$$

p_{m0} : désigne une probabilité initiale qui peut être similaire à celle de Schaffer

γ : est une constante qui permet de régler la décroissance en fonction du temps

Quant aux choix des probabilités de croisement, des études restent toujours en cours, mais les probabilités les plus couramment utilisées appartiennent à l'intervalle [0.3 1] et rarement inférieures à 0.3.

2.5. Algorithmes génétiques hiérarchiques

2.5.1. Introduction.

Dans le souci de développer et de résoudre des problèmes complexes, les chercheurs se sont penchés profondément sur la structure interne de l'ADN et ils ont remarqué que la structure génétique du chromosome subit un certain nombre de changements au niveau de leurs gènes, et qu'ils sont arrangés de façon hiérarchique [11].

Aussi certains gènes dominent d'autres gènes, et qu'il y a aussi des gènes actifs et d'autres inactifs. Un tel phénomène a incité les chercheurs à imiter la structure de l'ADN de façon à ce qu'on puisse avoir une structure génétique plus complexe mais évolutive.

2.5.2. Inspiration biologique.

L'A.D.N. contient toutes les informations génétiques de chaque être vivant .Elle ressemble à un livre de recettes qui permet de construire toutes les protéines nécessaires au fonctionnement des cellules. Par exemple un être humain possède près de 100.000 gènes (informations) différents, dont une partie est transformée en acide aminée (polypeptide), qui peuvent à tout moment changer de critère de fonctionnement afin d'obtenir et développer le métabolisme d'une cellule voulue .C'est une étape décisive pour pouvoir réguler, et orchestrer les différents fonctionnements des cellules.

Les travaux de Jacob et Monod en 1961 [11] ont montré qu'il existe deux types de gènes : les gènes de structure (G.S.) et les régulateurs de séquence (R.S.) appelés aussi gènes de contrôle (G.C.).

Les G.S sont codés en polypeptides ou A.R.N. (Acide Ribonucléique), alors que les RS fonctionnent comme des leaders. Ils gèrent ou commandent les G.S. à les faire marcher ou arrêter. Les R.S. qui se trouvent dans l'A.D.N. sont appelés aussi des promoteurs. Ils activent ou désactivent les G.S. issus de la transcription. Cette initialisation est gouvernée par le TAF (Transcription Acting Factor). Cette transcription peut avoir lieu si un TAF particulier est lié au promoteur, un polypeptide est produit par l'intermédiaire d'une procédure de translation, dans laquelle le message génétique est codé en A.R.N. de message (m.A.R.N.).

Par conséquent on peut dire qu'il y a développement d'une structure hiérarchique à l'intérieur de la formation de l'ADN comme le montre la figure 2.6.

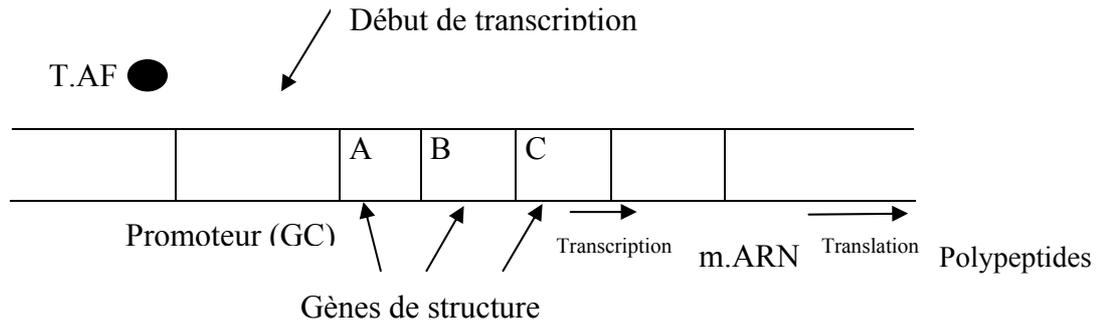


Figure 2.6 : Mécanisme de la transcription T.A.F. du promoteur.

Une autre importante découverte dans la biologie moléculaire est que dans les gènes de structure, il y a des gènes actifs et des gènes inactifs, et durant la transcription il y a un filtrage de gènes qui ne laissant passer que les gènes actifs pour former le polypeptide désiré de la cellule.

2.5.3. Structure hiérarchique de l'A.G.

Par analogie avec la structure de l'ADN, une structure multi-niveaux de contrôle peut être développée à l'intérieur du chromosome, ce qui permet d'introduire une nouvelle structure au niveau des A.G. Ceci consiste à activer un gène à l'aide d'un gène de contrôle de premier niveau. Une fois cette valeur est activée, celle ci constituera un deuxième niveau de contrôle et ainsi de suite jusqu'à la configuration totale des gènes de structure (voir figure 2.7). Par analogie avec cette inspiration, les gènes de contrôle sont représentés en binaires, et les gènes de structure peuvent être représentés en binaires, réels ou les deux à la fois.

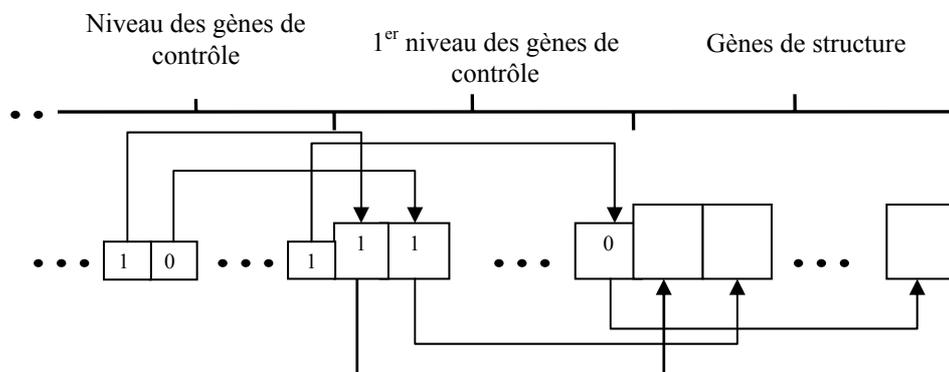


Figure 2.7 : Structure hiérarchique multi niveaux du chromosome.

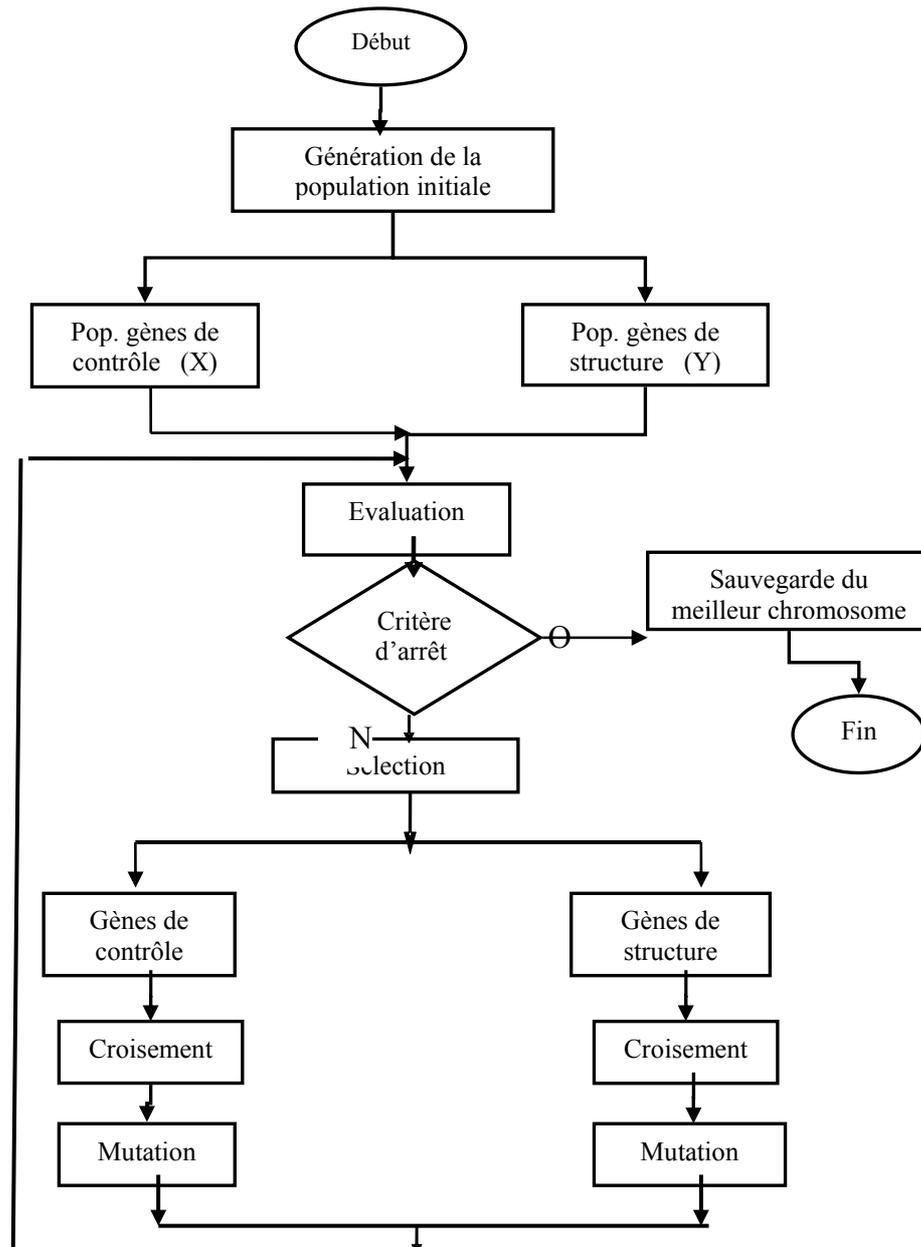


Figure 2.9 : Organigramme d'évolution d'un H.G.A.

2.5.4.1. Les opérateurs génétiques.

D'après l'organigramme de figure 2.9, l'opérateur de sélection est appliqué sur toute la population globale (X et Y) afin de donner le meilleur chromosome. Cette opération est accomplie en utilisant une fonction objective dépendante du problème ou du cahier de charge choisi. Quant aux autres opérateurs génétiques, ils sont appliqués séparément sur les gènes de contrôle et les gènes de structure auxquels on leur appliquera les opérateurs de croisement et de mutation décrits dans 2.3.2 et 2.3.3.

2.5.4.2. Probabilité de croisement et de mutation.

L'ordre de grandeur des probabilités est sensiblement le même que pour les A.G, sauf que les probabilités de gènes de contrôle sont généralement légèrement inférieures à celles des gènes de structure [11].

2.6. Algorithmes génétique en parallèle

La programmation des A.G par des architectures parallèles permet un temps de calcul relativement acceptable. On distingue plusieurs méthodes de parallélisme : globale, migration et diffusion. Ces dernières peuvent être à la base de différents chemins où le parallélisme peut être exploité tant sur la structure de la population que sur le mécanisme de recombinaison.

2.6.1. AG. global.

L'A.G. global traite toute la population comme ayant un seul mécanisme de production. Il peut être implémenté dans des multiprocesseurs à mémoires réparties dans lesquels les chromosomes d'une population sont mémorisés. Chaque processeur traite la population en exécutant un AG simple tout en étant totalement indépendant des autres processeurs. Une fois la fonction d'évaluation (fitness) est calculée, il donne le résultat au processeur principal pour lui donner l'ordre d'entamer la prochaine génération. Un travail identique est effectué pour tous les autres processeurs.

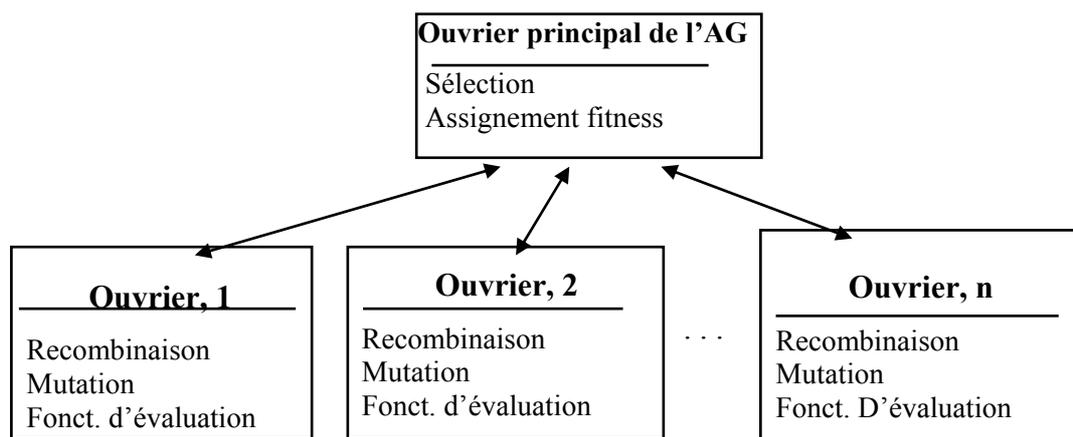


Figure 2.10 : Structure d'un AG global.

Pour effectuer ce travail de groupe, il faut programmer la gestion des tâches pour chaque processeur et veiller à leurs synchronisations. La figure 2.10 montre le fonctionnement général d'une telle méthode.

2.6.2. AG de migration :

C'est un autre type de traitement parallèle. Il divise la population en plusieurs sous populations et chaque sous population est considérée comme une unité de reproduction

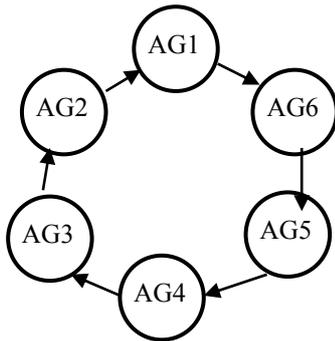


Figure 2.11 : Migration en boucle

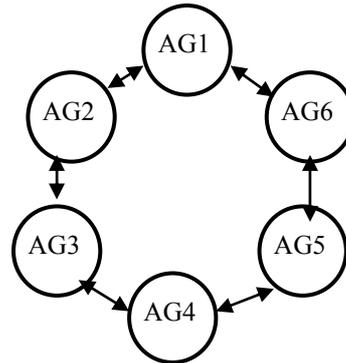


Figure 2.12 : Migration en voisinage

fonctionnant comme des AG conventionnels. Les meilleurs résultats passent d'une sous population d'AG à une autre suivant différentes topologies de migration, comme le montrent les figures 2.11 ,2.12 et 2.13.

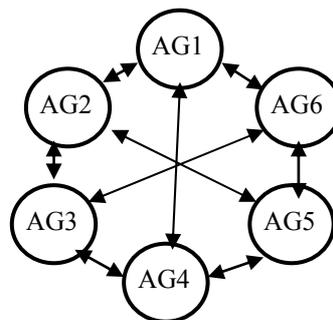


Figure 2.13 : Migration libre

2.6.3. A.G. par diffusion.

Dans cette méthode, on place toutes les sous populations dans un tableau géographique ou matricielle à 2 dimensions et elles fonctionnent simultanément en suivant

une topologie d'un AG globale du type migration en boucle comme le montre la figure 2.14 .

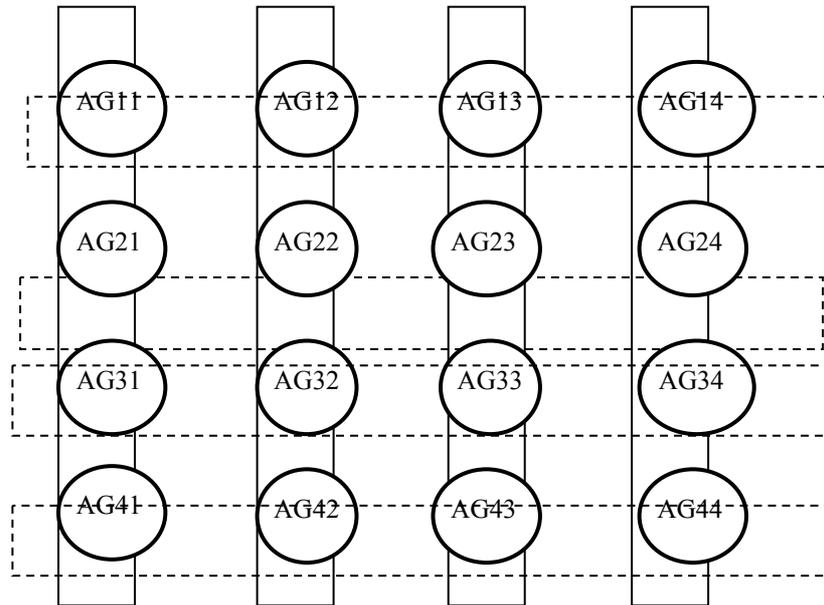


Figure 2.14 : Exemple de topologie de diffusion (4,4)

2.7. Conclusion

Dans ce chapitre on a pu voir un aperçu général sur les concepts des AG. En conclusion, bien que leur conception soit relativement simple, les AG peuvent résoudre des problèmes assez complexes. La résolution de ces derniers est obtenue grâce aux opérateurs génétiques. Les A.G. sont des procédures assez robustes pour résoudre des problèmes d'optimisation [4], néanmoins elles présentent certaines limites et des difficultés. Ces difficultés résident sur le choix des bons paramètres tels que : la taille de la population, le nombre de générations, les probabilités de croisement et de mutation et le choix des méthodes des opérateurs génétiques. Ceux-ci dépendent essentiellement du problème à résoudre et de sa codification.

CHAPITRE 3

OPTIMISATION DES PERFORMANCES D'UN CONTROLEUR FLOU PAR AG

3.1. Introduction.

De nos jours, la logique floue (en anglais « Fuzzy logic ») est un axe de recherche important sur lequel se focalisent de nombreux scientifiques. Des retombées technologiques sont d'ores et déjà disponibles, tant dans le domaine grand public (appareils photos, machines à laver, fours à micro-onde), que dans le domaine industriel (réglages et commandes de processus complexes liés à l'énergie, aux transports, à la transformation de la matière, à la robotique, aux machines-outils) [21].

Le but principal de la logique floue consiste à implémenter un savoir faire humain ou règles heuristiques, en programme informatique que la machine peut exécuter. La logique floue fournit un formalisme mathématique pour réaliser ce but. Actuellement les chercheurs, ingénieurs et tout ceux qui éprouvent un besoin de formaliser des méthodes empiriques pour d'automatiser les prises de décisions dans leurs domaines. Ils éprouvent un besoin de plus en plus énorme pour construire des systèmes basés sur l'intelligence artificiel afin d'effectuer des tâches habituellement prisent par les êtres humains.

Les algorithmes basés sur la logique floue, sont considérés comme des solutions intéressantes pour le réglage des systèmes non linéaires. Pourtant lorsque le système à commander est connu de façon imprécise, la conception du contrôleur flou n'est pas une chose simple. Le concept de la commande floue a été introduit pour la première fois par Mamdani et Assilian en 1975 [32] en réalisant un contrôleur flou sur un moteur à vapeur. A partir de 1985 [33][34] les Japonais commençaient à utiliser la logique floue dans le domaine industriel pour résoudre les problèmes de réglage et de commande. Vers 1990 l'utilisation de la logique floue s'intensifie au Japon avec le contrôle du métro de Sendai.

3.2. Principe de la logique floue. [22].

Le concept de la logique floue vient du fait que la variable binaire booléenne classique ne peut prendre que deux états de vérités vrais ou faux, et qui n'est qu'un cas particulier de la plupart des phénomènes courants, d'où la nécessité d'introduire une nouvelle logique qui prend en charge les phénomènes intermédiaires par des valeurs pouvant prendre des vérités comprises entre 0 et 1.

3.3. Généralités sur les sous ensembles flous [23].

Un sous ensemble flou est un ensemble sans limite stricte. La notion d'appartenance à un sous ensemble est graduelle, c'est-à-dire, un élément peut appartenir plus ou moins fortement à cet ensemble. De là découle l'élément majeur du raisonnement concernant l'environnement flou.

Un sous-ensemble flou A dans un univers du discours U est caractérisé par sa fonction d'appartenance $\mu_A(x)$ qui associe à chaque élément x de U une valeur dans l'intervalle des nombres réels $[0, 1]$.

$$\mu_A : X \rightarrow [0, 1] \quad (3.1)$$

Ainsi un sous-ensemble flou A dans X peut être représenté par un ensemble de couples ordonnés

$$A = \{ (x, \mu_A(x)) / x \in U \}. \quad (3.2)$$

Nous utiliserons parfois dans la suite de ce mémoire la terminologie ensemble flou pour citer un sous-ensemble flou.

3.3.1. Variable linguistique.

Une variable linguistique représente un état dans le système à régler ou une variable réglable dans un contrôleur flou, qui d'après Lotfi Zadeh comme étant des mots ou des phrases qui permettent de décrire dans un cadre très général la connaissance acquise sur une variable, même lorsqu'elle est vague ou imprécise.

Elle est caractérisée par le quintuple $\{ x, T(x), U, M, G \}$ où :

x : le nom de la variable .

T(x) : l'ensemble des valeurs linguistiques que peut prendre x

U : l'univers de discours associé à la valeur.

M : la règle sémantique pour associer un sens à chaque valeur linguistique de x .

G : la règle syntaxique pour générer les valeurs linguistiques de x à partir de $T(x)$

3.3.2. Proposition floue.

Une proposition est obtenue à partir de propositions floues élémentaires en utilisant les formes générales suivantes :

Si [x est A]	alors	y est B
Antécédent		Conséquence
Si [x est A et y est B]	alors	z est C
Si [x est A ou y est B]	alors	z est C

3.3.3. Règles linguistiques.

Les règles floues permettent de déduire des connaissances concernant l'état du système en fonction des qualifications linguistiques. Habituellement, les règles floues sont déduites des expériences acquises par des opérateurs ou des experts. Ces connaissances sont traduites en règles simples pouvant être utilisées dans un processus d'inférence floue.

L'idée principale des systèmes basés sur la logique floue est de traduire la connaissance humaine sous la forme de règles linguistiques de la forme **Si ... Alors**, composées de deux parties et tel que chaque règle est basée sur l'implication floue.

Par exemple, « **Si** la température de l'eau est chaude, **Alors** il faut ajouter de l'eau froide ». Dans le cas où il y a plusieurs règles, la forme générale d'écriture est :

Règle 1	Si proposition flou (1)	Alors proposition flou (2).....	ou ...
Règle i	Si proposition flou (i)	Alors proposition flou (i+1).	...

Aujourd'hui, il est cependant possible de constituer une base de règles floues grâce à des méthodes d'apprentissages automatiques, c'est à dire, sans avoir nécessairement besoin d'un expert humain. Cette stratégie sera décrite plus en détail dans la suite de ce chapitre.

3.3.4. Inférence floue.

Une inférence floue est basée généralement sur l'implication floue de type modus ponens généralisée qui a la forme suivante :

Antécédent :	prémisse 1 :	S1 est A ₁
	Prémisse 2 :	Si S1 est B ₁ alors S2 est B ₂
Conséquence :		S ₂ est A ₂

Où : S_1 et S_2 sont des variables linguistiques et A_1, A_2, B_1, B_2 sont des valeurs linguistiques définies dans l'univers de discours de S_1 et S_2

3.4. Structure générale d'un contrôleur flou.

Lorsqu'un opérateur humain commande manuellement un système, les actions qu'ils réalisent sont dictées par une connaissance subjective du fonctionnement de ce système. Par exemple, s'il fait «froid» dans une pièce, on «augmente» le chauffage ; s'il fait «très froid», on «chauffe plus». Cette commande du système peut être envisagée de façon différente selon la personne qui la réalise : la sensation de «froid» n'est donc pas directement liée à une mesure de la température. Ce principe de raisonnement est à la base de la commande floue. La mesure réalisée sur le système («température») est prise en compte par l'intermédiaire d'une variable linguistique («froid», «tiède», «chaud»), qui est issue d'une analyse par un expert humain. Ensuite, l'action à réaliser est déduite à la fois d'un ensemble de règles de commandes (« S'il fait froid, **Alors** on chauffe plus » . . .) et de l'état du système, qualifié par la variable linguistique. Enfin, la commande finale du système est créée en utilisant les conclusions de la déduction. Pour mettre en pratique cette algèbre, les chercheurs ont mis en évidence des techniques basées essentiellement sur le raisonnement suivant .

3.4.1. Structure d'une inférence floue.

Un système d'inférence floue comporte quatre composants principaux et qui sont représentés dans la figure 3.1.

- L'interface de fuzzification : elle permet de traduire les entrées réelles en des qualifications linguistiques, en utilisant des règles sémantiques définies par un expert.
- Machine d'inférence : elle n'est ni plus ni moins qu'une relation floue définie entre deux sous-ensembles. On applique un raisonnement flou pour ces valeurs afin d'obtenir un résultat flou . On utilise souvent dans le domaine du contrôle , les inférences floues définies par Mamdani et T.Sugeno. [23]
- Une base de connaissance : qui comprend deux parties :
 - * Un ensemble de règles floues (la base des règles).
 - * Un ensemble de fonctions d'appartenance (la base des données)

- Une interface de défuzzification : qui transforme le résultat flou en valeur réelle
La prise de décision est effectuée par la machine d'inférence en utilisant les règles de la base de connaissance qui constituent le noyau de tout système flou.

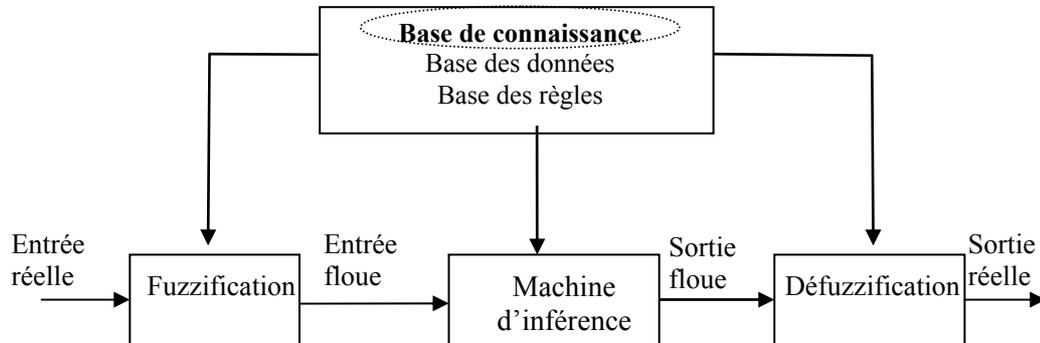


Figure.3.1 : Structure générale d'un système d'inférence floue

3.4.2. Structure d'un contrôleur flou .

Sa structure repose essentiellement sur le système d'inférence floue auquel on lui ajoute en chaîne directe le processus, et une chaîne de retour unitaire [24] comme le montre le schéma de la figure 3.2.

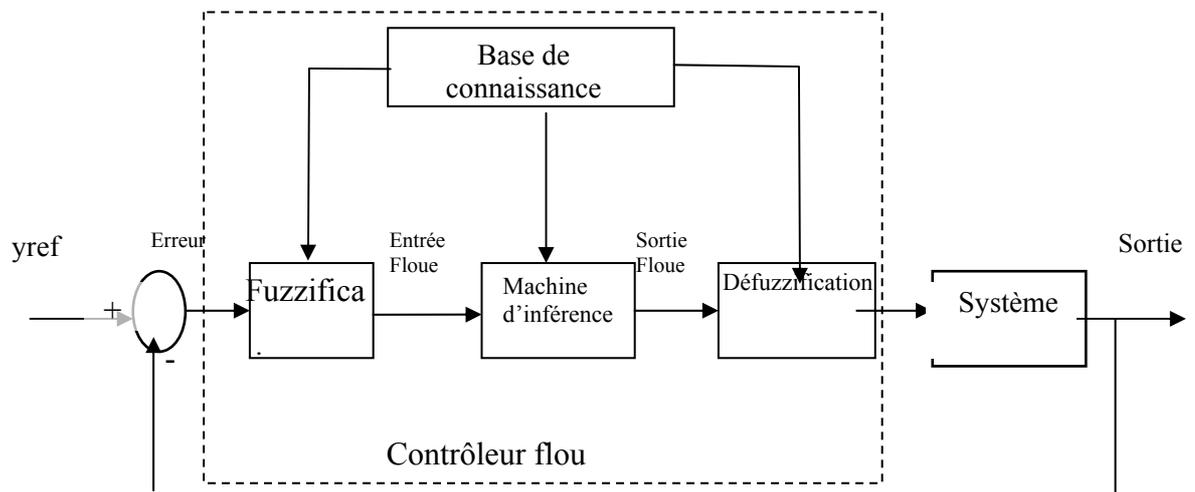


Figure 3.2 : Structure générale d'un contrôleur flou en chaîne directe

De là, on peut déduire plusieurs types de contrôleurs flous, dont les plus utilisés sont :

- Contrôleur flou du type proportionnel où l'entrée est une erreur.
- Contrôleur flou du type proportionnel dérivée où dans ce cas il possède deux entrées : l'erreur et sa dérivée.

Afin de pouvoir obtenir le comportement désiré d'un contrôleur flou, il faut identifier les paramètres importants qui influent sur le contrôleur. On distingue généralement deux méthodes principales :

- La méthode directe qui requiert la présence d'un expert humain en la matière.
- La méthode automatique est basée sur les méthodes de l'intelligence artificielle tels que les réseaux de neurones, la logique floue, les algorithmes génétiques ou les deux à la fois.

La méthode automatique est généralement utilisée lorsque la connaissance du problème est incomplète, ou que le problème est complexe [31].

L'efficacité d'un contrôleur flou dépend de plusieurs paramètres, la sélection de bonnes variables, leurs univers de discours et les règles floues correspondantes. Le tableau 3.1 résume les principaux paramètres à ajuster afin d'améliorer les performances d'un contrôleur flou.

Classe	Paramètres
<i>Logique</i>	Mécanisme de raisonnement Opération floue Type de fonction d'appartenance Méthode de défuzzification
<i>Structure</i>	Variables importantes Nombre de fonctions d'appartenance Nombre de règles
<i>Connexion</i>	Antécédent des règles Conséquence des règles poids des règles
<i>Opérationnel</i>	Valeur des fonctions d'appartenance Type de fonction d'appartenance

Tableau 3.1 : Classification des paramètres d'un système d'inférence floue

3.5. Application de l'AG au contrôleur flou

Le présent travail consiste à utiliser un AG pour optimiser les performances d'un contrôleur flou de manière automatique [25]. La synthèse de celui-ci est réalisée sous forme d'une table de décision [35] qui délivrera une commande nécessaire au C.S.T.R. afin d'obtenir la concentration désirée avec le minimum d'erreur possible, car le système est irréversible.

Dans un contrôleur flou les paramètres les plus influençables, sont les fonctions d'appartenance des variables d'entrée et de sortie, et l'ensemble des règles floues [26]. Par conséquent l'objectif de l'AG est d'optimiser ces paramètres.

Dans le premier cas, un contrôleur flou du type proportionnel est adopté, ayant deux entrées : l'erreur de la température et l'erreur de la concentration et comme sortie la commande du débit du liquide réfrigérant. Afin que ce contrôleur flou puisse fonctionner sans expert, on adoptera la méthode automatique qui fait appel aux algorithmes génétiques [27] et qui sera détaillée dans les schémas d'asservissement des figures 3.3 et 3.4 où :

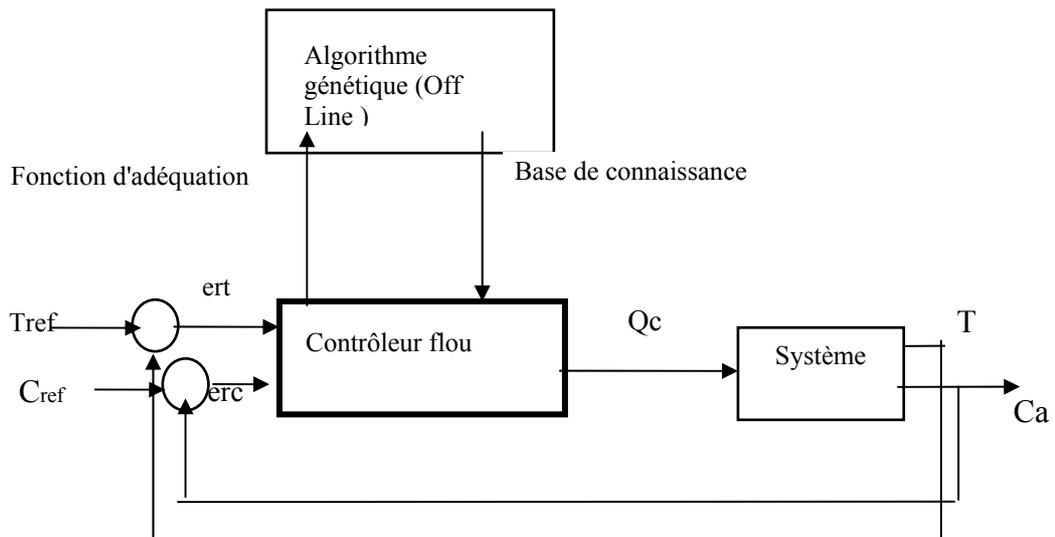


Figure 3.3 : Structure d'un contrôleur proportionnel flou à deux entrées piloté par un AG

C_{ref} , T_{ref} : Concentration et température désirées

$erc = C_{ref} - C_a$: Erreur de concentration .

$ert = T_{ref} - T$: Erreur de température .

$derc$: Dérivé de l'erreur de concentration

Q_c : Débit du réfrigérant.

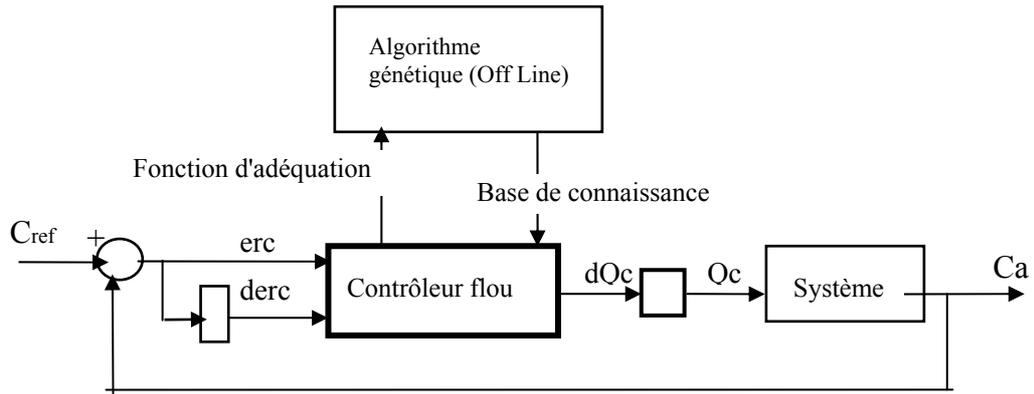


Figure 3.4 : Structure d'un contrôleur flou du type PD piloté par un AG

Dans le deuxième cas on utilisera un contrôleur du type proportionnel dérivé (PD) qui prend en considération l'erreur erc et sa dérivée $derc$ comme le montre la figure 3.4. Dans ce cas on aura une seule boucle de retour, et une seule consigne C_{ref} .

Pour les deux cas cités précédemment, on leur appliquera deux types d'algorithmes génétiques. L' A.G.S. sera appliqué dans le cas où on tiendra compte de toutes les fonctions d'appartenance et les règles possibles. Quant à l' HGA, il sera utilisé pour réduire la complexité de la base de connaissance de l'inférence floue tout en optimisant les règles et les fonctions d'appartenance. Cette procédure consiste aussi à choisir automatiquement les règles et le nombre de règles et le nombre de fonctions d'appartenance.

3.5.1. Optimisation de la base de connaissance à l'aide d'un A.G.S.

Cette optimisation repose sur plusieurs paramètres à prendre en considération (voir tableau 3.1) et dont les plus importants sont : la base des données et les règles d'inférence [28] [29]. De même, il faut tenir compte aussi du choix de codage du chromosome, de la fonction d'adéquation et des opérateurs génétiques

3.5.1.1. La base de donnée.

Dans le premier cas, les fonctions d'appartenance choisies sont illustrées par les figures 3.5. a,b,c. On a utilisé pour les trois variables d'entrées et de sortie du contrôleur flou proportionnel, cinq fonctions d'appartenance (deux trapézoïdales et trois triangulaires) et des univers de discours répartis comme suit :

* Pour l'erreur de concentration $erc \in [-.06 \ .06]$ ayant les ensembles flou suivants :

NG(Négative Grand), **NP**(Négative Petit), **Z** (zéro), **PP** (Positive Petit) et **PG** (Positive Grand).

* Pour l'erreur de la température $ert \in [-12 \ 12]$ et ayant comme ensemble flou [**NG NP Z PP PG**].

* Le débit $Qc \in [100 \ 140]$ et ayant comme ensemble flou [**NG NP Z PP PG**]

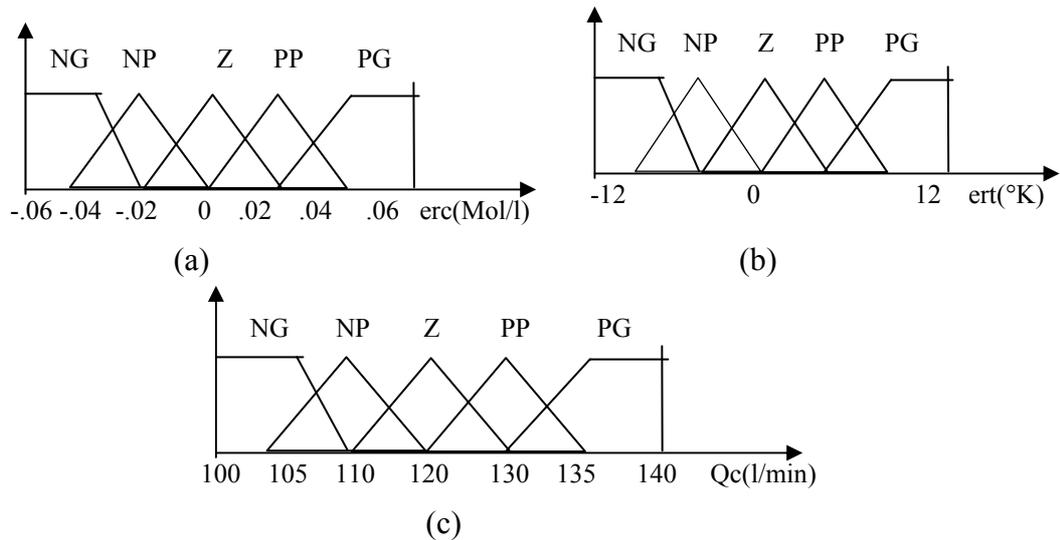


Figure 3.5 : Répartition des fonctions d'appartenance de
 a : l'erreur de concentration
 b : l'erreur de température
 c : du débit réfrigérant

3.5.1.2. Règles d'inférence.

Si N représente le nombre de fonctions d'appartenance des entrées et M le nombre de fonctions d'appartenance de sortie, alors le nombre maximum de règles à manipuler est $N \cdot M$, et comme $N=5$ et $M=5$, alors on a 25 règles possibles. Généralement dans le domaine de l'automatique on utilise la formulation de Mamdani :

If [proposition] **Then** [proposition]

qui se traduit dans notre cas par :

If "erc" is {NG, NP, Z, PP, PG} **and** "ert" is {NG, NP, Z, PP, PG} **Then** Qc is {NG, NP, Z, PP, PG}

La combinaison des règles permettra d'établir le tableau 3.2 de décision et qui permet de déduire les valeurs linguistiques de la commande Qc .

erc ert	NG	NP	Z	PP	PG
NG	.	NG		.	.
NP	NG	NP	Z	PG	PP
Z	.	.	Z	.	.
PP	.	Z	.	PP	.
PG	.	.		.	NP

Tableau 3.2 : Tableau de décision de la sortie Qc.

Pour augmenter l'espace de recherche, l'A.G. choisit les écartements des fonctions d'appartenance des variables d'entrée. Dans le présent cas on a opté pour cinq écartements possibles que l'on peut voir sur la figure 3.6

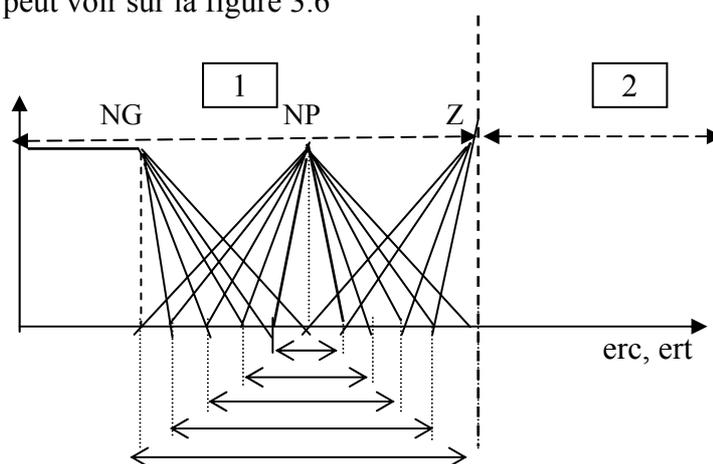


Figure 3.6 : Schéma des pics et écartement des triangles et trapèzes

Les parties 1 et 2 sont symétriques par rapport à l'axe Z

3.5.1.3. Codage du chromosome.

D'après, les résultats précédents, on peut déduire que la longueur de la chaîne du chromosome est de 35 éléments réels repartis comme suit :

* Les 25 premiers éléments représentent les conséquences des règles floues codées en base 6. De cette manière on adoptera la correspondance : {NG=1, NP=2, Z=3, PP=4, PG=5}.

* Les 10 derniers éléments représentent les différentes écartements possibles des pics des fonctions d'appartenance .Si on prend le cas où ces fonctions sont symétriques par rapport à leur pic, on réduit la longueur des 10 derniers éléments du chromosomes à 6, et

de cette manière on passe à une longueur totale du chromosome de 31 éléments. La figure 3.7 illustre les détails du codage du chromosome.

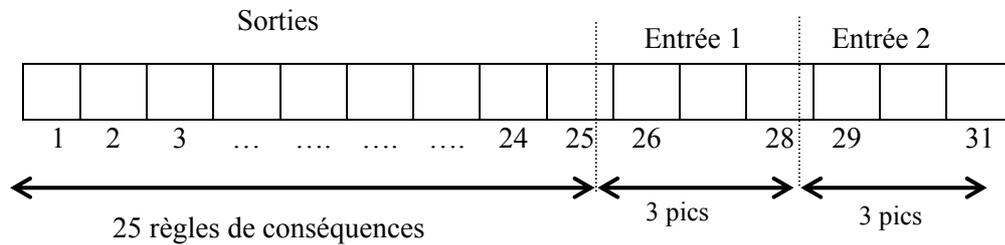


Figure 3.7 : Configuration du chromosome du type AGS.

3.5.1.4. Choix de la fonction d'adéquation.

Pour le choix de la fonction objective, on a opté pour la minimisation de l'erreur quadratique dans le but de diminuer les erreurs au cours de la phase transitoire.

Dans le cas où on a deux variables à contrôler fonction objective tient compte de :

$$\text{Fit1} = \sum_{i=1}^n (\text{ert}(i)^2 + \text{erc}(i)^2). \quad (3.3)$$

Dans le cas où on a une variable à contrôler, la fonction objective tient compte de :

$$\text{Fit2} = \sum_{i=1}^n \text{erc}(i)^2 \quad (3.4)$$

Où, erc et ert représentent respectivement les erreurs de concentration et de température

3.5.1.5. Choix des opérateurs génétiques.

- Pour la reproduction : on a utilisé la wheel-sélection, afin d'éviter une convergence prématurée, ainsi les meilleurs individus ayant une bonne valeur d'adéquation peuvent être copiés plusieurs fois dans la génération future.
- Pour le croisement : on a utilisé le cas d'un point de croisement aléatoire avec une probabilité P_c égale à 0.8.
- Pour la mutation : elle est du type réelle et constante tel que P_m égale à 0.03.

3.5.2. Optimisation du contrôleur flou par un H.G.A.

Dans ce cas deux stratégies différentes sont étudiées, la première consiste à optimiser les règles et le nombre de règles valides possibles, quant à la deuxième, on verra

comment le HGA optimise un cas plus complexe à savoir l'optimisation et la minimisation du nombre de règles et le nombre de fonctions d'appartenance actives [30].

3.5.2.1. Stratégie d'optimisation des règles floues (H.G.A1)

Dans cette stratégie, pour chaque variable d'entrée et de sortie du contrôleur flou, on a choisi les mêmes types de fonctions d'appartenance que celui d'un AGS. (Voir figure 3.5) et on a opté pour une seule boucle de retour (retour de la concentration). Le choix des caractéristiques de l'H.G.A1 sera comme suit :

a) Codage du chromosome : si on utilise 5 fonctions d'appartenance pour l'erreur **erc** de concentration et pour sa dérivée **derc**, alors on aura ,d'après le tableau 3.3 de décision, au maximum 25 règles à manipuler donc une chaîne de 25 éléments . Pour les gènes de contrôle on utilisera 10 éléments binaires, tel que les 5 premiers seront réservés pour activer les règles de l'erreur de concentration **erc** et les 5 autres pour l'activation de sa dérivée **derc** . Quant aux gènes de structure, ils sont codés en base 6 de type entier et ayant pour valeur [1 ou 2 ou 3 ou 4 ou 5] et avec la correspondance suivante { 1=NG, 2 =NP, 3= Z, 4= PP et 5=PG }.

	erc	NG	NP	Z	PP	PG
derc						
NG						Z
NP		NG	NP	Z	PG	PP
Z				Z		
PP			Z			
PG						NP

Tableau 3.3 : Exemple de table de décision de la dérivée de sortie dQc

Ainsi le chromosome aura la représentation de la figure 3.8.

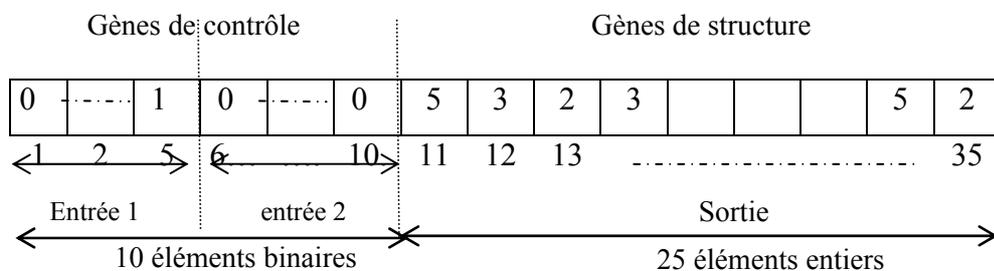


Figure 3.8 : Méthode de codage du chromosome du type HGA1

b) Cycle génétique : l'organigramme de la figure 3.9 explique le fonctionnement général de l'HGA1 lorsqu'on effectue l'optimisation des règles floues seulement.

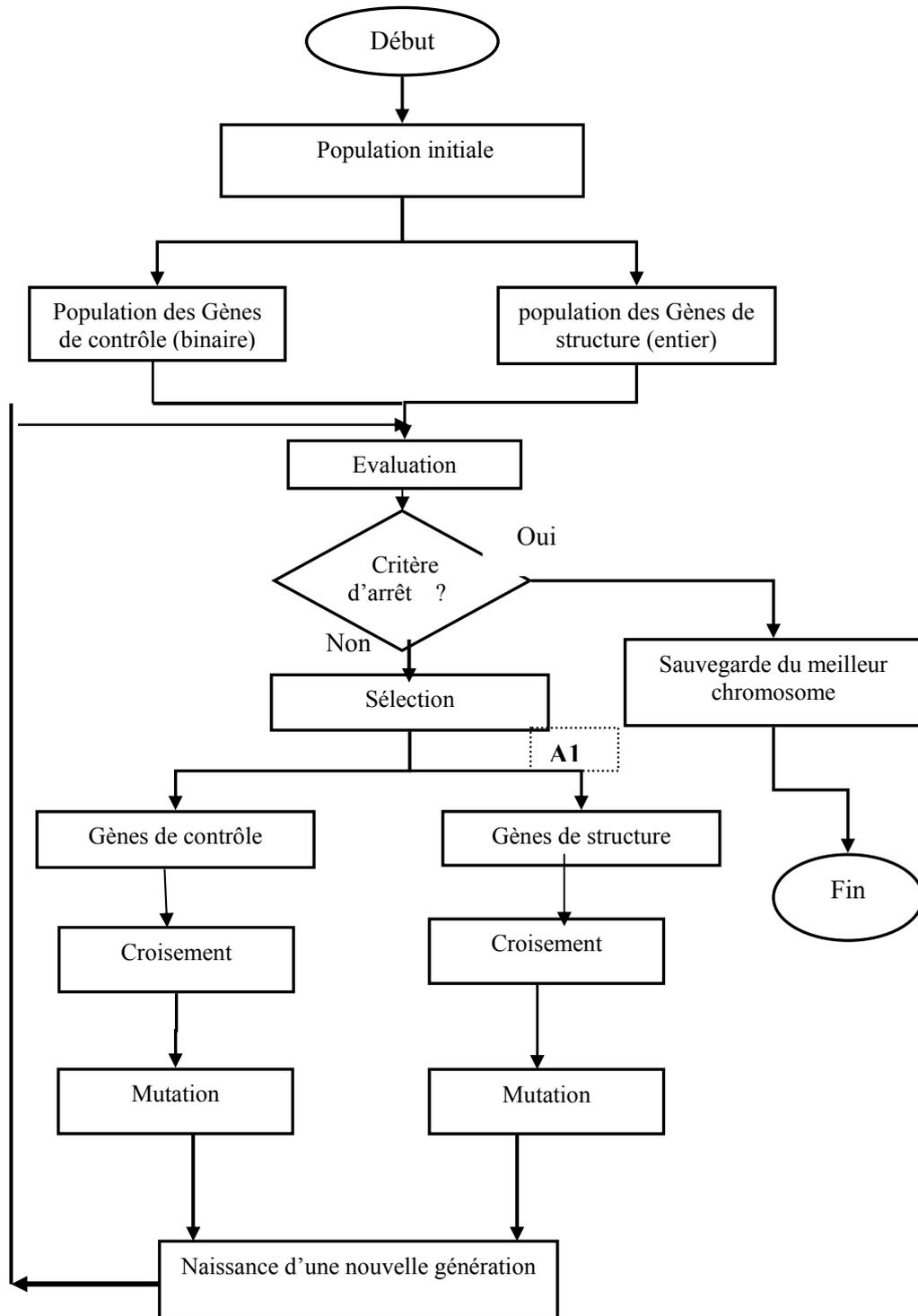


Figure 3.9 : Organigramme du cycle génétique du type HGA1

c) Opérations génétiques : on utilisera les mêmes opérateurs que dans l'AGS mais avec des probabilités au niveau des gènes de contrôle légèrement différentes que celles des gènes de structure soient :

$$\left\{ \begin{array}{l} P_{cc} = 0.4 : \text{probabilité de croisement des gènes de contrôle.} \\ P_{cs} = 0.8 : \text{probabilité de croisement des gènes de structure.} \end{array} \right.$$

$$\left\{ \begin{array}{l} P_{mc} = 0.02 : \text{probabilité de mutation des gènes de contrôle.} \\ P_{ms} = 0.04 : \text{probabilité de mutation des gènes de structure.} \end{array} \right.$$

d) Fonction d'adaptation : appelée aussi fonction d'évaluation, elle est du type linéaire et pour transformer le problème de minimisation en un problème de maximisation En utilisant 3.4, on a choisi la fonction d'adaptation de la forme :

$$Fe = \frac{a}{b + \frac{1}{n} Fit2}$$

Où : a, b : coefficient de pondération

Fit2 : erreur quadratique de la concentration.

3.5.2.2. Stratégie d'optimisation des règles et des fonctions d'appartenance (HGA2)

Dans la plupart des cas, la répartition des fonctions d'appartenance et des règles d'inférence sont inconnues ou elles sont déterminées par un expert. Ainsi l'activation ou la désactivation des fonctions d'appartenances au niveau des entrées ou sorties du contrôleur flou , influe largement sur les performances de contrôle du processus [11], et la question qui se pose est alors : Est-ce que ce type d'HGA peut résoudre ce genre de problème ?. Pour ce faire on va développer la stratégie comme suit :

Comme dans le cas précédent, on va choisir les mêmes types de fonctions d'appartenance, mais le nombre reste à déterminer ultérieurement par l'algorithme génétique.

La structure du chromosome est composée de deux parties importantes, qui sont les gènes de contrôle (G.C.) et les gènes de structure (G.S.) .Ces derniers sont eux mêmes divisés en deux parties, à savoir la partie de dimensionnement des fonctions

d'appartenances notée GS1 et la partie du choix des règles d'inférence floues notée GS2.

a) codage du chromosome : Dans cette technique on va commencer par utiliser 7 fonctions d'appartenance pour chaque une des variables d'entrées et sortie (*erc* , *derc* et *dQc*) et comme variables linguistiques [NG,NM,NP,Z,PP,PM,PG].

De cette manière on aura 49 (7*7) règles à manipuler et qu'on va les prendre comme des entiers codés en base 8 ,qui vont constituer une partie des gènes de structure du chromosome. Les règles auront la forme :

$$\left\{ \begin{array}{l} \text{If } \mathit{erc} \text{ is } \{NG, NM, NP, Z, PP, PM, PG\} \\ \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \text{Then } \mathit{dQc} \text{ is } \{NG, NM, NP, Z, PP, PM, PG\} \\ \text{and } \mathit{derc} \text{ is } \{NG, NM, NP, Z, PP, PM, PG\} \end{array} \right.$$

Comme les fonctions d'appartenance sont du type triangulaire, alors comme le montre la figure 3.10, chaque fonction d'appartenance a besoin de 3 valeurs pour la représenter. Ainsi pour chaque variable (entrée ou sortie), on doit leur attribuer 21 (3*7) valeurs différentes

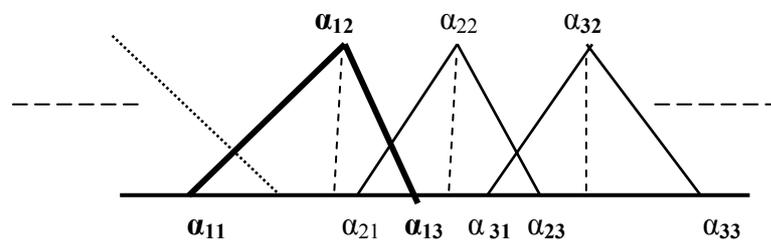


Figure 3.10 : Codage des fonctions d'appartenance

qui vont correspondre à 21 éléments réels : soit un total de 63 éléments réels qui représentent les gènes de structure GS1 et qui vont être compris dans un univers de discours approprié.

Les gènes de contrôle vont être utilisés pour l'activation et la non activation des fonctions d'appartenance. Pour cela, un codage binaire est choisi. Pour la présente application on a besoin 21 (3*7) bits, tel que les 14 premiers sont réservés pour les entrées et les 7 autres qui restent pour la sortie.

Ainsi le chromosome aura une taille totale de 133 éléments repartis comme le montre la figure 3.11.

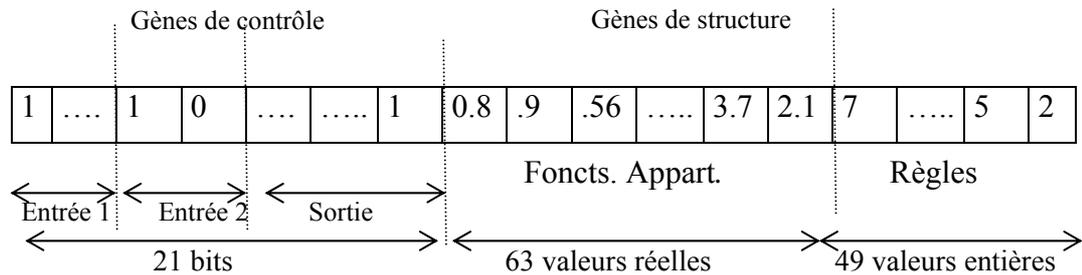


Figure 3.11 : Codage et configuration du chromosome du type HGA2

b) Opérations génétiques : on utilisera les mêmes opérateurs que l'HGA1, mais toujours avec des probabilités des gènes de contrôle différentes de celles des gènes de structure, à savoir :

$$\left\{ \begin{array}{l} P_{cc} = 0.5 : \text{probabilité de croisement des gènes de contrôle.} \\ P_{cs} = 0.8 : \text{probabilité de croisement des gènes de structure.} \\ P_{mc} = 0.02 : \text{probabilité de mutation des gènes de contrôle.} \\ P_{ms1} = 0.04 : \text{probabilité de mutation des GS1.} \\ P_{ms2} = 0.04 : \text{probabilité de mutation des GS2.} \end{array} \right.$$

c) Cycle génétique : l'organigramme du HGA2 sera presque le même que l'organigramme de la figure 3.9 sauf qu'il y a une différence au niveau de l'arborescence A1 à savoir que les gènes de structure sont divisés en deux parties. La première partie est réservée au codage des fonctions d'appartenance, quant à la deuxième partie elle est réservée au codage des règles floues. La figure 3.12 explique le fonctionnement général du cycle génétique de l'HGA2, lorsqu'on effectue simultanément l'optimisation des règles floues, et des fonctions d'appartenance.

d) Fonction d'adaptation : La fonction est choisie du type linéaire de la même forme que l'HGA1. Lorsqu'il y a désactivation de certaines fonctions d'appartenance (voir figure 3.13), on remarque que l'univers de discours des variables d'entrées et de sorties présentent des zones mortes, appelées aussi régions indéfinies, c'est à dire des zones où il n'y a pas de degrés d'appartenance dans une partie de l'univers de discours. Ceci peut engendrer une diminution des performances du contrôleur flou. Pour parer à cet inconvénient il faut associer un sous programme de recouvrement qui permet d'éliminer

ces régions indéfinies et d'arriver à une nouvelle répartition des fonctions d'appartenance (voir figure 3.14).

Cette méthode consiste à recouvrir la fonction inactive par les fonctions d'appartenance avoisinantes. Elle se fait par l'augmentation de la valeur linguistique de la fonction d'appartenance avoisinante de manière à éliminer la zone morte. Par exemple pour recouvrir la fonction d'appartenance F2, on a remplacé α_{1c} par α_{2c} et α_{3a} par α_{2a}

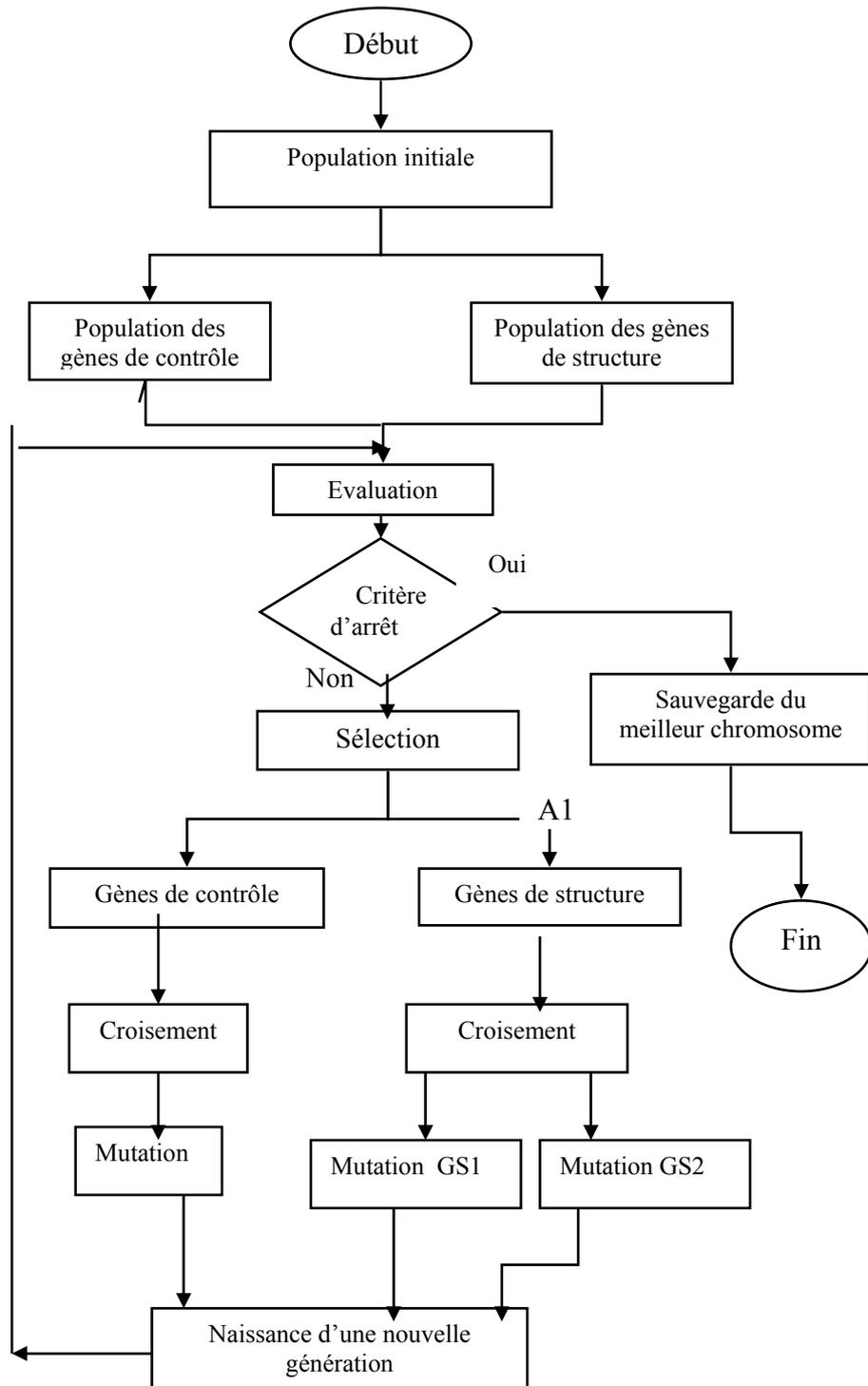


Figure 3.12 : Organigramme du cycle génétique du type HGA2

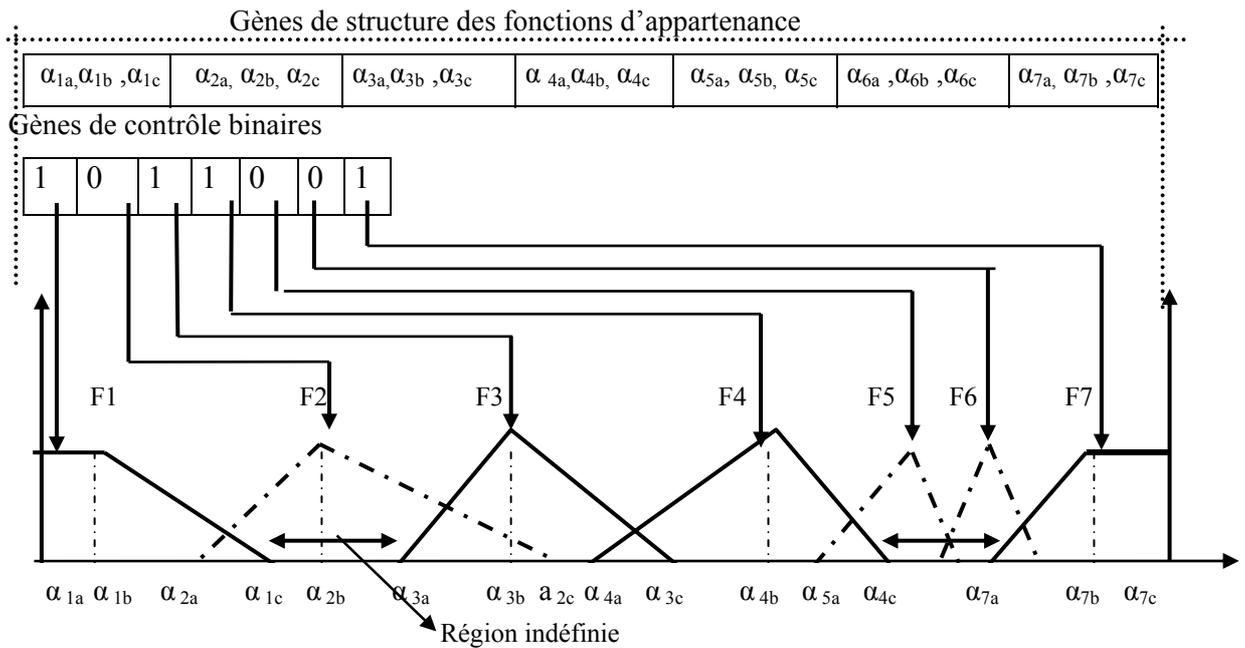


Figure 3.13 : Cas où l'univers de discours présente des régions indéfinies

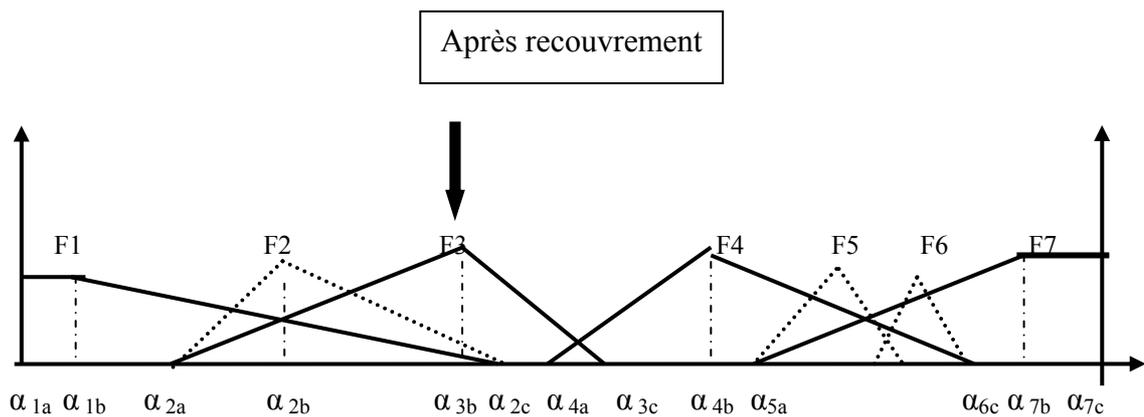


Figure 3.14 : Méthode de recouvrement des fonctions inactives F2, F5 et F6

3.6. Conclusion

Dans ce chapitre, plusieurs méthodes et techniques ont été présentées et qui permettent d'optimiser la base de connaissance d'un contrôleur flou à l'aide des algorithmes génétiques. Ces techniques de représentation peuvent faire intervenir trois types de codage, à savoir un codage binaire, un codage en base N, et un codage en nombres réels et cela pour plusieurs types d'AG. Pour démontrer l'efficacité des méthodes détaillées précédemment, on va les appliquer au contrôle de la température du réacteur chimique du type CSTR présenté au premier chapitre.

CHAPITRE 4

SIMULATIONS ET VALIDATIONS DES RESULTATS

4.1. Introduction

Les contrôleurs flous fournissent des solutions raisonnables et efficaces dans les problèmes de contrôle de processus complexes non linéaires et cela en générant les actions qui conviennent en utilisant d'une part un nombre suffisant de règles linguistiques de la formes Si (condition) **alors** (conséquence) et d'autre part, un choix judicieux, des fonctions d'appartenance et d'univers de discours pour toutes les variables utilisées. Les algorithmes génétiques sont là pour donner une alternative afin de fournir la meilleure base de connaissance possible et cela sans avoir recours à un expert dans le domaine.

4.2. Objectif

L'objectif de la présente étude est l'optimisation d'un contrôleur flou par un AG, qui permet de générer les commandes nécessaires qui minimisent l'erreur commise entre la concentration de référence et la concentration fournie par le réacteur chimique C.S.T.R. Dans cette application particulière, les différentes techniques d'optimisations de contrôle développées dans les chapitres précédents seront mises en œuvre. Plusieurs types de configurations du chromosome seront exposés à savoir :

- un algorithme génétique standard qui permet de régler les écartements des fonctions d'appartenance et le choix des règles floues.
- Un algorithme génétique standard qui optimise les règles floues.
- Un algorithme génétique hiérarchique qui optimise les règles et le nombre de règles.
- Un algorithme génétique hiérarchique qui optimise simultanément les règles et les fonctions d'appartenance.

Dans la suite des résultats de simulation seront présentées dans le cas où le contrôleur hybride dispose de deux entrées (l'erreur de température du réactif et l'erreur de concentration) et comme sortie le débit du réfrigérant, et aussi le cas où il a comme entrée l'erreur de concentration et sa dérivée comme sortie, la dérivée du débit.

Pour tester la robustesse du contrôleur on appliquera des perturbations de température T_c au niveau du réfrigérant.

4.3. Simulations et résultats

On commence les simulations dans le cas où on gère le contrôleur flou par un algorithme génétique standard AGS, puis le cas où on le gère par un HGA.

4.3.1. Résultats de la première stratégie avec deux boucles de retour

Dans ce cas, les schémas d'asservissement des figures 3.3 et 3.4 seront utilisés avec l'application de la technique du réglage des écartements des fonctions d'appartenance expliquée aux paragraphes 3.5.1.2 et 3.5.1.3. Les paramètres globaux de configuration du chromosome sont choisis comme suit :

Population : 40 chromosomes

Longueur du chromosome : 31 éléments

Type de sélection : sélection par roulette.

Type de croisement : aléatoire, en un point

Type de mutation : réelle aléatoire

Probabilité de croisement : 0.8.

Probabilité de mutation : 0.03.

Fonction d'adéquation : $f_a = a.f_{01} + b.f_{02} + c.f_{03} + d.err1$ avec :

$$err1 = 10 / (0.005 + (1/n) \cdot \sum_{i=1}^n erc(i)^2)$$

$$erc = |C_{ref} - C_a|$$

erc : Erreur de concentration.

n : nombre d'échantillons des mesures.

f_{01} , f_{02} , f_{03} : paramètres qui augmentent lorsque erc est à l'intérieur d'un intervalle d'erreur δec égale à 10% ou 5% ou 2% de C_{ref} .

a , b , c , d : coefficients de pondération respectivement égales à 0.1, 0.5, 1 et 3. Ils sont fixés avant le processus d'optimisation de façon à orienter la recherche selon une poursuite désirée.

4.3.1.1. Résultats et simulations sans bruit de température dans le réfrigérant

Les résultats de la figure 4.2 montre que le contrôleur hybride flou a bien convergé vers la concentration nominale $C_{ref} = 0.1 \text{ mol/l}$, mais avec une erreur statique inférieure à 1%, ce qui est tout a fait acceptable par le cahier de charge. De même, l'AG a convergé dès la première génération car on a donné à l'AGS et au contrôleur flou suffisamment de données (température de référence et symétrie du choix des fonctions d'appartenance) qui correspondent au régime de fonctionnement permanent du réacteur chimique.

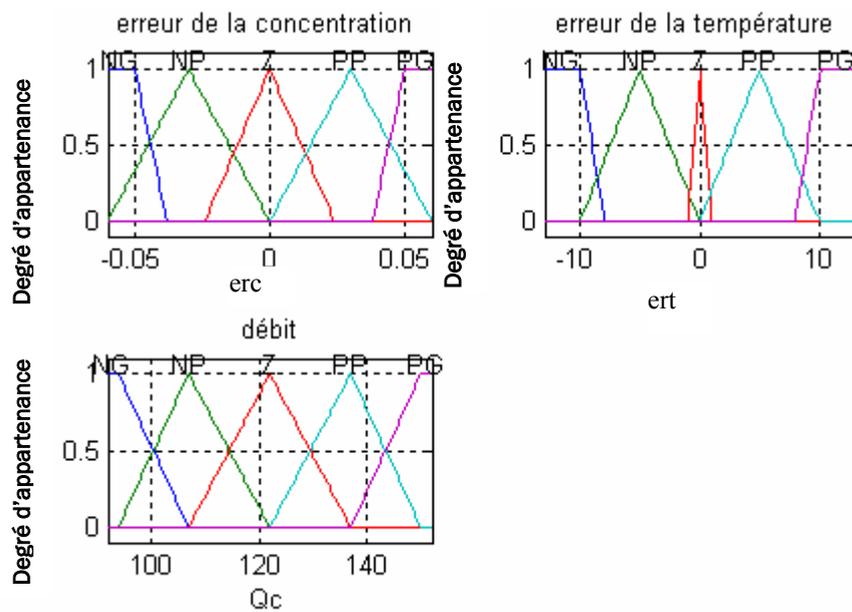


Figure 4.1 : Choix respective des fonctions d'appartenance de erc , ert et Q_c par l'AGS

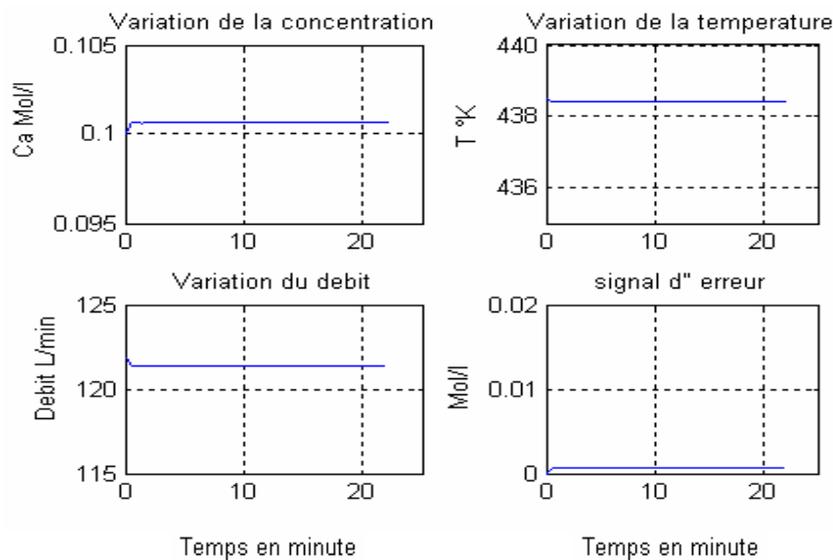


Figure 4.2 : Evolution des signaux de sorties de Ca , T , débit et erc

Les signaux de la figure 4.3 présentent plusieurs types de courbes, telles que :

- La courbe de qualité du meilleur chromosome, qui caractérise l'évolution de la fonction d'évaluation f_e du meilleur chromosome dans la population à travers les générations.

- La courbe de qualité de la génération est donnée par :

$$\text{Ind} = (1/\text{pop}) \cdot \sum_{i=1}^{\text{pop}} f_e(i)$$

Elle caractérise l'évolution de la qualité de la population à travers les générations.

- La courbe de mesure de la diversité des phénotypes (M.D.P.) est :

$$\text{MDP} = F_{\text{best}}/\text{moy}(f_e)$$

F_{best} : est la valeur de la meilleur f_e de la population

$\text{moy}(f_e)$: moyenne des f_e de la population au cours d'une génération

Ce paramètre nous renseigne sur le degré de diversification de la population à travers les générations.

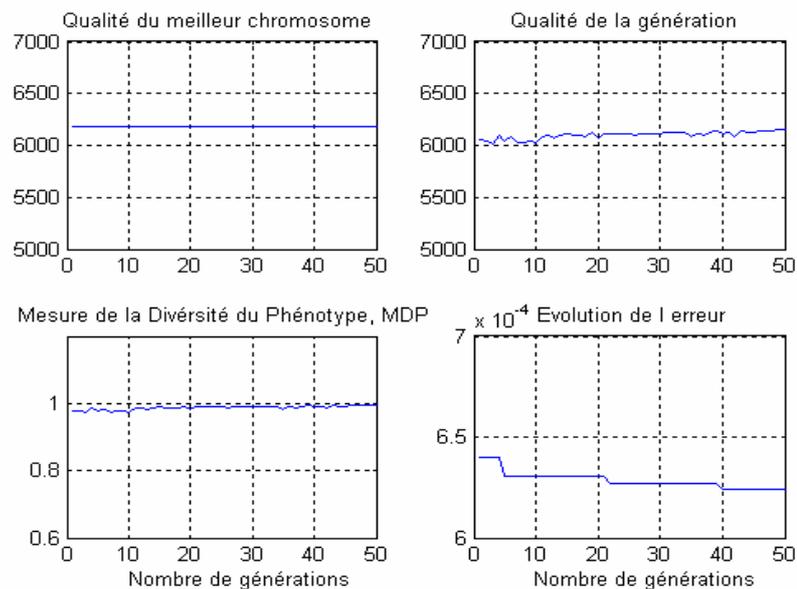


Figure 4.3 : Courbe de suivi de la qualité des générations

erc ert	NG	NP	Z	PP	PG
NG	NG	Z	Z	PP	NP
NP	Z	PG	PG	NP	NG
Z	NP	PG	Z	NG	PP
PP	Z	NG	NP	PP	Z
PG	PG	NP	PP	Z	PP

Tableau. 4.1 : Conclusion des règles floues déduites par l'AGS

4.3.1.2. Résultats de simulation en présence de bruit de température dans T_c .

Dans ce cas, un bruit blanc de température est introduit (figure 4.4) au niveau du réfrigérant T_c , qui peut correspondre à des erreurs de mesures ou d'isolement au niveau des conduits du réfrigérant ou bien du système de refroidissement, dans le but de voir la robustesse du contrôleur vis à vis de la température.

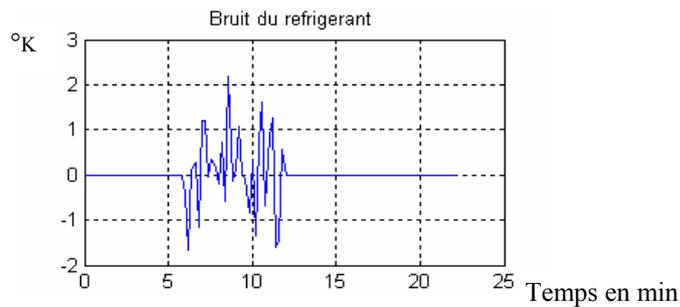


Figure.4.4 : Nature du bruit de température dans T_c

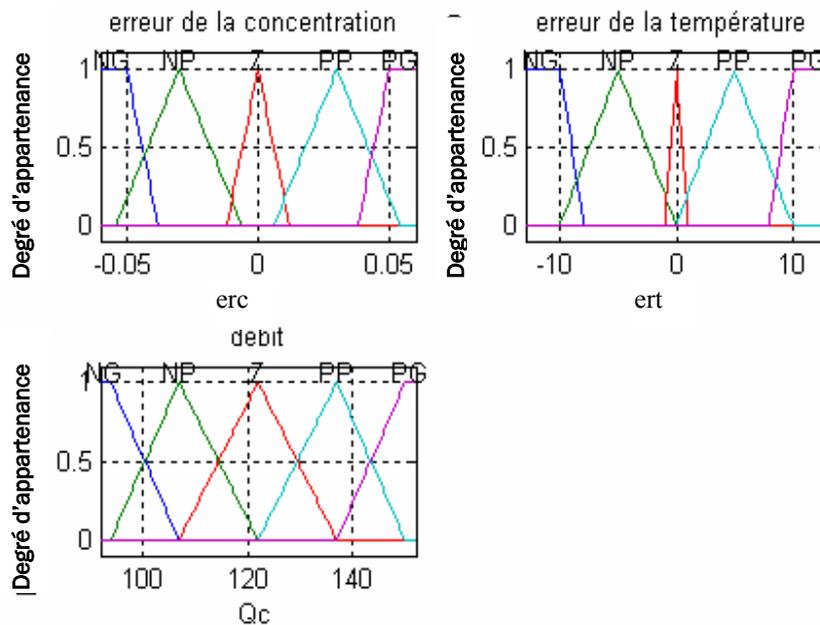


Figure 4.5 : Choix respectif des fonctions d'appartenance de erc , ert et Q_c par l'AGS

Les résultats de la figure 4.6 montrent que le contrôleur est robuste vis à vis d'un bruit important de température du réfrigérant ($-1.6 < \Delta T_c < 2.3$) (voir figure 4.4), car lors de la conceptions du réacteur chimique on attache une grande importance au système de refroidissement du réacteur. D'après la figure 4.6 la concentration C_a présente un dépassement inférieur à 6 % tout au long de la période de perturbation (acceptable par le cahier de charge), puis elle revient à sa valeur nominale C_{ref} lorsque la perturbation disparaît. D'après la figure 4.7, on remarque que la convergence commence à avoir lieu

dés les premières générations avec un M.D.P très proche de 1. On remarque aussi que la fonction d'évaluation fe du meilleur chromosome se stabilise très vite.

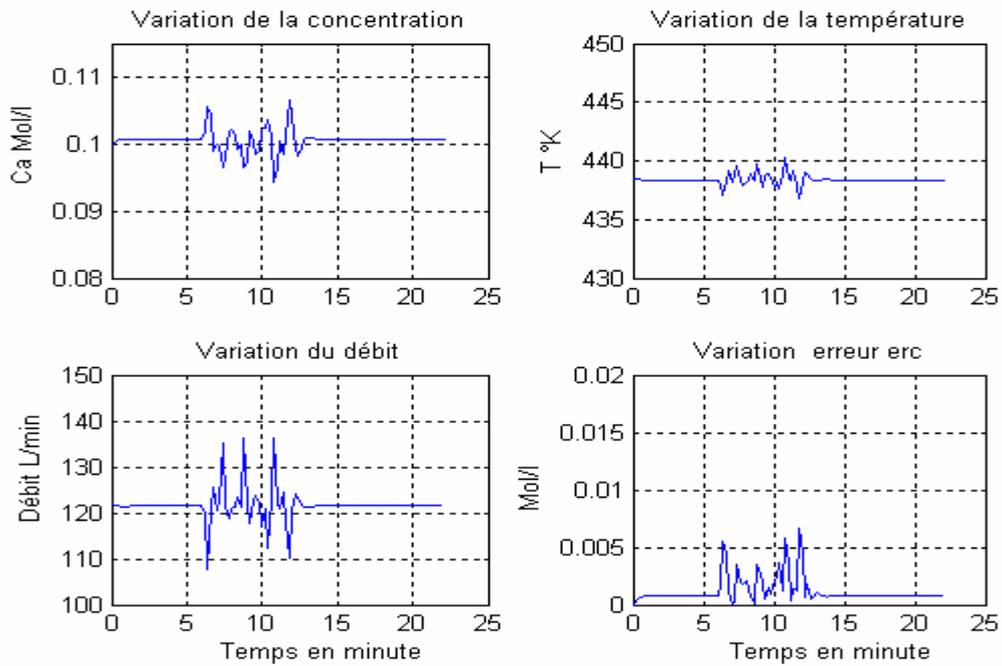


Figure 4.6 : Evolution des signaux de sorties de Ca, T, débit et erc

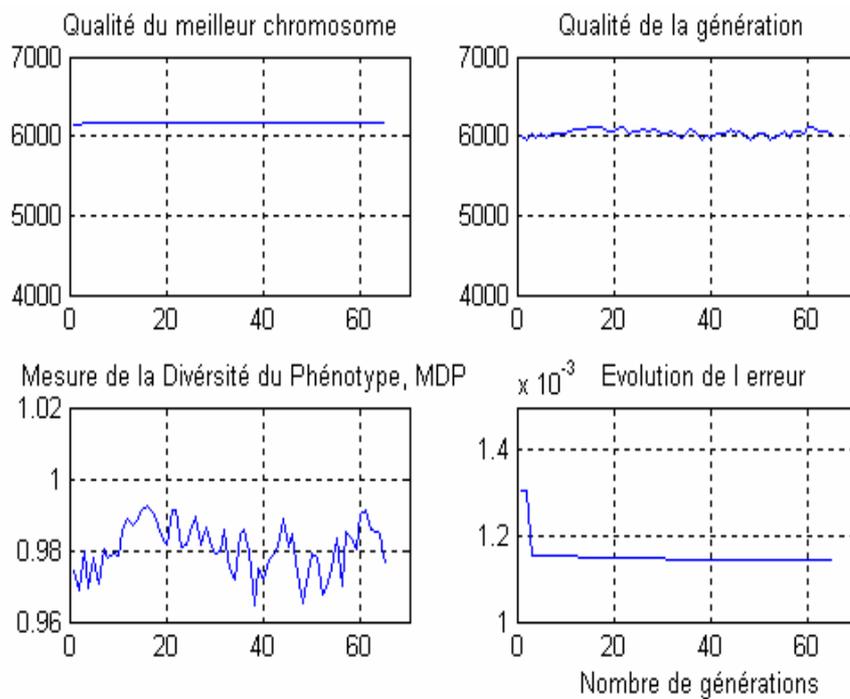


Figure 4.7 : Courbes de suivi de la qualité des générations

4.3.1.3 Résultats de simulation avec un signal de référence carré.

a) Sans bruit de température dans le réfrigérant T_c :

Dans cette simulation le schéma d'asservissement de la figure 3.4 est utilisé où on doit introduire deux paramètres de référence qui sont la concentration C_{ref} et la température de référence T_{ref} du réactif. D'après les résultats de la figure 4.9, on remarque que le contrôleur génère la commande du débit nécessaire pour que le réacteur suit la consigne d'une manière convenable. Néanmoins il y a une légère erreur statique inférieure à 2% avec C_{ref} . C'est un dépassement que l'on peut considérer très acceptable au niveau de la phase transitoire. La figure 4.10 montre que l'AGS converge rapidement dès la 15^{ème} génération.

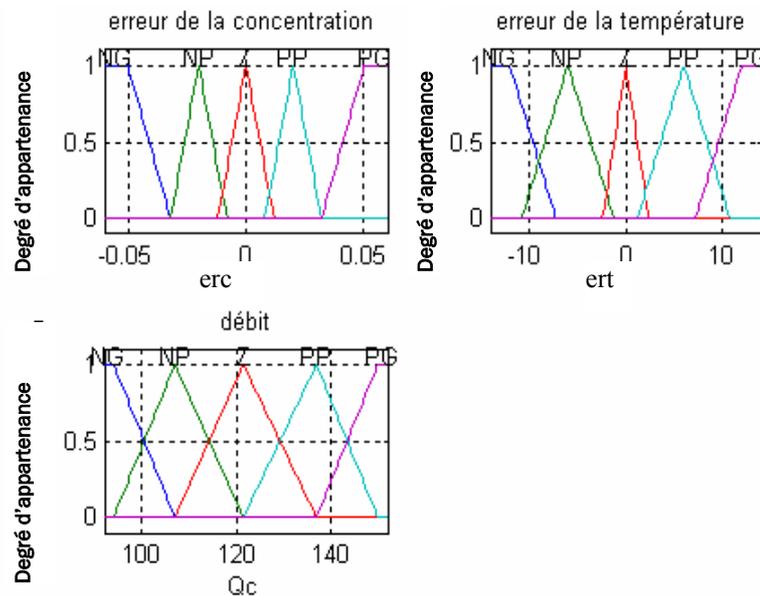


Figure 4.8 : Dédiction des fonctions d'appartenance de : erc, ert, et Q_c par l'AGS

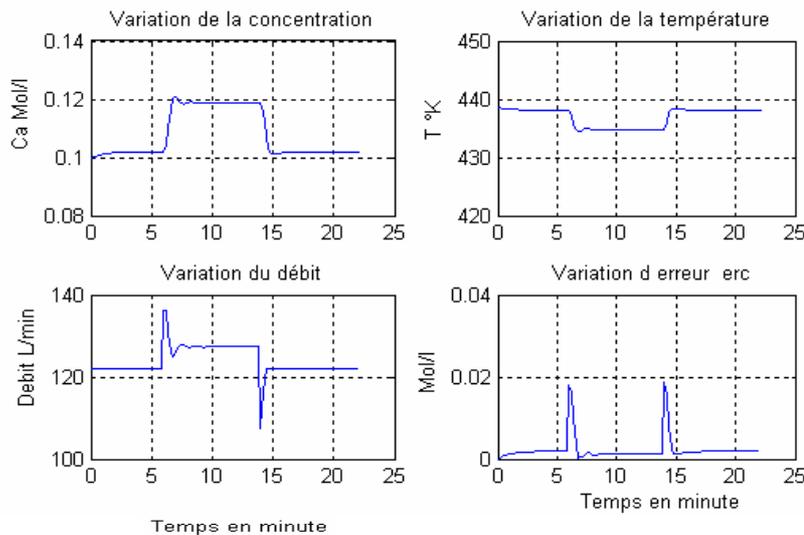


Figure 4.9 : Signaux de sorties de Ca, T, Débit et erc

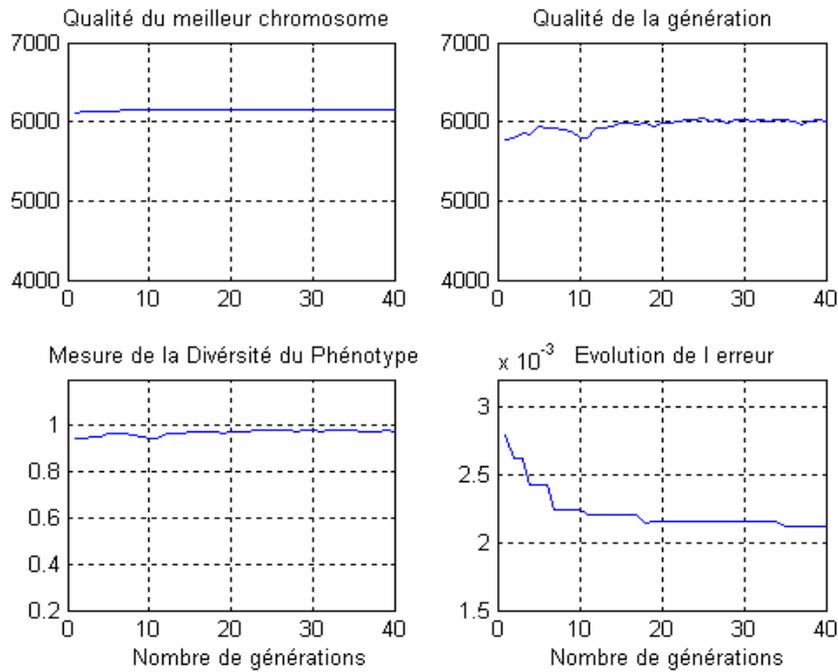


Figure 4.10 : Courbes de suivi de qualité des générations

4.3.2. Cas de la stratégie AGS avec retour de concentration

Dans ce cas le schéma d'asservissement de la figure 3.5 est utilisé, qui consiste à prendre en charge que le retour de concentration. Dans ce cas, le contrôleur flou aura comme entrée l'erreur de concentration erc , sa dérivée $derc$, et comme sortie, la dérivée du débit dQc . Les paramètres globaux de configuration du chromosome sont choisis comme suit :

Population : 40 chromosomes

Longueur du chromosome : 31 gènes

Type de sélection : sélection par roulette.

Type de croisement : aléatoire en un point

Type de mutation : réelle aléatoire

Probabilité de croisement : 0.8.

Probabilité de mutation : 0.03.

Fonction d'adéquation : $f_e = 0.1.f_{01} + 0.5.f_{02} + f_{03} + 3.terr1$

Avec : $f_{0i} = \text{abs}(erc) \leq \delta e$.

δe : intervalle de confiance .

f_{0i} : paramètre qui augmente lorsque erc est à l'intérieur d'un intervalle d'erreur δe égale à 10%, 5% ou 2%

4.3.2.1. Résultats de simulation sans bruit de température du réfrigérant Tc

Les résultats de la figure 4.11 montre la répartition des fonctions d'appartenance déduite par l'AGS.

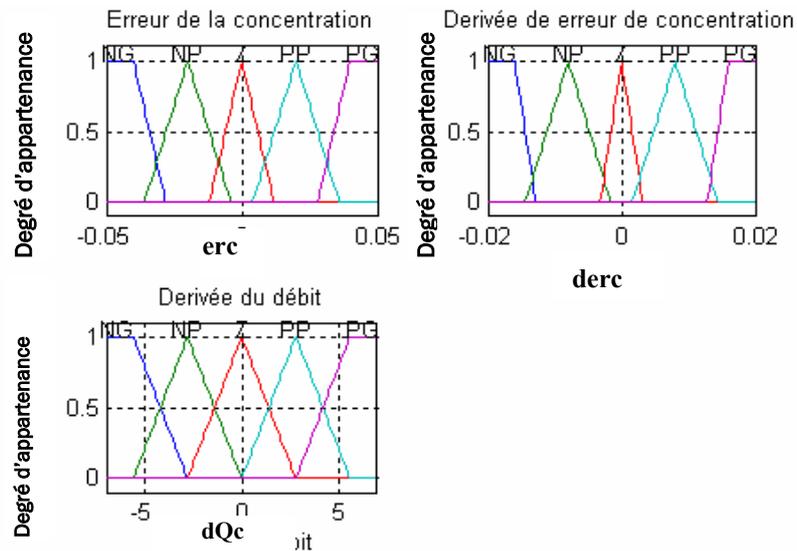


Figure 4.11 : Choix des fonctions d'appartenances de erc, derc , dQc déduite par l'AGS

La figure 4.12, montre les signaux de sortie du réacteur C.S.T.R. On peut remarquer, que, avec une seule boucle de retour l'algorithme génétique dimensionne convenablement les paramètres du contrôleur flou pour effectuer une bonne poursuite de la référence C_{ref} . Aussi, comme la montre la figure 4.12 l'erreur en régime permanent est presque négligeable.

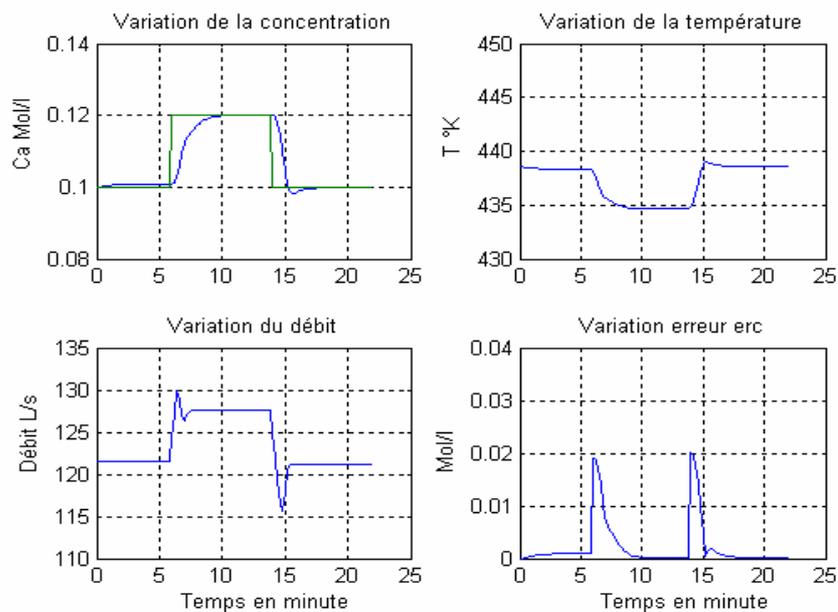


Figure 4.12 : Signaux de sorties de Ca,T, débit et erc

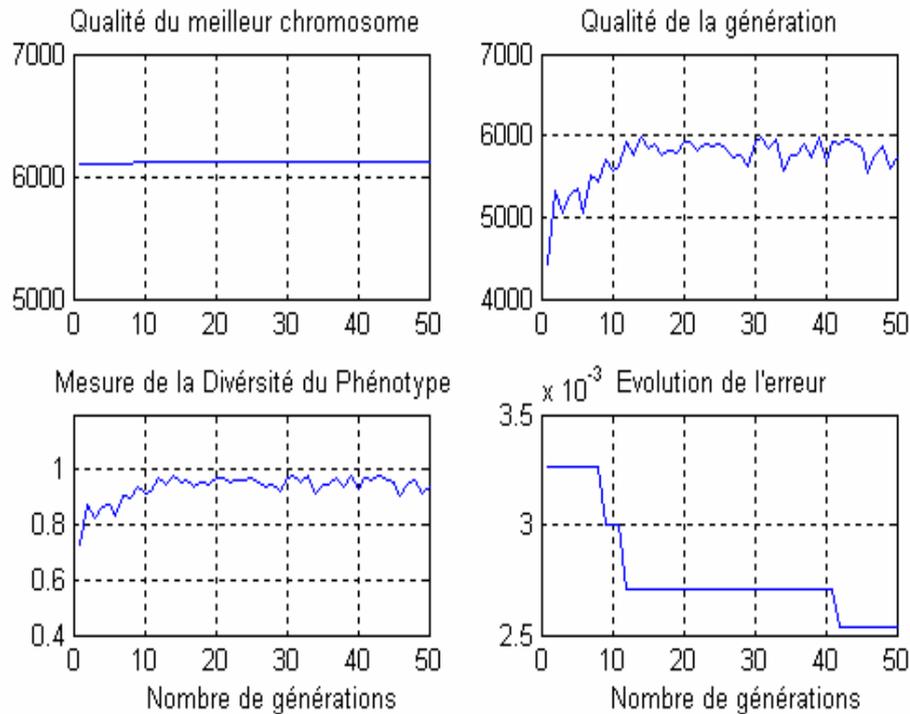


Figure 4.13 : Courbes de suivi de la qualité des générations

L'AGS donne des résultats acceptables dès les premières générations pour converger après la 15^{ème} génération (voir figure 4.13). Cette procédure élimine toute une chaîne de retour et cela avec tous les constituants qui la composent, et permettant d'obtenir un gain important sur le coût de production, qui est un facteur important pour la survie d'une entreprise.

4.3.2.2. Résultats en présence de bruit de température du réfrigérant

Dans cette simulation un bruit de température au niveau du réfrigérant T_c (voir figure 4.4) a été injecté et comme le montre la figure 4.14, l'algorithme génétique arrive à dimensionner la base de connaissance du contrôleur flou pour générer la commande nécessaire au réacteur CSTR. On remarque aussi que la concentration C_a subit une variation de 6% autour de C_{ref} au moment de la perturbation. Lorsque la perturbation disparaît le réacteur suit la concentration de référence C_{ref} d'une manière satisfaisante.

L'AGS commence à converger dès la 20^{ème} génération comme le montre la figure 4.15. Cette stratégie montre que même avec une seule boucle de retour, l'AGS peut optimiser la base de connaissance du contrôleur flou pour qu'il soit robuste vis-à-vis des perturbations de température.

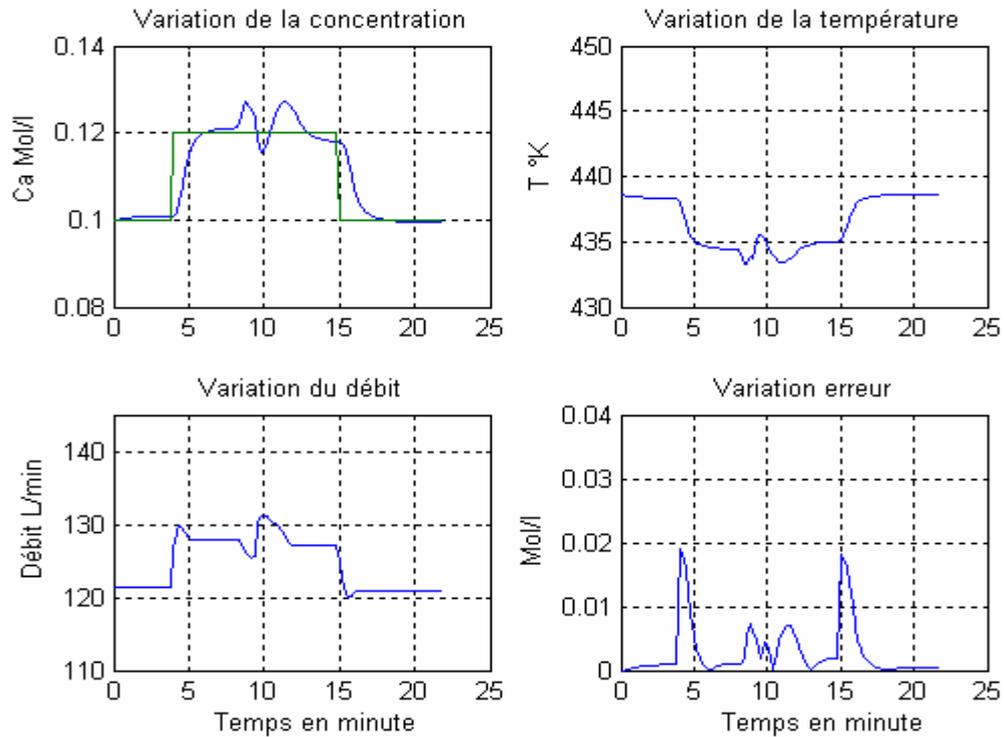


Figure 4.14 : Signaux de sorties de Ca, T, débit et etc

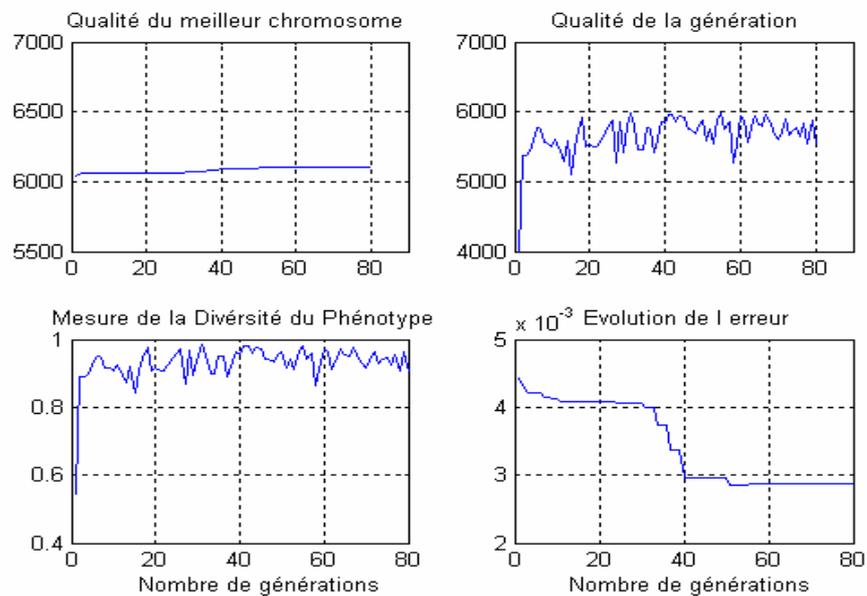


Figure 4.15 : Courbes de suivi de la qualité des générations

4.3.3. Utilisation de la stratégie HGA.

Dans ce cas nous allons voir l'évolution du contrôleur flou lorsqu'il est gouverné par la stratégie HGA1 et cela en appliquant la méthode exposée dans le paragraphe

3.5.2.1, dans ce cas on a optimisé et minimisé le nombre de règles. Le choix des fonctions d'appartenance sera fixé comme le montre la figure 4.16.

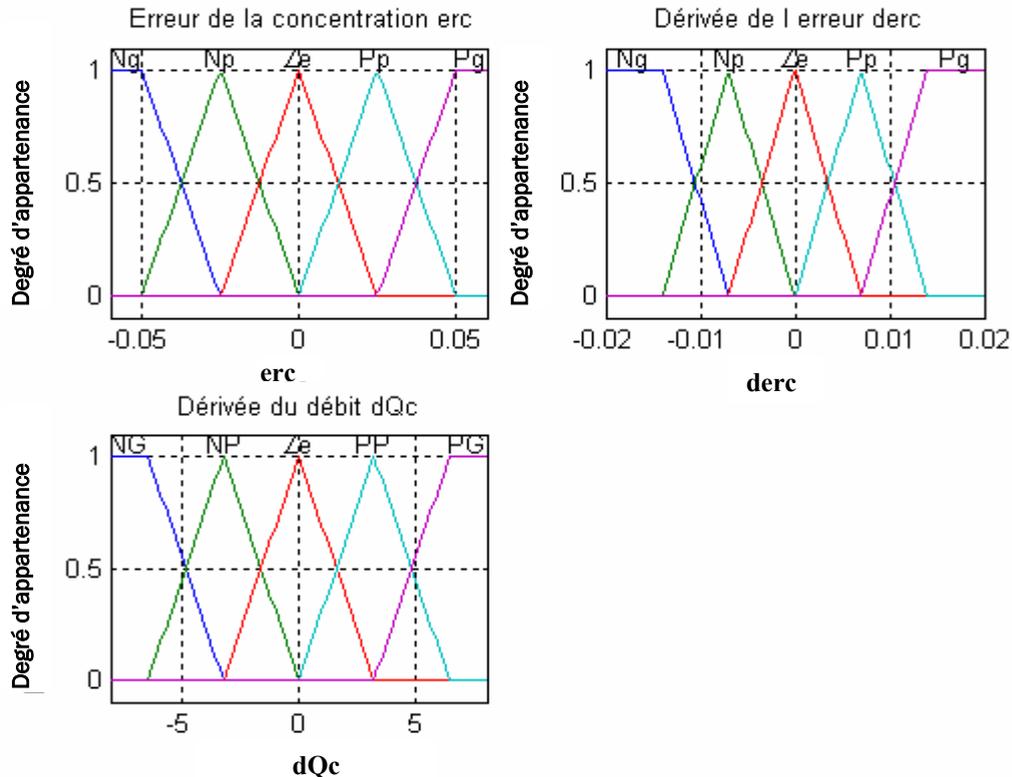


Figure 4.16 : Choix respectif des fonctions d'appartenance de, erc, derc, dQc

Pour ce faire, on a utilisé dans la programmation, le schéma de la figure 3.8 et l'organigramme de la figure 3.9 et les paramètres globaux de configuration du cycle génétique suivants :

Population : 40 chromosomes

Longueur totale du chromosome : 35 éléments dont 10 bits binaires

Type de croisement : aléatoire en un point

Type de mutation : réelle aléatoire pour les gènes de structures et binaire pour les gènes de contrôle

Probabilité de croisement : * Gènes de contrôle : $Pc2= 0.5$

* Gènes de structure : $Pc1= 0.8$

Probabilité de mutation : * Gènes de contrôle : $Pm2=0.02$

* Gènes de structure : $Pm1= 0.04$

Fonction d'adéquation : $fa = 0.1.f01 + 0.5.f02 + f03 + 3.err1 + 40.Nbr0$

Dans ce cas la fonction d'adéquation est la même que précédemment (4.3.2) mais à laquelle on a adjoint un autre objectif par l'ajout d'une variable Nbr0 et ceci pour en prendre en charge le nombre de zéros dans la chaîne des bits de contrôle binaires. Son but est de favoriser les chromosomes ayant un nombre de zéros importants et par conséquent il favorisera la minimisation du nombre de règles floues actives.

Le principal avantage de ce type de codage en base N est qu'il permet de faire intervenir des règles inutiles tout en conservant un nombre total de règles constant. En effet, une règle est inutile sera représentée par un zéro (0) (voir Tableau 4.2) et elle sera codée par un chiffre au même titre qu'une autre règle. De même une opération de mutation dans les GC peut faire disparaître ou apparaître une règle floue durant l'évolution de la population.

4.3.3.1. Résultats de simulation de la stratégie HGA1

Comme le montre la figure 4.18, l'algorithme commence à converger dès la 40^{ème} génération. Les résultats de la figure 4.17 montre la bonne poursuite de la concentration Ca avec la référence Cref. Néanmoins elle présente un léger retard au niveau du régime transitoire, ce qui est tout a fait normal car la température est un paramètre à inertie lente. Par contre l'erreur au régime permanent est presque nulle (voir figure 4.17).

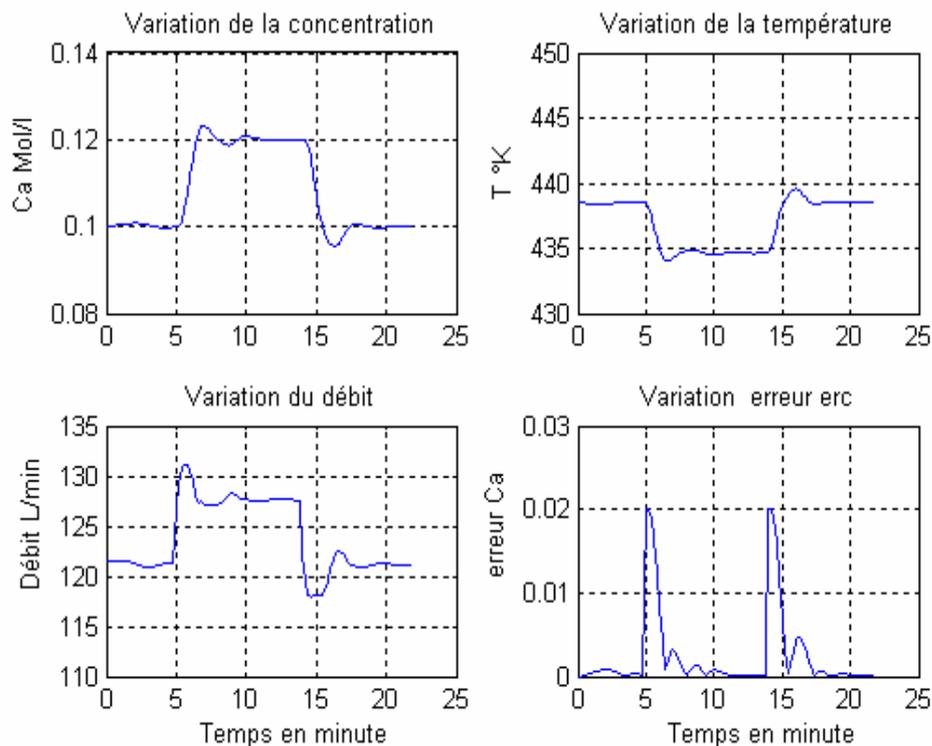


Figure 4.17 : Signaux de sorties de Ca, T, débit et erreur

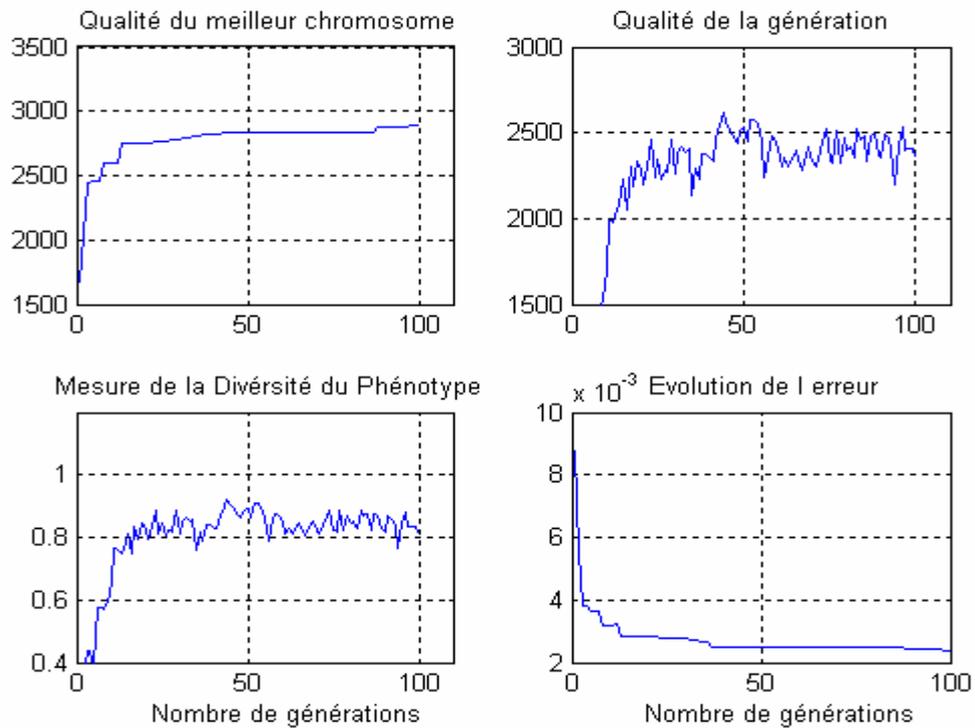


Figure 4.18 : Courbes de suivi de qualité des générations

derc erc	Ng	Np	Z	Pp	Pg
Ng	0	0	0	0	0
Np	Ng	Pg	Np	Pp	Np
Z	Np	Np	Z	Z	Pp
Pp	Z	Pp	Pp	Pg	Pg
Pg	Z	Pp	Ng	Z	Pp

Tableau. 4.2 : Conclusion des règles floues optimisées par le H.G.A.1

4.3.3.2. Résultats de simulation avec bruit dans Tc

Dans ce cas, on introduit une variation de température au niveau du réfrigérant comme le montre la figure 4.19. D'après la figure 4.20, l'HGA1 a optimisé les règles d'inférences floues nécessaires pour faire face aux perturbations de la température pour obtenir une variation de 5% de la concentration autour de Cref. Une fois la perturbation terminée on remarque qu'il y a une bonne poursuite de Ca avec Cref.

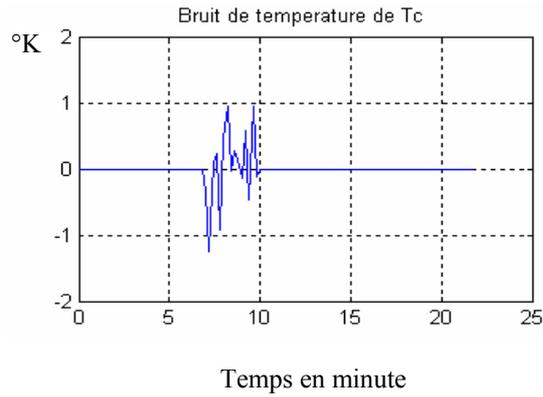


Figure. 4.19 : Bruit de température dans le réfrigérant

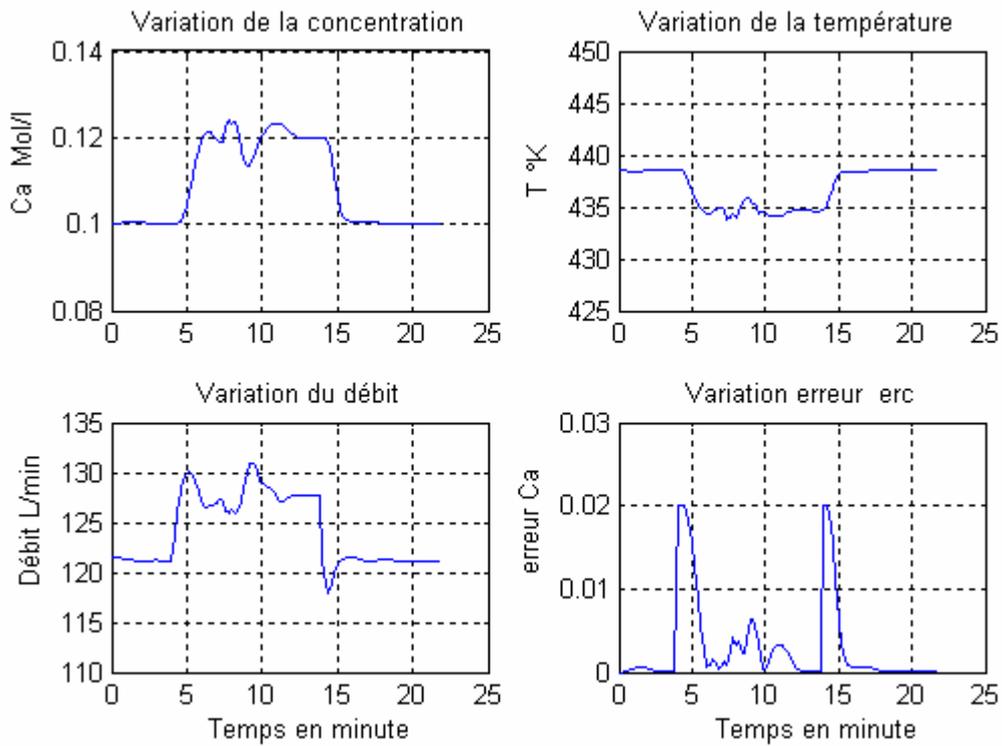


Figure 4.20 : Signaux de sorties de Ca,T, débit et erc

Comme le montre la figure 4.21, la convergence de l'algorithme commence à avoir lieu dès la 50^{ème} génération.

Dans le tableau 4.3, on remarque que le nombre de règles floues utilisées par le contrôleur flou est de 13 au lieu des 25 utilisées lors de l'utilisation d'un AGS. Donc à l'aide de cette stratégie, on a pu avoir une réduction de 50% des règles floues.

derc erc	Ng	Np	Z	Pp	Pg
Ng	0	0	0	0	0
Np	Ng	Pp	Np	Pg	0
Z	Ng	Pp	Z	Z	0
Pp	Pg	Pp	Pp	Pg	Np
Pg	0	0	0	0	0

Tableau 4.3 : Tableau résumant les règles floues optimisées par le HGA1 en présence du bruit

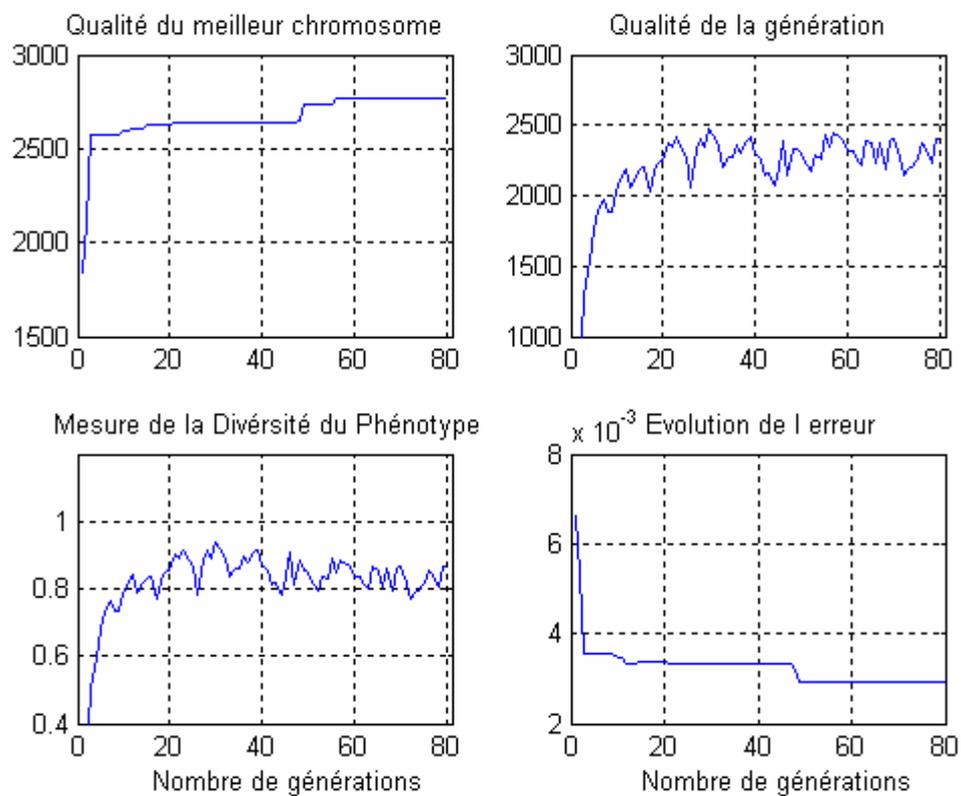


Figure 4.21 : Courbes de suivi de la qualité des générations

4.3.4. Stratégie d'optimisation simultanée

Maintenant est considérée l'évolution du contrôleur flou lorsqu'il est gouverné par un HGA2. Dans ce cas les règles, le nombre de règles et le nombre de fonctions d'appartenance seront optimisées. Pour ce faire, on commence par dimensionner le chromosome pour qu'il accepte sept fonctions d'appartenance pour chaque une des entrées

et de sorties. Dans le programme de simulation on a utilisé, le schéma de la figure 3.5 et l'organigramme de la figure 3.12 et comme paramètres généraux de configuration du cycle génétique suivant :

- * Population : 60 chromosomes.
- * Longueur totale du chromosome : 133 éléments dont 21 en binaires.
- * Type de croisement : aléatoire en un point.
- * Type de mutation : réelle aléatoire pour les gènes de structure et binaire pour les gènes de contrôle.
- * Probabilité de croisement :
 - * Gènes de contrôle : $Pc2 = 0.5$
 - * Gènes de structure : $Pc1 = 0.8$
- * Probabilité de mutation :
 - * Gènes de contrôle : $Pm2 = 0.02$
 - * Gènes de structure : $Pm1 = 0.04$
- * Fonction d'adéquation : $fa = 0.1.f01 + 0.5.f02 + f03 + 3.err1 + 40.Nbr0$

4.3.4.1 Résultats et simulations sans bruit dans Tc

En utilisant la stratégie HGA2 et d'après la figure 4.22, on a une réduction significative du nombre de fonctions d'appartenance utilisées pour chaque une des entrées et sorties du contrôleur flou. Ceci engendrera une réduction du nombre de règles floues.

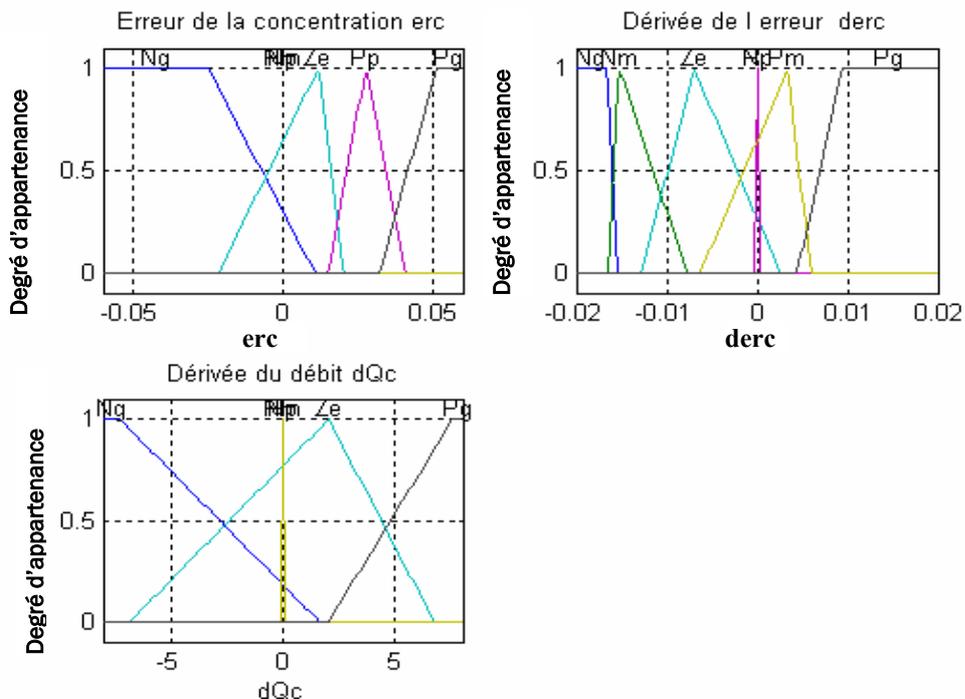


Figure 4.22 : Répartition des fonctions d'appartenance de erc, derc et dQc déduite par l'HGA2 sans bruit.

Comme le montre le tableau 4.4, non seulement on a pu réduire le nombre de règles actives à 7 règles seulement ce qui correspond à une réduction de 84%, mais d'autre part on a obtenu une bonne poursuite de la concentration C_a avec la concentration de référence C_{ref} (voir figure 4.23.), avec une erreur en régime permanent presque négligeable.

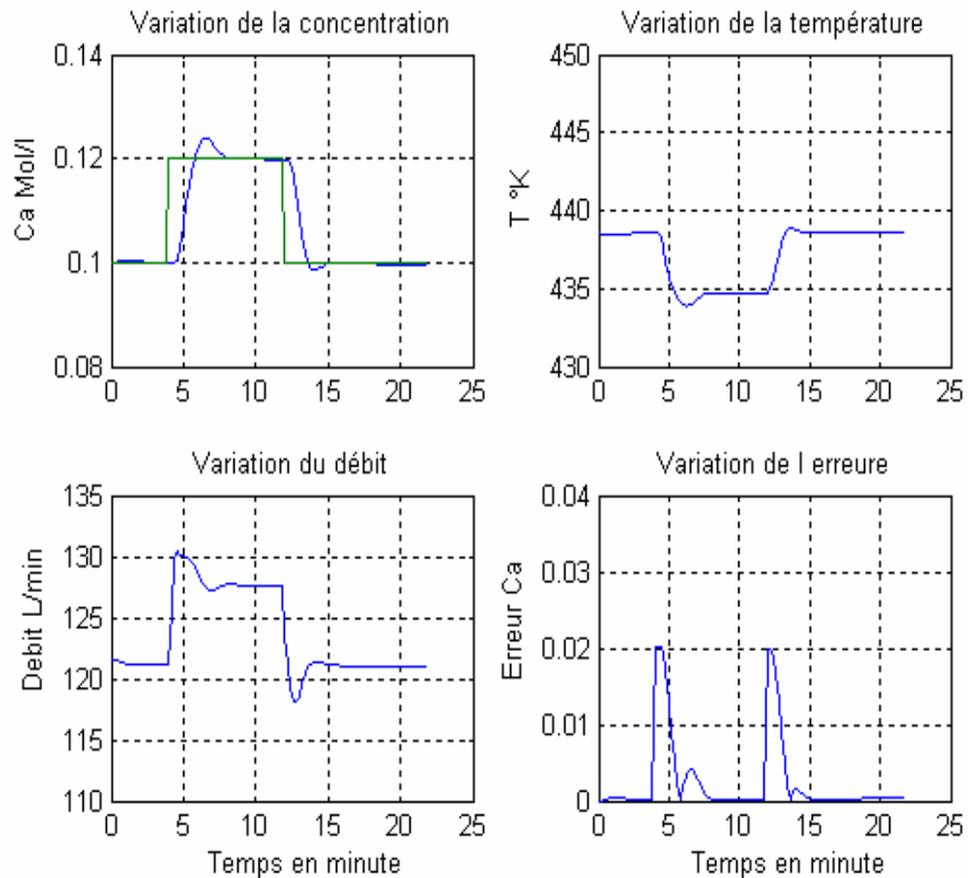


Figure 4.23 : Signaux de sorties de Ca, T, débit et erc

Donc l'utilisation de l'optimisation des écartements des fonctions d'appartenance a pu éliminer les erreurs statiques que génère généralement un contrôleur flou classique. Néanmoins, la convergence de l'algorithme est ralenti et s'est effectuée d'après la figure 4.24 au environ de la 80^{ème} génération.

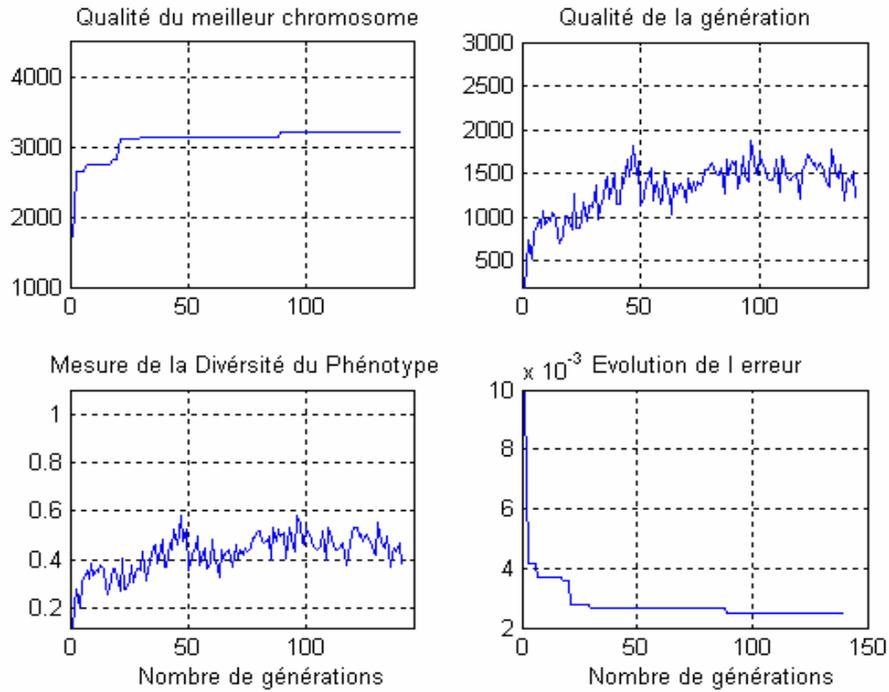


Figure 4.24 : Courbes de suivi de qualité des générations

derc erc	Ng	Nm	Np	Z	Pp	Pm	Pg
Ng	0	0	0	Ng	0	Z	0
Nm	0	0	0	0	0	0	0
Np	0	0	0	0	0	0	0
Z	Ng	0	0	Z	0	Pg	Pg
Pp	0	0	0	0	0	Pg	0
Pm	0	0	0	0	0	0	0
Pg	0	0	0	0	0	0	0

Tableau 4.4 : Conclusion des règles floues, optimisées par le HGA2

4.3.4.2. Résultats de simulation en présence de bruit dans Tc

Dans cette simulation, l'optimisation du contrôleur flou en présence d'une perturbation de température momentanée au niveau du réfrigérant Tc est étudiée.

Comme le montre la figure 4.26, l'optimisation du contrôleur flou par le HGA2 a pu réduire l'influence du bruit dans le réacteur, ce qui a permis une variation de la concentration de l'ordre de 6% au moment de la perturbation. Quant à la commande, on

remarque qu'il y a des fluctuations dans un domaine de variation admissible. Il y a lieu aussi de remarquer que, à partir du tableau 4.5 on a une diminution importante de 80% du nombre de règles floues actives. D'après la figure 4.27 la convergence de l'algorithme a eu lieu au environ de la 100^{ème} génération.

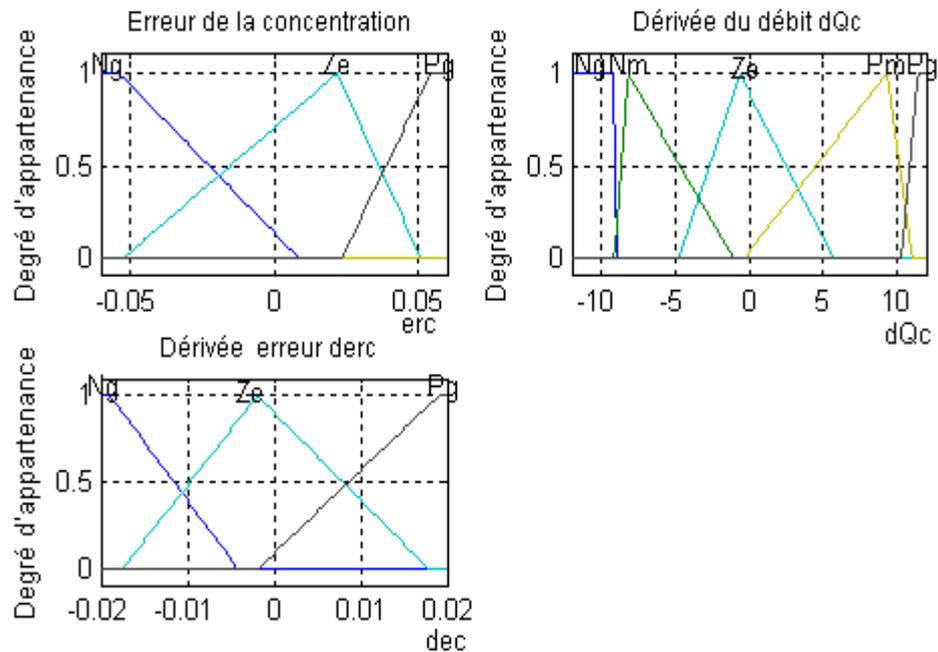


Figure 4.25 : Répartition des fonctions d'appartenance de erc, decr, dQc déduite par l'HGA2 en présence du bruit

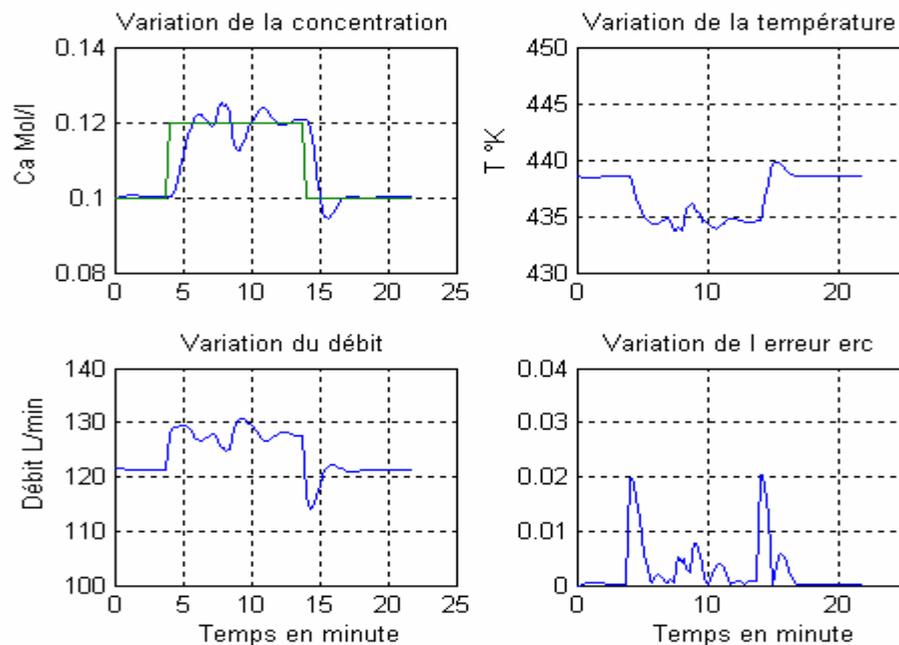


Figure 4.26 : Signaux de sorties de Ca, T, débit et erc en présence de bruit

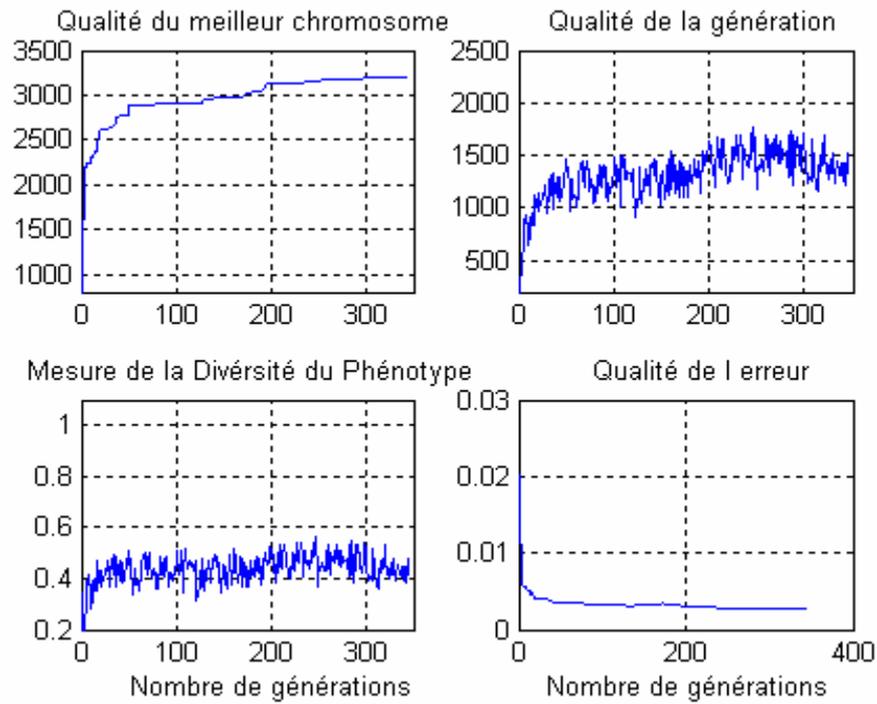


Figure 4.27 : Courbes de suivi de qualité des générations

derc erc	Ng	Nm	Np	Z	Pp	Pm	Pg
Ng	0	0	0	Ng	0	0	Z
Nm	0	0	0	0	0	0	0
Np	0	0	0	0	0	0	0
Z	Ng	0	0	Z	0	0	Pm
Pp	0	0	0	0	0	7	0
Pm	0	0	0	0	0	0	0
Pg	Nm	0	0	Pg	0	0	Pm

Tableau 4.5 : Table des règles floues optimisées par le HGA2

4.4. Résultats d'un contrôleur flou classique

Maintenant une comparaison avec un contrôleur flou classique utilisé généralement pour contrôler un processus dynamique est présentée. A cet effet le tableau des règles floues proposé par Mac Vicar-Whelan [21] (voir Tableau 4.6 à 5 valeurs linguistiques) avec des fonctions d'appartenance du type triangulaire pour l'entrée et la sortie du contrôleur flou (voir figure 4.28), est utilisé.

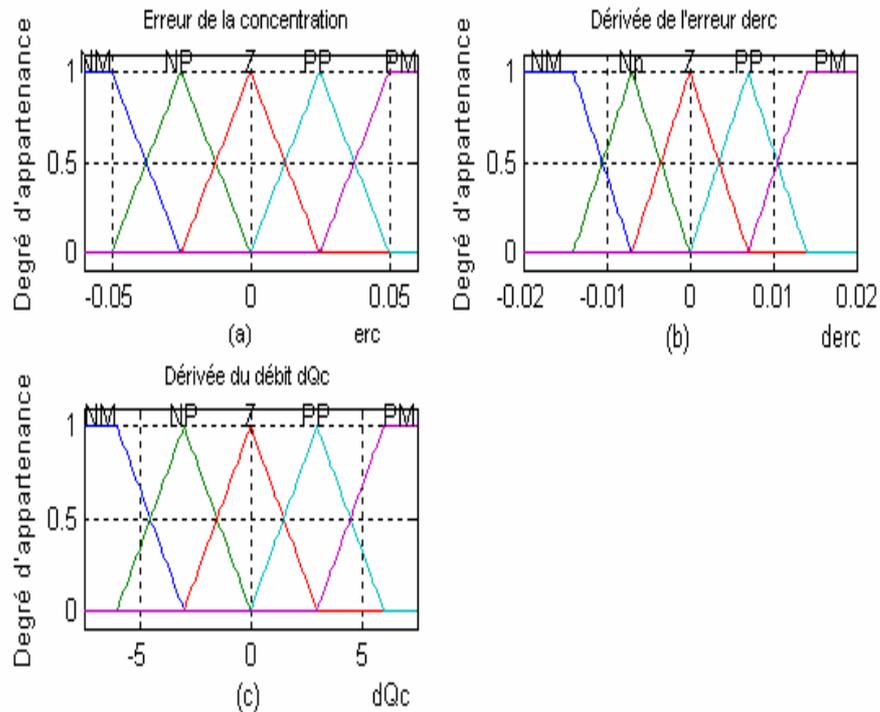


Figure 4.28 : Répartition des fonctions d'appartenance pour :
 a : erreur de la concentration $-0.06 \leq \text{erc} \leq 0.06$ mol/l
 b : dérivée l'erreur de concentration $-0.02 \leq \text{derc} \leq 0.02$ mol/l
 c : dérivée du débit $-8 \leq \text{dQc} \leq 8$ l/min

derc erc	Nm	Np	Z	Pp	Pm
Nm	Nm	Nm	Nm	Np	Z
Np	Nm	Nm	Np	Z	Pp
Z	Nm	Np	Z	Pp	Pm
Pp	Np	Z	Pp	Pm	Pm
Pm	Z	Pp	Pm	Pm	Pm

Tableau 4.6 : Conclusion des 25 règles floues classiques

Les univers de discours ont été choisis identiques avec ceux de la stratégie HGA1 développée au 4.3.3. D'après les figures 4.17, et 4.29, on peut conclure que les résultats d'un contrôleur flou à réglage automatique sont nettement meilleurs au niveau du temps de réponse et des dépassements, ce qui prouve que l'optimisation des règles par le HGA1 a été effectuée d'une manière convenable.

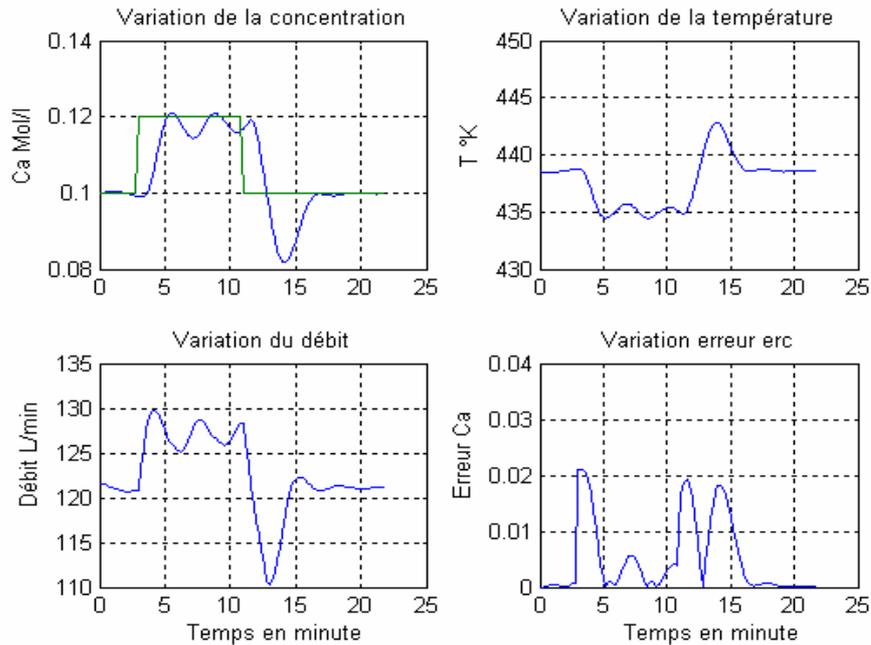


Figure 4.29 : Signaux de sorties de Ca, T, débit et erc

Comme le montre aussi les figures 4.23 et 4.29, la même remarque reste aussi valable lorsqu'on utilise la stratégie optimisant simultanément les règles et les fonctions d'appartenance. Maintenant en prenant une table de décision à 7 termes linguistiques (tableau 4.7), elle est composée de 49 règles floues où les fonctions d'appartenance sont équi-réparties comme le montre la figure 4.30

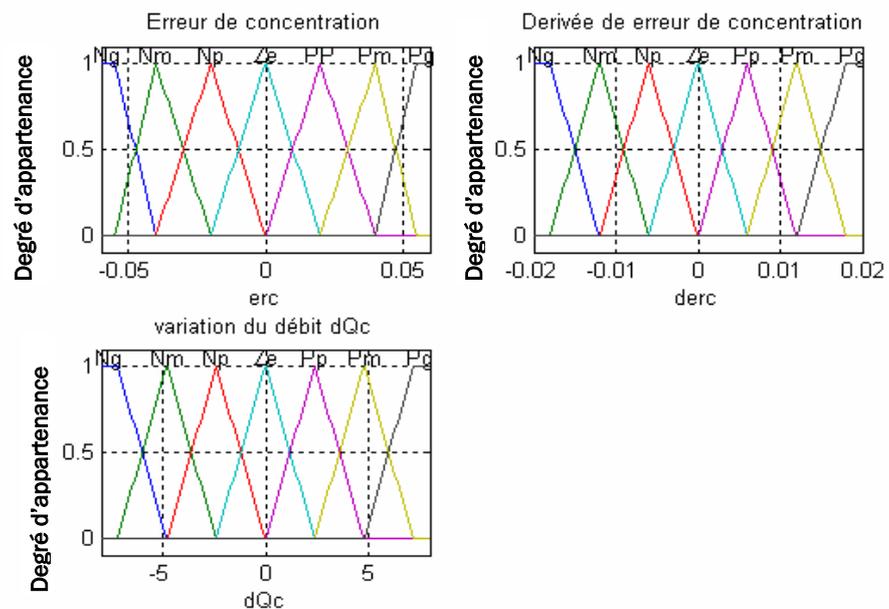


Figure 4.30 : Répartition des 7 fonctions d'appartenance de erc, derc et dQc

derc erc	Ng	Nm	Np	Ze	Pp	Pm	Pg
Ng	Ng	Ng	Ng	Ng	Nm	Np	Ze
Nm	Ng	Ng	Ng	Nm	Np	Ze	Pp
Np	Ng	Ng	Nm	Np	Ze	Pp	Pm
Ze	Ng	Nm	Np	Ze	Pp	Pm	Pg
Pp	Nm	Np	Ze	Pp	Pm	Pg	Pg
Pm	Np	Ze	Pp	Pm	Pg	Pg	Pg
Pg	Ze	Pp	Pm	Pg	Pg	Pg	Pg

Tableau 4.7 : Conclusion des 49 règles floues classiques

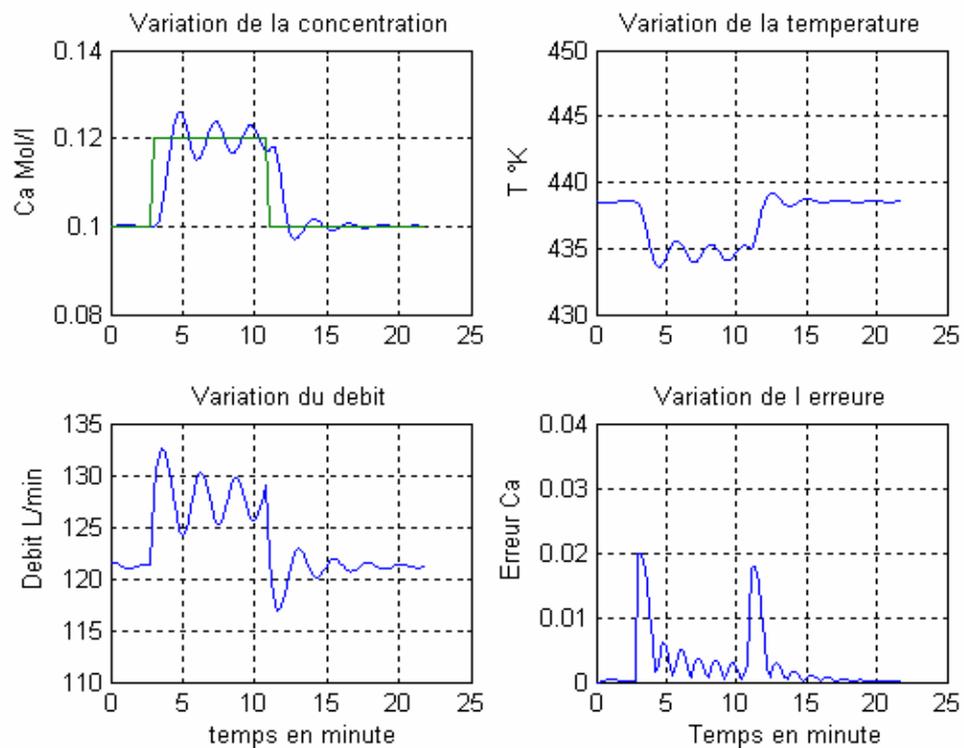


Figure 4.31 : Signaux de sorties de Ca, T, débit et erc (7 Fonctions d'appartenance)

D'après la figure 4.31, on remarque qu'il y a une sensible amélioration de la poursuite en augmentant le nombre de fonctions d'appartenance. Cependant les résultats obtenus à partir des tables de décision automatisées des AGS, HGA1 et HGA2 donnent une meilleure poursuite de la concentration de référence comme le montre respectivement les figures 4.12, 4.17 et 4.23. Le tableau 4.8 résume les principaux résultats de simulation et montre l'efficacité des méthodes automatiques par rapport à un contrôleur flou classique.

<i>Type</i>	<i>Nombre de boucles</i>	<i>Longueur chromosome</i>	<i>Nombre de règles</i>	<i>Nombre Itérations</i>	<i>Erreur statique</i>	<i>Dépassement</i>
<i>AGS</i>	2	31	25	8	4%	1%
<i>AGS</i>	1	40	25	15	nul	0
<i>HGA1</i>	1	35	20	40	nul	4%
<i>HGA2</i>	1	133	7	80	nul	4%
<i>Classique</i>	1	-	49	-	15%	20%

Tableau 4.8 : Tableau récapitulatif des résultats obtenus

4.5. Conclusion

Dans ce chapitre les résultats de simulation des différentes configurations des AG qui permettent d'optimiser la base de connaissance d'un contrôleur flou sont présentés. Les résultats obtenus montrent l'efficacité des AG dans le contrôle du réacteur chimique C.S.T.R, et cela, par la bonne poursuite de la concentration de référence en utilisant une seule boucle de retour (retour de concentration) ou deux boucles de retour. De même, il présente une robustesse en présence de perturbations extérieures dans T_c

D'après les résultats obtenus, il y a une augmentation du temps de convergence de l'AG par l'utilisation du processus d'optimisation à l'aide des HGA, par contre il y a une réduction considérable des règles actives utilisées. Aussi, malgré la diminution excessive du nombre de règles floues, les performances du système sont restées dans un domaine acceptable.

Enfin, ce chapitre se termine par une comparaison entre un contrôleur flou du type PD classique et un contrôleur flou optimisé par les AG. On a remarqué que ces derniers donnent de bons résultats et cela avec une réduction remarquable des règles floues. Ceci prouve que l'automatisation de ces dernières constitue un bon alternatif et améliore les performances d'un contrôleur flou, de manière appréciable.

CONCLUSION

Ce travail a montré la contribution des AG pour configurer la base de connaissance d'un contrôleur flou en absence d'un expert, ou dans des situations pratiques qui ne permettent pas de connaître suffisamment le fonctionnement du système. De même, on a vu qu'on peut établir la loi de commande du système sans avoir recours à un calcul mathématique exact de la commande. A cet effet plusieurs méthodes ont été présentées pour la synthèse d'un contrôleur flou à l'aide des algorithmes génétiques qui optimisent :

- Les fonctions d'appartenance des termes linguistiques permettant la qualification des entrées du contrôleur.
- Les règles floues intervenant dans le processus d'inférence.
- Les fonctions d'appartenance des termes linguistiques décrivant l'état des sorties.
- Les fonctions d'appartenance et les règles floues simultanément.

L'algorithme génétique a permis aussi de définir au mieux la structure du contrôleur flou qui assure le réglage dynamique du système. Cet algorithme génétique est utilisé pour exploiter les données issues du réacteur chimique de type CSTR afin que le contrôleur flou génère la commande nécessaire pour effectuer une bonne poursuite de la concentration de référence C_{ref} . Les performances du système asservi, analysées par l'intermédiaire de critères calculés sur sa réponse à l'échelon, permettent de définir la fonction d'évaluation utilisée par l'algorithme génétique durant la phase d'optimisation.

Pour ce faire, on doit tout d'abord définir et choisir les codages nécessaires pour représenter tous les paramètres du contrôleur que l'on souhaite optimiser. Cela ne doit théoriquement poser aucun problème, puisque la méthode d'optimisation décrite peut traiter des paramètres codés en binaire, en base N ou en nombre réel. Pour définir assez simplement le mode de codage, on peut utiliser les quelques règles suivantes :

- Le processus d'optimisation doit permettre d'intégrer au mieux les fonctions d'appartenance des termes qualifiant les entrées lorsque celles-ci correspondent à des mesures réalisées sur le système. Le choix d'un type de codage est guidé principalement par la précision souhaitée lors de la prise en compte d'une entrée :

codage par une chaîne binaire de longueur fixe pour une précision modérée et codage par un nombre réel pour une précision élevée.

- Les règles floues sont également définies par l'intermédiaire du processus d'optimisation. Les règles connues à priori, peuvent être utilisées directement. Dans ce cas, elles ne sont pas codées dans les individus définissant une solution possible du problème d'optimisation. Pour les règles devant être définies par le processus d'optimisation, on utilise un codage en base N où la base sera conçue en fonction du nombre de termes linguistiques décrivant les conclusions des règles.
- La synthèse par l'algorithme génétique permet également de définir les fonctions d'appartenance des termes linguistiques qualifiant les sorties, qui correspondent aux paramètres du système. Lorsqu'une sortie prend uniquement un certain nombre de valeurs connues a priori, on ne définit pas de variable linguistique pour la qualifier et aucun codage n'est alors nécessaire.

Après un bref aperçu sur l'aspect dynamique du réacteur chimique CSTR, nous avons vu qu'on peut le commander de deux manières différentes, soit en utilisant deux boucles de retour, soit une seule boucle de retour. Dans les deux cas l'utilisation des AGS et des HGA ont permis d'aboutir à une base de connaissance qui donne la commande de débit nécessaire au contrôle du CSTR. Quoique la poursuite n'est pas idéale mais on est resté toujours dans les limites de performance souhaitées et cela au bout d'une trentaine de générations. Donc les AG viennent constituer une alternative puissante vis à vis d'un expert et intéressante pour ce genre de problème, car ils ont montré aussi une certaine robustesse au cours des perturbations extérieures qui peuvent altérer le système.

Les résultats de simulations ont montré que les AG donnent de bons résultats et plus rapidement si on tient compte des données connues par le système à contrôler (utilisation de Tref)

Cette étude a montré qu'il est facile d'intégrer un critère multi objectif dans le processus d'optimisation.

Cette étude a également montré que l'introduction de critères multi – objectifs au niveau de la fonction d'adaptation a pu diminuer les erreurs statiques que génère généralement un contrôleur flou classique et cela par le biais de l'exploration et l'exploitation des AG pour optimiser les règles d'inférence floue.

En ce qui concerne la topologie des AG, on a remarqué qu'en utilisant les HGA, une certaine souplesse est observée dans le mécanisme de réduction assez conséquent des règles et des fonctions d'appartenance, tout en restant toujours dans les meilleures conditions de fonctionnement du système.

Nous pensons que le travail décrit dans ce mémoire peut se poursuivre dans plusieurs directions. Une voie de recherche reste à développer dans le parallélisme pour diminuer le temps de convergence de l'algorithme, et d'autre part afin de valider complètement cette méthode on doit l'appliquer à d'autres système pour résoudre les problèmes de réglages dynamiques des paramètres du contrôleur flou .Une autre voie à développer , est la recherche automatique des paramètres de contrôle qui conditionnent le bon fonctionnement des algorithmes génétiques .

APPENDICE A
LISTE DES ABREVIATIONS

ADN	: acide désoxy ribonucléique
AG	: algorithme génétique
AGS	: algorithme génétique Standart
AGH	: algorithme génétique hiérarchique
ARN	: acide ribonucléique
CSTR	: continuous stirred tank reactor
Fitness	: fonction d'adéquation
GC	: gène de contrôle
GS	: gène de structure
MDP	: mesure de la diversité du phénotype
P	: proportionnel
P D	: proportionnel dérivé
RS	: régulateur de structure
RC	: régulateur de contrôle
TAF	: transcription action facteur
fa	: fonction d'adaptation
fe	: fonction d'évaluation
m ARN	: message acide ribonucléique

REFERENCES

1. Holland, J.H., "Adaptation in Natural and Artificial Systems ", University of Michigan Press, Ann Arbor, (1975).
2. Davidor, Y., "Genetic Algorithms and Robotics ", Word Scientific, (1991) .
3. Davis, L.D., "Handbook of Genetic Algorithms ", Van Nostrand Reinhold, New York, (1991).
4. Goldberg, D.E., "Genetic Algorithms in Search, Optimisation and Machine Learning ", Addison-Wesley Reading, M.A., (1989).
5. Goldberg, D.E., "Algorithmes Génétiques. ", Edition Addison Wesley, France, (1994).
6. Jean Michel, R., "Algorithmes Génétiques, Réseau de neurones. Application à la commande des processus ", Edition Hermès, Paris, (1995).
7. Mitchell M., "An Introduction to Genetic Algorithms ", a Bradford Book ,The MIT Press , England, fifth printing, (1999).
8. Syswerda, G., "Uniform Crossover in genetic Algorithm ", Proceeding of Third International conference on Genetic Algorithms, (1989), 2-9
9. Michalewicz, Z., "Genetic Algorithms + Data Structure = Evolution ", Springer-Verlag, New-york , Second, Extended Edition, (1994).
10. Arabas, J., Michalewicz Z., Mulawka J., " a Genetic Algorithm With Varying Population Size GAVaPS ", Proc of First IEEE, Conference on Evolutionary Computation, (1994), 73-78.
11. Man K.F., Tang K.S. and Kwong S., "Genetic Algorithms, Concepts and Designs ", Edition Springer-Verlag , (2001) .
12. De Jong A., " An analysis of the behaviour of a class of genetic adaptive systems. " ph.D. Dissertation 76-9381, University of Michigan, (1975).
13. Schaffer J.D., Caruna R.A., Eshelman L.J. et Das R., " A study of control parameters online performance of genetic algorithms for function optimization " , dans Proceedings of the third Int. Conf. on Genetic Algorithms, (1989), 51–60.

14. Hesser J. et Männer R. , “ Towards an optimal mutation probability in genetic algorithms. “, dans *Parallel Problem Solving from Nature*, (1991), 23–32.
15. Trambouz, P., “ Matériel et équipement “, Édition tecnip Institut Français de Pétrole, (1999).
16. Villermaux, J., “ Génie de la réaction Chimique , Conception et fonctionnement des réacteurs “ , Éd Lavoisier ,(1985).
17. Mouklouv, I., “ Principe de la technologie Chimique “ , Ed. Mir, (1986).
18. Trambouz, P., Vanlandeghem, H., “ les réacteurs chimiques, conception, calcul, mise en œuvre “ , Ed. Tecnip, Paris, (1984).
19. Horak, J. et Pasek, J., “ Conception des réacteurs chimiques industrielles sur la base de données de laboratoire “ , Ed. Eyrolles, Paris, (1981).
20. Ramirez, W.F., “ Computational Methods for process Simulation “ , Ed. ButterWorth , (1989).
21. Bernadette, B.M., “ La logique floue et ses applications “ , Edition Cépaduès, (1997).
22. Gacone, Louis., “ Eléments de logique floue “ , Edition Hermès, (1997).
23. Henk, S., “ Logique floue et régulation, Théorie et Pratique de la régulation “ , Edition Elektor, Pays Bas, (1995).
24. Lee, C.C., “ Fuzzy logic in control system : fuzzy logic controller “, Part I and Part II, *IEEE Transaction on Systems, Man and Cybernetics* 20(2), (1990) , 404-435.
25. Carse, B., Fogarty, T.C. and Munro, A., “ Evolving fuzzy rule based controllers using genetic algorithms “ , *Fuzzy Sets and Systems*, Vol. 80, (1996) , 273-293.
26. Kinzel, J., Klawonn, F. and Kruse, R., “Modifications of genetic algorithms for designing and Optimisation fuzzy controller “ , 1st IEEE int. Conference Evolutionary computation , (1994), 28-33 .
27. Krop K., “ Optimisation of fuzzy logic controller inference rules using genetic Algorithms “ , *EUFIT 93* , (1993),1090-1093.
28. Togai, M. et Watanabe, H., “Expert system on a chip: An engine for real-time approximate reasoning “ , *IEEE Expert Syst. Mag.*, 1, (1986) , 55–62.
29. Homaifar et McCormik, E., “Simultaneous design of membership functions and rules Sets for fuzzy controllers using genetic algorithms “ , *IEEE Trans. On Fuzzy Systems*, (1995), 129–139.
30. Chwee, Ng.K. et Li, Y., “Design of sophisticated fuzzy logic controllers using genetic algorithms. “, Dans *Proc. Of ICCI94 / Fuzzy Systems*, (1994), 1708–1712.

31. Lee, M.A. et Takagi, H., “ Dynamic control of genetic algorithms using fuzzy logic techniques “, dans Proceedings of the 5th Int’l Conf. on Genetic Algorithms, (1993) , 76–83.
32. Mamdani, E.H., Assilian, S., “An experiment in linguistic synthesis with a fuzzy logic Controller “, Int. J. Man Mach. Studies, 7(1), (1975), 1–13.
33. Takagi, T. , Sugeno, M., “ Fuzzy identification of systems and its applications to modeling and control. “, IEEE Trans. Syst. Man and Cybern., SMC15(1) , (1985), 116–132.
34. Sugeno, M., “Industrial Application of Fuzzy Control “, Elsevier Science Pub.Co., (1985).
35. Herrera, F., Lozano, M. et Verdegay, J.L., “ Generating fuzzy rules from examples using genetic algorithms“, dans Fuzzy Logic and Soft Computing, World Scientific , Singapore, (1995), 11–20 .
36. Zadeh, L.A., “Soft computing and fuzzy logic“, IEEE Software, 11(6), (1994), 48–56.
37. Chen, C.H., “Fuzzy logic and neural network handbook “, McGraw-Hill, (1996).
38. Lighbody, G. and Irwin, G.W., “ Nonlinear Control Structures Based on Embedded Neural System Models , “ IEEE Trans. on Neural Networks, Vol.8 , N°3, (May 1997), 553-567.