

République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université de Batna

Faculté des Sciences de l'Ingénieur

Département d'Electrotechnique

MEMOIRE

Pour l'obtention du diplôme

Magister en électrotechnique

Option : matériaux électrotechniques

Présenté et soutenu par :

Barkat toufik

Thème :

Intelligence en essaim, état de l'art et applications dans les matériaux diélectriques

Le :

Devant le jury composé de :

M.Kadjoudj	M.C	Université de Batna	Président
I.Mokhnache	M.C	Université de Batna	Rapporteur
S.Taibi	M.C	Université de Batna	Co-Rapporteur
N.Nait Said	M.C	Université de Batna	Examineur
M.Benmohamed	Prof	Université de Constantine	Examineur

Sommaire

INTRODUCTION GENERALE

CHAPITRE I : GENERALITES SUR LES DIELECTRIQUES

I.1 INTRODUCTION.....	1
I.2 LES DIELECTRIQUES GAZEUX	1
I.2.1 Caractérisation des décharges gazeuses.....	1
I.2.1.1 Principes de base et phénomènes d’amorçage en tension continue	1
I.2.1.2 Caractéristiques de la décharge dans les gaz	3
I.2.3 Influence de différents paramètres sur la rigidité diélectriques dans les gaz.....	5
I.2.3.1 Influence de la distribution du champ électrique appliqué.....	5
I.2.3.2 Influence du type de tension appliquée.....	6
I.3 DIELECTRIQUES LIQUIDES	7
I.3.1 claquage dans les liquides.....	7
I.3.1.1 Mécanisme électronique.....	7
I.3.1.2 Mécanisme de claquage avec phase gazeuse.....	8
I.3.1.3 Mécanisme de claquage par pont.....	9
I.3.2 Influence de différents paramètres sur la rigidité diélectrique.....	10
I.3.2.1 humidité.....	10
I.3.2.2 Pression hydrostatique.....	10
I.3.2.3 Distance entre électrodes.....	10
I.3.2.4 Température.....	11
I.3.2.5 Nombre de claquages.....	11
I.3.2.6 Influence des écrans.....	11
I.3.2.7 Durée d'application de la tension.....	11
I.4 LES DIELECTRIQUES SOLIDES.....	12
I.4.1 Conduction électrique dans les solides.....	12
I.4.2 Les pertes diélectriques.....	14
I.4.2.1 Permittivité relative.....	14
I.4.2.2 Angle de pertes diélectriques et facteur de dissipation diélectrique.....	15

I.4.3 Mécanismes de claquage dans les diélectriques solides.....	16
I.4.4 Influence de divers paramètres sur la rigidité diélectrique.....	17
I.4.5 Vieillissement des isolants solides sous l'effet du champ.....	17
I.4.6 Tenue des matériaux isolants solides aux contraintes thermiques.....	18
I.4.7 Vieillissement des isolants solides sous l'effet de l'environnement.....	18
I.4.8 Tenue des matériaux isolants solides aux contraintes mécanique.....	19
I.5 CONCLUSION.....	19

CHAPITRE II : L'ETAT DE L'ART DE CLASSIFICATION ET LES SYSTEMES COMPLEXE

II.1 INTRODUCTION.....	20
II.2 LES TROIS PRINCIPALES ETAPES DE LA CLASIFICATION.....	21
II.2.1 La préparation des données.....	21
a) Variables et sélections.....	21
b) Distances et similarités.....	22
II.2.2 Le choix de l'algorithme.....	23
a) La taille des données.....	23
b) La nature des données.....	23
c) La forme des clusters.....	24
d) Le type de résultats attendus.....	24
II.2.3 L'exploitation des clusters.....	24
II.3 LA SIMILARITE.....	25
II.3.1 Propriétés formelles de base.....	25
II.3.2 Similarité et variables numériques.....	27
II.3.3 Similarité et variables symboliques.....	28
II.4 LES DIFFERENTES METHODES DE CLUSTERING.....	28
II.4.1 Le clustering hiérarchique.....	30
a) Les hiérarchies strictes.....	31
b) Les pseudo- hiérarchies.....	32
II.4.2 Le clustering par partitionnement.....	34
a) Le partitionnement strict.....	34
b) Pseudo- partitions et partitions flous.....	35
II.5 PHENOMENE D'EMERGENCE ET L'INFORMATIQUE.....	38
II.6 CARACTERISATION D'UN PHENOMENE EMERGENT.....	39
II.7 L'AUTO- ORGANISATION COMME TECHNIQUE D'EMERGENCE.....	40
II.7.1 Définitions.....	41
II.7.2 Les mécanismes de l'auto- organisation.....	42
II.8 L'INTELLIGENCE EN ESSAIM.....	43
II.9 EMERGENCE ET SYSTEMES ARTIFICIELS COLLECTIFS.....	44
II.9.1 Algorithmes évolutionnaires.....	44
II.9.2 Les algorithmes génétiques.....	45
II.9.3 Les réseaux de neurones.....	49
II.9.4 Algorithmes à essais de particules.....	52
II.9.5 Les algorithmes de fourmis artificielles.....	54

II.10 CONCLUSION.....	55
------------------------------	-----------

CHAPITRE III : LES FOURMIS ARTIFICIELLES

III.1 INTRODUCTION	56
III.2 GENERALITES SUR LES FOURMIS.....	57
III.3 QUELQUES CONCEPTS S DE BASE	60
III.3.1 Problème d’optimisation.....	60
III.3.2 Méthodes de résolution.....	60
a) Les heuristiques.....	61
b) Les métaheuristiques.....	61
III.3.3 Les fourmis artificielles.....	61
III.4 LES ALGORITHMES DE FOURMIS ARTIFICIELLES POUR L’OPTIMISATION COMBINATOIRE.....	61
III.4.1 Inspiration biologique : le fourragement collectif par stigmergie.....	62
a) Le pont binaire.....	62
b) optimisation du chemin pour la recherche de nourriture.....	64
c) Le pont avec un obstacle.....	65
III.4.2 Modèles de fourragement pour la résolution des problèmes d’optimisation..	65
III.4.2.1 optimisation par colonies de fourmis et problème du voyageur de Commerce.....	65
III.5 LES ALGORITHMES DE FOURMIS ARTIFICIELLES POUR LA CLASSIFICATION AUTOMATIQUE.....	68
III.5.1 Inspiration biologique.....	68
III.5.2 Modèles du tri du couvain et application à la classification de données.....	68
a) Les pionniers.....	68
b) Modèle de Lumer et Faieta.....	70
III.6 CONCLUSION.....	73

CHAPITRE IV : SIMULATION ET RESULTATS

IV.1 INTRODUCTION.....	74
IV.2 LE LANGAGE DE PROGRAMMATION.....	74
IV.2.1 Les tortues	74
IV.2.2 Les patchs.....	74
IV.2.3 L’observateur.....	75
IV.2.4 Les commandes.....	75
V.2.4.1 Commandes primitives.....	76
IV.2.5 Reporters.....	77
IV.2.6 Boutons.....	78
IV.2.7 Les variables.....	78
IV.2.7.1 Les variables locales.....	79
IV.2.7.2 Les variables prédéfinies.....	79
IV.2.8 Les fonctions.....	80

IV.2.9 Affichage de résultats	81
IV.2.10 Exemple d'une interface du NetLogo	81
IV.3 LES RESULTATS DE CLASSIFICATION	81
IV.3 .1 La variation de rayon de détection	83
IV.3 .2 la variation des nombres des fourmis	83
IV.3 .3 la variation des nombres des d'objets	84
IV.3 .4 augmentation des types isolants	85
IV.3 .5 Changement de la taille de l'environnement	86
IV.4 CONCLUSION	87
CONCLUSION GENERALE	

Listes des tableaux

Tableau 2.1 : Propriétés mathématiques des différentes mesures de disimilarité.....26

Listes des figures

Figure 1.1 Courbe de Paschen montrant la variation de la tension de claquage en fonction de pd	3
Figure 1.2 : Caractéristiques d'une décharge gazeuse.....	3
Figure 1.3 : La configuration plan-plan les lignes de champs sont parallèles et le champ est parfaitement homogène dans l'espace inter électrodes. Dès qu'un streamer prend naissance il se propage jusqu'à l'anode.....	5
Figure 1.4 : La configuration pointe positive plane à la terre, les lignes de champ montrent la dissymétrie du champ électrique entre électrodes.....	5
Figure 1.5 : Conduction dans un fil de longueur l et de section s	13
Figure 1.6 : Conduction dans une plaque d'épaisseur e et de surface S	13
Figure 1.7 : Circuit équivalent d'une isolation.....	15
Figure 1.8 : l'angle de pertes.....	15
Figure 2.1 Les différentes étapes du processus de clustering.....	22
Figure 2.2 Taxonomie de la classification inspirée de celle de Jain et Al.....	30
Figure 2.3 : Exemple de dendrogramme.....	31
Figure 2.4 Algorithme divisif hiérarchique DIANA.....	32
Figure 2.5 Exemple de pyramide.....	33
Figure 2.6 Algorithme de construction d'une pyramide.....	34
Figure 2.7 Exemples de croisements (gauche et centre) et d'inversions (droite) dans les pyramides.....	34
Figure 2.8 Algorithme de partitionnement k-moyennes.....	36
Figure 2.9 Algorithme de partitionnement k-moyennes flou.....	37
Figure 2.10 Le proto-émergentisme (en haut) et le néo-émergentisme (en bas).....	39
Figure 2.11 Codage de l'information.....	47
Figure 2.12 Modèle de la roulette.....	
Figure 2.13 Opérateur de croisement mono point appliqué à deux chromosomes codés sur 8 bits.....	48
Figure 2.14 Opérateur de mutation appliqué à deux chromosomes codés sur 8 bits.....	49
Figure 2.15 Structure générale d'un algorithme génétique.....	49
Figure 2.16 Structure d'un neurone biologique.....	51
Figure 2.17 le neurone formel de Mac Culloch et Pitts.....	52
Figure 2.18 Quelques formes de réseau de neurones, (a) Réseau à couches, (b) Réseau complètement connecté.....	52
Figure 2.19 Les boids de Reynolds.....	54
Figure 2.20 Le voisinage local d'un boid.....	54
Figure 2.21 Les trois règles trouvées par Reynolds pour programmer des mouvements collectifs complexes.....	55
Figure 3.1 Pont Binaire de Deneubourg.....	65
Figure 3.2 Les fourmis suivent indifféremment les deux branches du pont.....	65
Figure 3.3 : Les fourmis ont tendance à emprunter le même chemin (celui d'en bas).....	65
Figure 3.4 Expérience de sélection des branches les plus courtes par une colonie De fourmis : (a) au début de l'expérience, (b) à la fin de l'expérience.....	66
Figure 3.5 : Effet de dépôt d'un obstacle entre la source de nourriture et le nid.....	67
Figure 3.6 Le problème du voyageur de commerce optimisé par l'algorithme AS, les points représentent les villes et l'épaisseur des arêtes la quantité de phéromone déposée. (a) exemple de trajet construit par une fourmi, (b) au début du calcul, tous les chemins sont explorés, (c) le	

chemin le plus court est davantage renforcé que les autres, (d) l'évaporation permet d'éliminer les solutions les moins bonnes.....	69
Figure 3.7. Grille de classification de Lumer et Faieta (d'après Labroche, 2003].....	72
Figure 3.8 : une classification symbolique (prendre la couleur comme une base la (Classification) des isolants.....	74
Figure 4.1 : tableau qui définit l'état d'une fourmi.....	84
Figure 4.2 : les phases de classification.....	86
Figure 4.3 : les phases de classification avec une variation du rayon de détection Avec un rayon R=12 pixels.....	87
Figure 4.4 : le temps de classification en fonction du nombre des fourmis.....	88
Figure 4.5 : le temps de classification en fonction du nombre d'objets.....	88
Figure 4.6 : addition d'autres types d'isolants.....	89
Figure 4.7 : la modification de la taille de l'environnement.....	90

REMERCIEMENTS

Mes remerciement vont tout premièrement à dieu tout puissant pour la volonté, la santé et Patience, qu'il ma donné durant toutes ces années d'études.

Ainsi je dois remercier infiniment mon directeur de mémoire le **Dr. Leila Mokhnache**, pour les précieux conseils, son soutien et son optimisme à tout. Mes remerciements sont lui surtout adressés pour son suivi continuél tout le long de réalisation de ce mémoire avec la compétence que chacun de nous lui reconnaît.

Je remercie, **Monsieur S.Taibi (co-encadreur), M.C Université de Batna**, De son aide. Et aussi Je remercie également, **Monsieur R.abdessemed. M.A Université de Batna**, son expérience et sa rigueur scientifiques m'ont été d'une grande aide dans la progression de ce travail.

Mes sincères remerciements s'adressent à Monsieur le président de jury : **M.Kadjoudj** Maître de conférences à l'université de Batna de m'avoir fait l'honneur de présider le jury

Je tient également à remercier le **Professeur M.Benmohamed**, de l'université de Constantine, **Mr.N.Nait Said** Maître de conférences à l'université de Batna, qui accepté de juger ce travail, ainsi que leur honorable présence.

Je tiens à remercier aussi **Mr CH .Hammouda** directeur de l'unité de recherche de l'université de Batna, d'avoir accepté de m'accueillir dans son unité de recherche, et aussi pour la confiance qu'il a prodigué en ma personne, et pour tous les moyens mis à ma disposition afin d'accomplir mon mémoire de magister.

Mes remerciements aussi aux membres de l'unité de recherche, **B.Yacine, R.Bachir, Malek et C. Mohamed, O.Saida, et Madame Bicha**. qui m'ont beaucoup aidé à m'insérer dans l'unité de recherche.

Mes remerciements vont aussi à mes collègues d'étude, plus particulièrement, **T.youb, B.Sofyne**. Enfin, que toutes les personnes qui ont participé de près ou de loin à la réalisation de ce travail trouvent ici l'expression de mes sincères remerciements.

T. barkat

Dédicace

Je dédie ce modeste travail à :

Mes parents

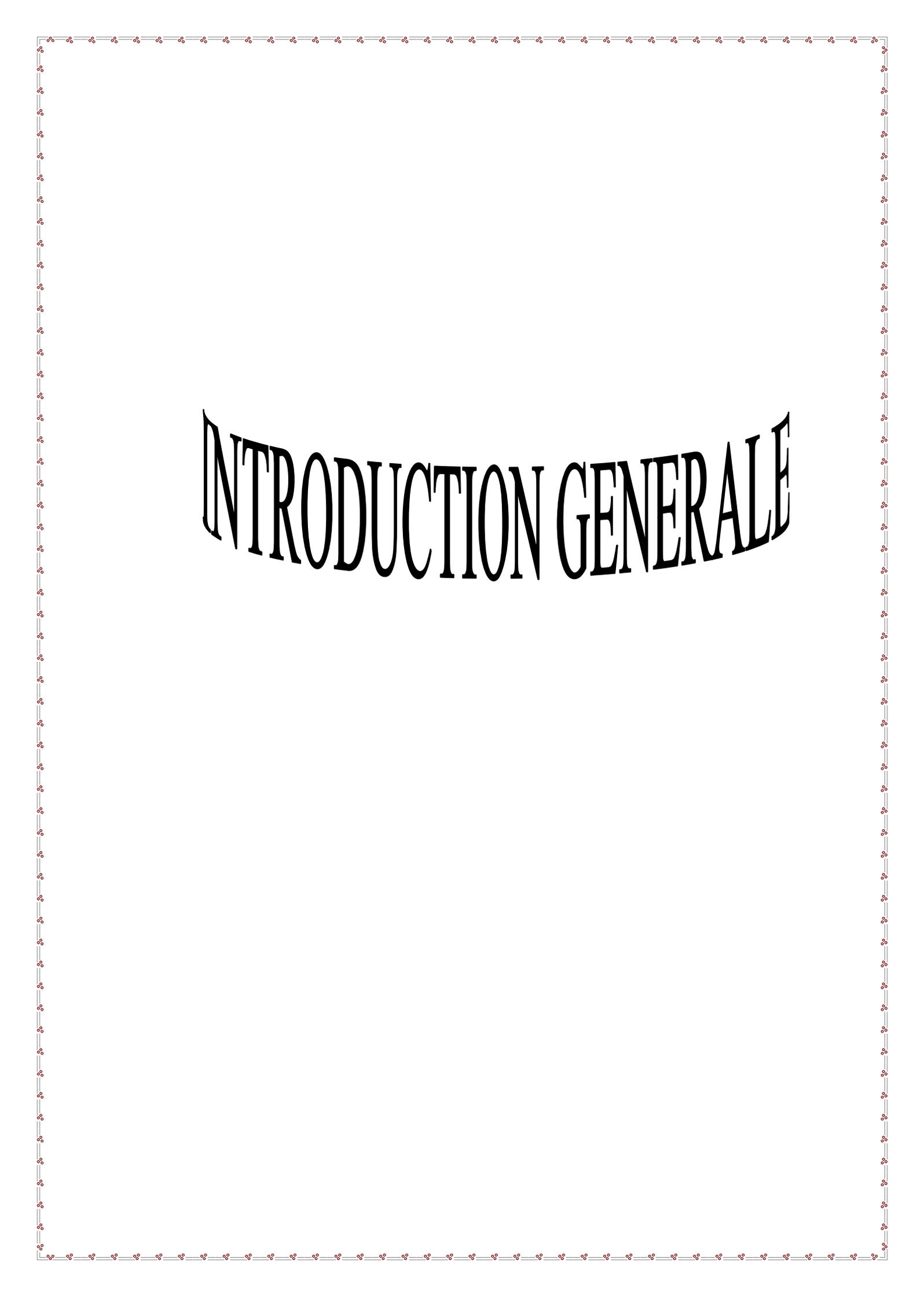
Mes frères

Mes soeurs

Mes amis

Ma promotion

T .Barkat



INTRODUCTION GENERALE

INTRODUCTION GENERALE

Tout ce qui nous entoure, qu'il s'agisse de choses physiques ou abstraites, nous apparaît de façon organisée. Lorsque nous voyons un animal, nous le désignons systématiquement par l'espèce à laquelle il appartient. En effet on attribue les premières recherches théoriques sur la classification, aux besoins exprimés par les biologistes, de spécifier des classes ou espèces animales. Cette classification était censée permettre, étant donnée une description (simple) d'un animal, de l'" étiqueter " par tel ou tel nom d'espèce. Le problème s'est avéré plus complexe qu'il n'y paraissait... [Cle-04]

Les applications se sont multipliées, chacune apportant son lot de problèmes, plus complexes les uns que les autres. Après les biologistes, ce sont les historiens, les médecins, les sociologues ou les didacticiens qui ont manifesté le besoin de classer leurs reliques, leurs patients ou leurs élèves. Aujourd'hui, en nous intéressant aux problèmes issus des technologies de l'information et de la communication, nous sommes amenés à traiter des données très complexes, par leur nature, leur taille, leur provenance, leur diversité. Ces données peuvent être, par exemple, du texte, des images, des séquences vidéo, des bandes sonores, des données numériques ou symboliques, et tout support combinant ces différents types de données. L'envie d'organiser pour simplifier a progressivement évolué vers l'ambition de classer (ou classifier) pour comprendre et, pourquoi pas, pour prédire. Cette évolution a conduit à dégager deux stratégies de classification : supervisée et non-supervisée. La notion de prédiction fait référence à une stratégie particulière d'Apprentissage Automatique (AA), appelée apprentissage supervisé. Ce domaine d'étude à part entière, peut être considéré comme une sous-thématique de l'Intelligence Artificielle (IA) [Cle-04]. De façon synthétique, l'apprentissage supervisé consiste à faire émerger d'un ensemble de données d'entraînement pré-classifiées, les caractéristiques nécessaires et suffisantes pour permettre de classer correctement une nouvelle donnée. Dans ce type d'approche, les classes sont connues à l'avance, cette connaissance est utilisée dans le processus d'apprentissage. La classification non-supervisée correspond davantage au problème initial d'organisation, mentionné ci-dessus, et puise ses fondements théoriques en Analyse de Données (AD) [Cle-04].

En fonction de l'application, la tâche de classification non-supervisée a pour but d'organiser un ensemble d'objets sans connaissance a priori d'une classification, afin de simplifier, ou structurer les données par l'extraction ou la reconnaissance des " formes " récurrentes dans ces données[Fra-06]. On parle, en apprentissage, du problème de la reconnaissance de formes (RF) faisant appel à des méthodes de regroupement et des modèles des agents mobiles dans le but d'accomplir des tâches spécifiques et complexe. Pour mettre en œuvre cette coopération, de

nouvelles approches ont vu le jour cette dernière décennie, [Qui-04] s'inspirant essentiellement du vivant. La colonie des fourmis capable de réaliser des tâches complexes sans qu'une fourmi n'est consciente de l'œuvre globale accomplie. Les systèmes multi agents représentent l'un des modèles de simulations les plus utilisés dans ce cas. Il est à noter que la réactivité des agents ainsi considérés leur permet d'être sensible aux changements de l'environnement dans lequel ils évoluent, et de réagir directement en conséquence, sans passer par une phase de raisonnement [Fra-06]. Il est important de souligner aussi que les robots sont complètement autonomes ; en l'absence d'un agent de contrôle central, les colonies de fourmis par exemple, ont la capacité de travailler ensemble au profit de la société toute entière. Chaque fourmi se comporte selon sa perception locale malgré ceci, des comportements globaux significatifs en résultent. Toutes les fourmis participent à l'émergence d'une tâche à l'échelle macro, en revanche, aucune n'est indispensable à la dynamique globale, ce qui procure une particularité très intéressante à ces systèmes.

Le travail qui a conduit à l'accomplissement de ceci est organisé à travers ce mémoire comme suit :

- * une introduction générale.
- * Dans le premier chapitre, il s'agit à l'étude des propriétés des isolants et les paramètres qui ont une influence sur des phénomènes de vieillissements et de claquage,.....etc. A partir de cette connaissance nous pouvons classer l'état de l'isolant (aider à la décision de sa maintenance).
- * Pour le chapitre deux, il adresse un panorama des différentes approches proposées pour regrouper les données. Nous présentons en particulier les méthodes hiérarchiques, et les méthodes de partitionnement.
- * pour le troisième chapitre, nous présentons une brève introduction au monde des fourmis, ensuite nous décrivons en détail chacun des modèles de fourmis artificielles et nous exposons un algorithme de classification qui permet de nous donner un aperçu général sur la démarche du processus de classification avec les fourmis.
- * Enfin dans le chapitre quatre nous présentons les résultats de la simulation qui en découlent, et nous donnons une description sur l'environnement « netlogo » qui est basé sur la simulation des multi agents.
- * à la fin une conclusion générale clôture le présent manuscrit commentant les différents chapitres et relatant les perspectives de continuation de ce travail.

CHAPITRE I: GENERALITES SUR LES DIELECTRIQUES

I.1 INTRODUCTION

I.2 LES DIELECTRIQUES GAZEUX

I.2.1 Caractérisation des décharges gazeuses

I.2.1.1 Principes de base et phénomènes d'amorçage en tension continue

I.2.1.2 Caractéristiques de la décharge dans les gaz

I.2.3 Influence de différents paramètres sur la rigidité diélectriques dans les gaz

I.2.3.1 Influence de la distribution du champ électrique appliqué

I.2.3.2 Influence du type de tension appliquée

I.3 DIELECTRIQUES LIQUIDES

I.3.1 claquage dans les liquides

I.3.1.1 Mécanisme électronique

I.3.1.2 Mécanisme de claquage avec phase gazeuse

I.3.1.3 Mécanisme de claquage par pont

I.3.2 Influence de différents paramètres sur la rigidité diélectrique

I.3.2.1 humidité

I.3.2.2 Pression hydrostatique

I.3.2.3 Distance entre électrodes

I.3.2.4 Température

I.3.2.5 Nombre de claquages

I.3.2.6 Influence des écrans

I.3.2.7 Durée d'application de la tension

I.4 LES DIELECTRIQUES SOLIDES

I.4.1 Conduction électrique dans les solides

I.4.2 Les pertes diélectriques

I.4.2.1 Permittivité relative.

I.4.2.2 Angle de pertes diélectriques et facteur de dissipation diélectrique

I.4.3 Mécanismes de claquage dans les diélectriques solides

I.4.4 Influence de divers paramètres sur la rigidité diélectrique

I.4.5 Vieillissement des isolants solides sous l'effet du champ

I.4.6 Tenue des matériaux isolants solides aux contraintes thermiques

I.4.7 Vieillissement des isolants solides sous l'effet de l'environnement

I.4.8 Tenue des matériaux isolants solides aux contraintes mécanique

I.5 CONCLUSION

I.1 INTRODUCTION

La connaissance des propriétés électriques des matériaux pour les concepteurs des systèmes électriques ou autres est impérative. En effet, la considération des propriétés électriques est souvent très importante, particulièrement lors du choix des matériaux pour la conception d'un système donné. Le comportement électrique, pour un grand nombre de matériaux, est varié selon l'application et l'environnement de travail auquel il est destiné. Certains matériaux doivent être de très bons conducteurs par contre d'autres, doivent être de très bons isolants (diélectrique). Un diélectrique est caractérisé par ses propriétés électriques, mécaniques, chimiques et thermiques

Les isolants sont utilisés pour :

- Assurer une séparation électrique entre des conducteurs portés à des potentiels différents afin de diriger l'écoulement du courant dans les conducteurs désirés, protection des personnes et des équipements.
- Supporter les éléments d'un réseau électrique et les isoler les uns par rapport aux autres et par rapport à la terre

Ce chapitre est consacré à l'étude des propriétés des isolants et les paramètres qui ont une influence sur les phénomènes de vieillissements et décharge,.....etc.

I.2 LES DIELECTRIQUES GAZEUX

I.2.1 Caractérisation des décharges gazeuses

I.2.1.1 Principe de base et phénomènes d'amorçage en tension continue

Pour obtenir des décharges dans un gaz, il faut appliquer un champ électrique entre les deux électrodes. En présence de champ électrique \vec{E} , les électrons libres présents dans le gaz sont accélérés et deviennent assez énergétiques pour ioniser d'autres atomes, ce qui entraîne la production d'électrons supplémentaires qui sont à leur tour accélérés par le champ électrique \vec{E} , et ainsi de suite. Si la différence de potentiel est suffisamment élevée, c'est-à-dire supérieure à un certain seuil, on aura une cascade de collisions : c'est le phénomène d'avalanche qui conduit à l'ionisation partielle du gaz qui devient conducteur, ce qui forme un plasma. Si l'on introduit un gaz entre une cathode soumise à une tension négative, et l'anode reliée à la masse, la probabilité d'ionisation par unité de longueur X sera proportionnelle au nombre de collisions par unités de longueur multiplié par la probabilité que la collision provoque une ionisation. On peut écrire [S8]..

$$\alpha = \frac{1}{\lambda} \exp\left(-\frac{V_i}{eE\lambda}\right) \quad (1.1)$$

Où λ : est le libre parcours moyen

$eE\lambda$: est l'énergie gagnée par l'électron entre les collisions (en eV)

V_i : est un potentiel effectif d'ionisation qui tient compte des effets des pertes d'ions.

Le terme exponentiel exprime donc la probabilité pour que l'électron provoque une ionisation.

On sait que le libre parcours moyen est inversement proportionnel à la pression p . On peut donc décrire le champ électrique E_d de la décharge au moyen de la formule suivante [S8].

$$E_d = \frac{Ap}{(C + \ln(p.d))}$$

(1.2)

Où pd est le produit pression distance (Torr.cm), A et C sont des constantes qui dépendent du gaz.

La tension de claquage pour une tension continu V_c est en fonction du gaz utilisé de la pression (p) du gaz, et de la distance (d) entre la cathode et l'anode. Cette dépendance exprimée par la relation suivante [S8].

$$V_c = \frac{C_1(p.d)}{C_2 + \ln(p.d)} \quad (1.3)$$

Avec C_1 et C_2 sont des constantes dépendantes de la nature du gaz.

En examinant la formule de la tension de la décharge, on constate que pour les larges installations c'est-à-dire pour des valeurs très élevées du produit pd , la tension de claquage est proportionnelle directement ou inversement à pd . Cette dépendance est nommée loi de Paschen, qui reflète, en fait, la dépendance de la décharge luminescente vis à vis des électrons secondaires.

Si la distance entre les électrodes est petite ou la pression est faible, les électrons secondaires émis de la cathode peuvent arriver à l'anode et provoquer un nombre insuffisant de collisions pour générer un plasma par les électrons secondaires. D'autre part, si la pression est trop élevée, les électrons ne peuvent pas avoir une énergie suffisante entre deux collisions successives pour produire un nombre suffisant d'ions. Si la distance entre la cathode et l'anode est très grande, seulement une petite fraction d'ions produits arrivera à la surface de la cathode et créera des électrons secondaires.

Aux deux points extrêmes de la valeur du produit pd , la probabilité d'ionisation et/ou la collection d'ions est petite et la tension de claquage exigée pour soutenir la décharge est trop élevée. La tension de claquage atteint un minimum entre les deux limites. Ce comportement est

décrit par la courbe de Paschen qui est représentée dans la **figure (1.1)** qui reprend l'évolution de la tension d'amorçage en fonction du produit pd . [S8]

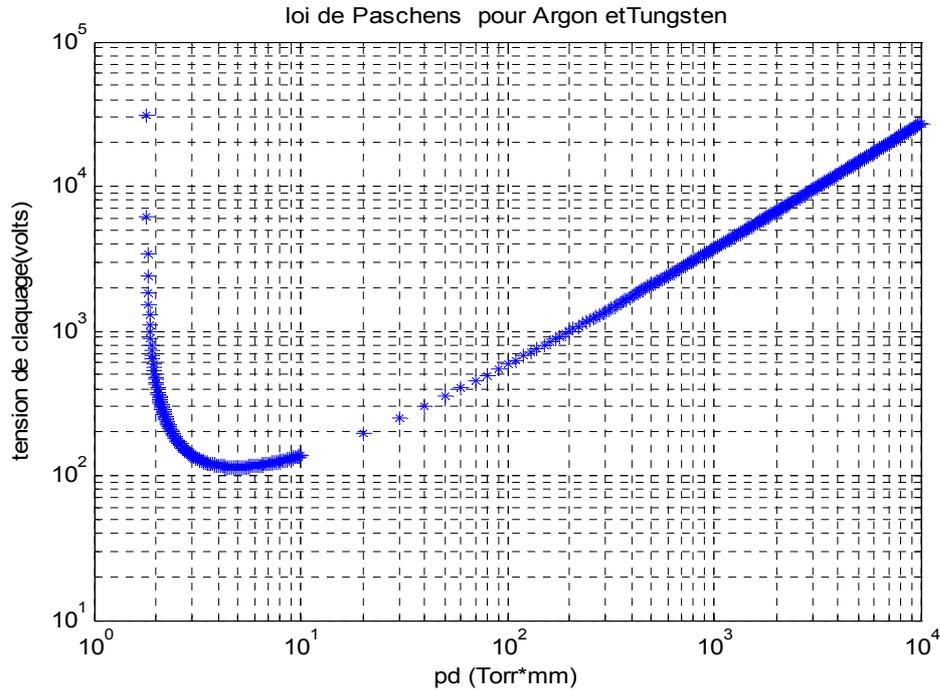


Figure 1.1 : Courbe de Paschen montrant la variation de la tension de claquage en fonction de pd .

Sur la courbe, on constate pour les deux gaz, que la tension de claquage minimale (minimum, de Paschen) est comprise entre 100 et 500 V et se produit pour des valeurs de pd comprises entre 10^{-1} et 10 Torr.cm [S8],[S9].

I.2.1.2 Caractéristiques de la décharge

Le plasma est défini électriquement par une caractéristique tension courant représenté dans (Figure 1.2)

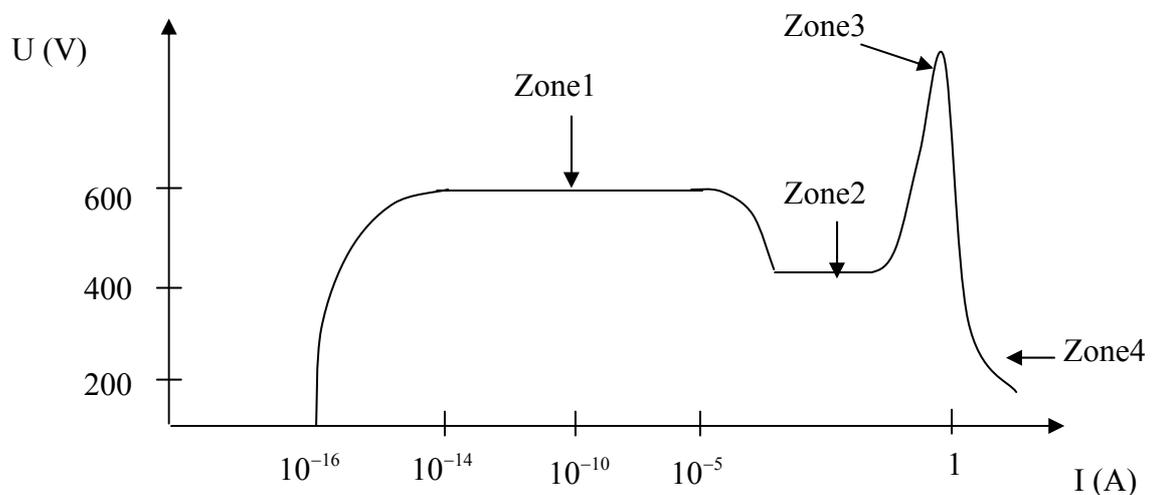


Figure 1.2 : Caractéristiques d'une décharge gazeuse

- (1): décharge de Townsend.
 (2): décharge normale.
 (3): décharge anormale, c.a.d dans c'est zone l'évolution est très rapide, (arc électrique)
 (4): décharge d'arc.

Aux faibles courants, la décharge est dépendante, elle ne s'entretient que lorsqu'elle est exposée à un agent d'ionisation extérieur tels que les rayons cosmiques, la lumière ultraviolette, etc. Si la différence de potentiel entre les électrodes est suffisante, les électrons vont pouvoir créer des paires d'ion-électron, ces derniers pouvant à leur tour ioniser d'autres particules de gaz. C'est l'effet d'avalanches électroniques. Le processus d'avalanche a pour conséquence d'augmenter fortement le courant de décharge, c'est la décharge de Townsend, et la distribution de courant est homogène. Lorsque le courant augmente, la tension entre les électrodes chute et la décharge devient luminescente sur les bords de la cathode, à cause de la désexcitation des atomes de gaz et de la neutralisation des ions. Si le courant croît jusqu'au point où la décharge devient luminescente sur toute la surface de la cathode, la tension entre les électrodes augmente et la décharge devient anormale. C'est ce mode anormal qui est utilisé en pulvérisation cathodique, d'une part en raison de la différence de potentiel élevée entre les électrodes et d'autre part à cause d'un courant de décharge important. Pour des valeurs de courant élevées, la cathode s'échauffe et émet spontanément des électrons qui peuvent provoquer des arcs (instabilité du plasma). Le courant électronique à une position z de l'intervalle inter-électrodes est donné par la relation suivante :

$$dI_e(z) = \alpha I_e(z) dz \quad (1.4)$$

Le courant de Townsend est donné par la relation suivante :

$$I = \frac{I_0 e^{\alpha d}}{1 - \gamma \cdot (e^{\alpha d} - 1)} \quad (1.5)$$

Avec :

I_0 : Le courant électronique à la cathode (A) juste à l'application de la tension.

α : Le 1^{er} facteur de Townsend.

γ : Le coefficient d'émission d'électrons secondaires.

d : La distance entre les deux électrodes (cm).

I.2.3 Influence de différents paramètres sur la rigidité diélectrique

Le vide (ou le gaz) est idéalement le meilleur isolant, mais ceci est affecté par plusieurs facteurs.

I.2.3.1 Influence de la distribution du champ électrique appliqué

Si les électrodes sont des plans parallèles, le champ est homogène. Si son intensité est telle qu'un streamer prend naissance, son développement sera jusqu'à la cathode. La condition de claquage de l'intervalle (d'air ou de gaz) se ramène à la condition d'apparition du streamer.

Si les électrodes sont dissymétriques, par exemple une configuration pointe plan, les avalanches se développeront préférentiellement dans la région de fort champ c'est-à-dire près de la pointe. Si le streamer prend naissance il se propagera tant que le champ électrique total dans la région active permet de reproduire des avalanches secondaires.

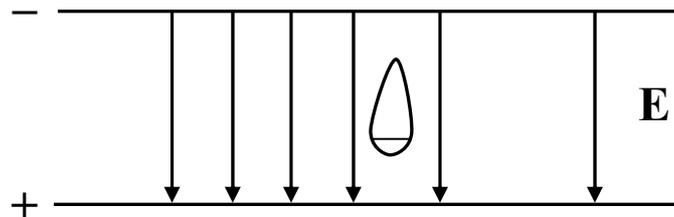


Figure 1.3 : La configuration plan-plan les lignes de champs sont parallèles et le champ est parfaitement homogène dans l'espace inter électrodes. Dès qu'un streamer prend naissance il se propage jusqu'à l'anode

Si la pointe est portée à une tension positive par rapport au plan, les avalanches se développent vers la pointe et le streamer, en sens inverse vers le plan.

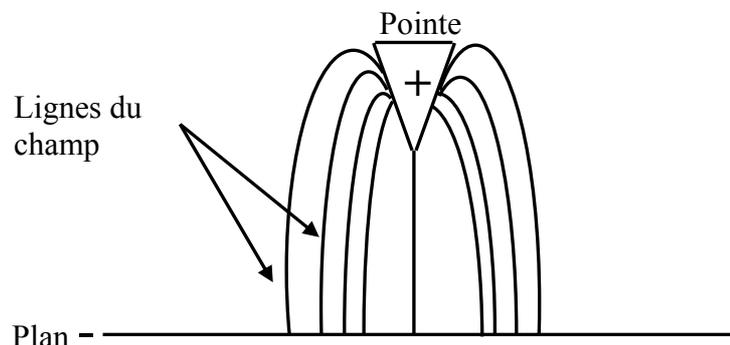


Figure 1.4 : La configuration pointe positive plane à la terre, les lignes de champ montrent la dissymétrie du champ électrique entre électrodes.

Si la pointe est à un potentiel négatif, les avalanches s'éloignent de la pointe et les streamers aussi. Car la tête de l'avalanche dans ce cas est chargée par les électrons qui ont une vitesse plus grande que celle des ions qu'ils ont laissés derrière eux.

I.2.3.2 Influence du type de tension appliquée

La forme de la tension appliquée est importante pour décrire le déclenchement du streamer. Elle peut être continue (un échelon), alternative ou une rampe. Raisonons d'abord dans la configuration pointe positive plan à la masse pour laquelle les mécanismes sont mieux connus.

A- en tension continue

On peut définir un niveau de tension minimum qui correspond à l'apparition du premier streamer. Si ce niveau est appliqué ou même dépassé, un streamer se développe. S'il atteint le plan, un court-circuit limité par le circuit extérieur prend place. Si le champ n'est pas suffisant pour que le streamer traverse tout l'intervalle inter électrode, la partie "passive" a pour effet de réduire le champ au voisinage de l'électrode privant tout phénomène d'ionisation. Les processus de recombinaison et de dérive et de diffusion vont dégager petit à petit le voisinage de la pointe. **[Men-01]** Le champ électrique augmente et les conditions d'initiation d'avalanche et de streamer réapparaissent. Le résultat est une succession de petites décharges streamer à fréquence régulière dépendant du niveau de la tension. Ce streamer porte aussi le nom de "dard".

B- en tension alternative

(50 ou 60 Hz), les processus sont identiques car la fréquence de la tension appliquée est très inférieure à celle des streamers. La fréquence de répétition des couronnes est du domaine de l'audible. C'est précisément les bourdonnements que l'on entend sous les lignes HT. Dans ce cas, les streamers se développent tout autour du câble HT et on parle de "décharge couronne".

Il existe d'autres paramètres qui agissent sur le claquage dans les gazes nous citons quelques uns

- Métal des électrodes.
- Contamination extérieure.
- Température de l'intervalle.
- Pression du gaz.

I.3 DIELECTRIQUES LIQUIDES

I.3.1 claquage dans les liquides

Les phénomènes de décharges dans les diélectriques liquides restent jusqu'à ce jour moins connus par rapport aux gaz et solides malgré l'existence d'une importante quantité d'informations [Bou-03]. Cela est dû principalement à l'intervention d'une multitude de facteurs physico-chimiques, thermiques, électroniques, hydrodynamiques..., lors de l'apparition d'une décharge (phase de génération) ou durant la phase de propagation dénommée "streamers" jusqu'au claquage. De nombreux chercheurs ont tenté d'appliquer aux liquides les théories proposées pour les gaz et les solides. [Bou-03]

Il existe deux principaux mécanismes pouvant être présentés en ce qui concerne le claquage des diélectriques liquides. En plus de ces mécanismes, le claquage peut être dû à la présence d'impuretés de diverses natures en suspension dans le diélectrique liquide. On parle dans ce cas de mécanisme de claquage par pont. Ce dernier mécanisme est le plus proche de celui concernant les huiles de transformateur. Ces derniers sont soumis à différentes contraintes en exploitation, plus particulièrement au vieillissement électrique et au vieillissement thermique en présence d'impuretés de différentes natures nous citons les différents mécanismes de claquage dans les liquides.

I.3.1.1 Mécanisme électronique

Il a été constaté dans le Xénon et l'Argon liquides, que le courant croît de manière exponentielle lorsque la tension appliquée augmente. Cela a été interprété comme le résultat d'un processus d'avalanche.

D'après (Von Hippel) La condition limite d'apparition d'une ionisation électronique des particules du liquide est donnée par: [Bou-03]

$$e.E_c.\lambda = c.h.v \quad (1.6)$$

E_c : Intensité de champ.

λ : l.p.m d'un électron dans le liquide.

$h.v$: Quantum d'énergie perdu par un électron dans les processus non ionisants.

c : Constante liée aux conditions expérimentales.

Dans les hydrocarbures liquides, d'après Lewis, l'électron fournit son énergie principalement dans les processus d'excitation, lors des vibrations des liaisons chimiques C-H. La rigidité diélectrique E_c , peut être formulée par: **[Bou-03]**

$$E_c = \frac{c \cdot h \cdot \nu_{C-H}}{e} \cdot N \cdot \sum n_i \cdot Q_i \quad (1.7)$$

h : Constante de Planck.

ν_{C-H} : Fréquence de vibration des liaisons C-H.

N : Nombre de particules par unité de volume.

n : Nombre de groupes (CH, CH₂, CH₃) dont la section efficace correspond à la collision avec un électron Q_i .

Les pertes d'énergie des électrons sont essentiellement liées aux processus d'excitation lors des vibrations des liaisons C-C. La rigidité diélectrique est donnée par: **[Bou-03]**

$$E_c = \frac{h \cdot \nu_{C-C}}{e} \cdot d \cdot l \cdot (n-1) \cdot N \cdot \frac{\rho}{M} \cdot A \quad (1.8)$$

ν_{C-C} : Fréquence de vibration des liaisons chimiques C-C.

d : distance moyenne entre les axes des molécules.

l : Longueur de la projection des liaisons C-C sur l'axe des molécules.

n : Nombre d'atomes de carbone dans la molécule.

ρ : Densité du liquide.

M : Masse moléculaire du liquide - A: nombre d'Avogadro.

Les théories de Lewis et d'Adamczewski peuvent expliquer la variation expérimentale de la rigidité diélectrique impulsionnelle des hydrocarbures liquides très purs, en fonction de leurs propriétés physiques et chimiques.

Il existe aussi d'autres théories électroniques de claquage des liquides et dont la formulation générale est très proche de la théorie de Townsend pour la disruption des gaz. Ces théories restent encore peu développées.

I.3.1.2 Mécanisme de claquage avec phase gazeuse

Dans les liquides ultra purs, et en géométrie pointe plan, sous tension continue, il a été observé, qu'à partir d'un certain seuil de tension, des bulles gazeuses se forment au voisinage de la pointe, puis sont violemment chassées vers le plan. **[Bou-03]** Cette phase gazeuse résulterait

soit de la vaporisation et de la nucléation des bulles, soit de la cavitation. Dans ce mécanisme il y'a deux sous mécanismes :

- **Mécanisme thermique** : D'après Waston et Sharbaugh, une bulle de vapeur est générée dans le liquide par une injection de courant à partir d'aspérités sur la cathode. Ce courant a la forme suivante

$$I = AV^n \quad (1.9)$$

Avec :

A : cste

$$1.5 \leq n \leq 2$$

V : La tension

- **Mécanisme de cavitation** : Si un liquide est chauffé à pression constant (ébullition), ou s'il est soumis à une dépression à température constante (cavitation), il apparaît des bulles ou cavités de vapeur ou de mélange de vapeur et gaz. La cavitation désigne la séquence de formation, d'expansion de la cavité. Elle peut se produire dans le liquide ou sur les parois. **[Bou-03]**

I.3.1.3 Mécanisme de claquage par pont

Dans les isolants liquides utilisés en technique, peuvent apparaître, en suspension, des impuretés de différentes natures (bulles de gaz, gouttelettes d'eau, particules solides isolantes ou conductrices). La présence d'impuretés conduit au renforcement local du champ électrique. La déformation du champ dépend de plusieurs paramètres comme: forme et dimensions des impuretés, intervalle et forme des électrodes, permittivité et conductivité des impuretés, concentration des impuretés entre les électrodes, valeur des charges libres existant à la surface des particules. Le champ local (E_{local}) est généralement déterminé en attribuant aux impuretés des géométries simples (sphères, ellipsoïde). L'élévation du champ peut conduire à des claquages partiels dans le liquide, dans le cas des impuretés solides conductrices. Dans le cas des impuretés solides isolantes, celles-ci se polarisent dans le champ électrique et tendent à se déplacer vers les zones où le champ est le plus intense ($\epsilon_{\text{impuretés solides}} > \epsilon_{\text{liquide}}$). Si on considère des particules sphériques polarisées, de permittivité ϵ_p , la force agissant sur une particule serait: **[Bou-03]**

$$F = \frac{1}{2} \cdot r^3 \cdot \frac{\epsilon_p - \epsilon_L}{\epsilon_p + 2 \cdot \epsilon_L} \cdot \Delta E^2 \approx \frac{1}{2} \cdot r^3 \cdot \Delta E^2 \quad (1.10)$$

ϵ_p : Permittivité de la particule.

ϵ_L : Permittivité du liquide.

E : Champ électrique.

r : Rayon de la particule.

Pour qu'un pont de particules se forme entre les électrodes, il faut que le temps d'application de la tension soit relativement long. Ce pont nécessite en effet une grande concentration d'impuretés au voisinage des électrodes. Quand ce pont relie les électrodes, il s'établit un fort courant circulant à travers ce pont, qui est plus conducteur que le liquide. On aura ainsi, un échauffement local important, pouvant engendrer le claquage. Le claquage peut aussi avoir lieu durant la formation du pont (champ élevé en tête de pont).

Parmi les théories de claquage par pont, la plus connue est celle de Kok [Bou-03]. Le critère établi par Kok, définit une valeur moyenne du champ électrique, au-dessus de laquelle la vitesse des particules se dirigeant vers le domaine de champ maximum devient supérieure à celle des particules repoussées vers l'extérieur (ce qui augmente la concentration des particules). Cette valeur limite du champ définit selon Kok, la rigidité diélectrique du liquide pendant le claquage par pont:

$$E_c = \frac{1}{2} \sqrt{\frac{k.T}{r^3}} \quad (1.11)$$

k : Constante de Boltzmann.

T : Température du liquide.

r : Rayon des particules impures.

I.3.2 Influence de différents paramètres sur la rigidité diélectrique

Il existe plusieurs facteurs qui ont une influence sur la valeur du champ disruptif dans les liquides. Parmi ces facteurs, les plus importants sont les impuretés et les gaz dissous dans le liquide, la pression hydrostatique, la configuration des électrodes, la température, la durée d'application de la tension et l'état antérieur du liquide [Bou-03].

I.3.2.1 humidité

La présence d'un peu d'humidité dans le liquide diminue considérablement la rigidité diélectrique de ce dernier.

I.3.2.2 Pression hydrostatique

L'augmentation de la pression hydrostatique améliore la rigidité diélectrique du liquide.

I.3.2.3 Distance entre électrodes

En augmentant la distance entre électrodes, la tension de claquage augmente. Dans le cas de champ électrique non uniforme lorsque la tension appliquée dépasse un certain seuil, l'effet de Couronne apparaît (formation de streamers).

I.3.2.4 Température

L'influence de la température est double. En augmentant la température, la concentration de gaz et de l'humidité diminue dans le liquide. Cependant, les conditions pour avoir un claquage thermoélectrique s'améliorent. Cette influence de double effet, fait apparaître un maximum local dans la caractéristique représentant la variation de la tension de claquage en fonction de la température.

I.3.2.5 Nombre de claquages

Il a été observé par plusieurs chercheurs [**Bou-03**], que la tension de claquage de l'huile, dans un système de configuration donnée, augmentait en fonction du nombre d'essais, restait ensuite constante et puis diminuait. Le nombre d'essais variait de 20 à 40. Ceci peut être expliqué par l'élimination des bulles de gaz par les premiers claquages, cependant des claquages répétés et nombreux détériorent le liquide.

I.3.2.6 Influence des écrans

Pour améliorer la rigidité diélectrique d'un système d'isolation, soit qu'on recouvre les électrodes avec une couche mince isolante, soit qu'on introduit une barrière isolante entre les électrodes. Il existe une position optimale de la barrière dans l'intervalle, pour laquelle la rigidité du système est la plus élevée. Dans le cas du système pointe plan, l'augmentation est observée dans les cas de la pointe sous haute tension de polarité positive ou alternative 50Hz, lorsque la barrière est située à proximité de la pointe.

I.3.2.7 Durée d'application de la tension

Si la durée d'application de tension diminue, la tension de claquage augmente. Plus la durée d'application augmente, plus l'effet des différentes impuretés se fait ressentir et la tension de claquage diminue. Pour les temps inférieurs à $10\mu\text{s}$, le claquage a un caractère électrique et pour les longues durées, il a un caractère thermoélectrique.

I.4 LES DIELECTRIQUES SOLIDES

I.4.1 Conduction électrique dans les solides

Les notions de résistance et de résistivité, qui sont généralement bien définies pour les conducteurs, peuvent être extrapolées aux matériaux isolants avec toutefois des spécificités que nous allons étudier. Lorsque l'on applique une tension continue à une éprouvette de matériau isolant, celui-ci n'étant pas parfait est traversé par un courant que l'on peut mesurer. Il est donc possible de déterminer une résistance. Dans certaines conditions, on peut alors calculer une résistivité qui présente théoriquement l'avantage sur la résistance de ne pas dépendre de la géométrie de l'éprouvette et des électrodes utilisées [Men-01]. Rappelons d'abord ce qui se passe pour un conducteur. Un fil métallique de section s et de longueur l est traversé par un courant I lorsqu'on lui applique (avec certaines précautions non détaillées ici) une différence de potentiel V (**figure 1.5**). La résistance R du fil se déduit de la loi d'Ohm :

$$R = \frac{V}{I} \quad (1.12)$$

La résistivité ρ du métal constituant le fil est donnée par la formule :

$$\rho = \frac{R \cdot s}{l} \quad (1.13)$$

ρ Est de l'ordre du ($\mu\Omega \cdot \text{cm}$) pour les métaux les plus conducteurs. La résistivité d'un métal est une caractéristique intrinsèque qui dépend faiblement de son état métallurgique et des paramètres D'environnement à l'exception généralement de la température. Examinons, le cas d'une éprouvette de matériau isolant qui se présente non sous forme d'un fil, mais plutôt sous la forme d'une plaque d'épaisseur e , portant sur ses faces opposées des électrodes de surface S (**figure 1.5**) On retrouve la résistivité de l'isolant en remplaçant la longueur du fil conducteur par l'épaisseur (e) de la plaque isolante et sa section s par la surface (S) des électrodes. La résistance de cette éprouvette vaut donc est donnée par (1.12)

Cette résistance porte en fait le nom de **résistance d'isolement**. On pourrait de la même manière calculer la résistivité de ce matériau en appliquant la formule précédente :

$$\rho = \frac{R \cdot S}{e} \quad (1.14)$$

Cela n'aurait aucun intérêt car, dans ce cas, (ρ) n'est pas une caractéristique intrinsèque du matériau mesuré. On peut s'en apercevoir par exemple en faisant varier les conditions

d'environnement, notamment l'humidité relative de l'air. On constate alors que cette résistivité varie considérablement (plusieurs ordres de grandeur). La raison en est que, dans cette expérience, le courant circule principalement dans un film à la surface du matériau constitué essentiellement d'humidité adsorbée et contenant des impuretés ionisées. Il existe une autre différence majeure, de ce point de vue, entre métal et isolant : il s'agit de l'ordre de grandeur du courant à mesurer dans le cas de l'isolant qui s'exprime en nanoampères, picoampères. La mesure de courants aussi faibles est difficile et nécessite des appareils et des techniques de mesure très spécifiques. Pour les matériaux isolants, on est de ce fait amené à définir plusieurs types de résistances (et de résistivités), par exemple :

- la résistance transversale (ou volumique)
- la résistance superficielle (ou de surface)

En général, ce courant passe essentiellement dans une couche superficielle de l'éprouvette ainsi que dans le dépôt d'humidité et d'impuretés superficielles, mais il comprend également une composante circulant dans le volume du matériau.

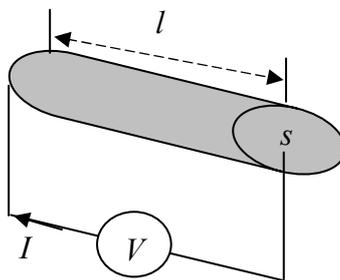


Figure 1.5: Conduction dans un fil de longueur l et de section s

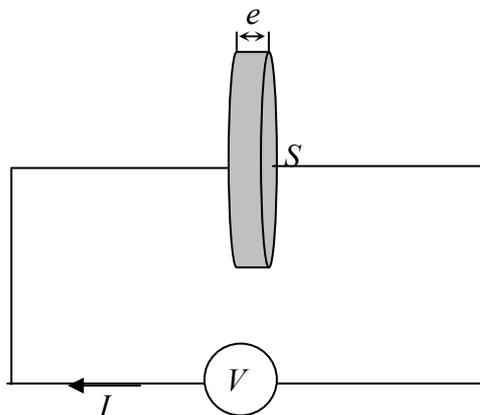
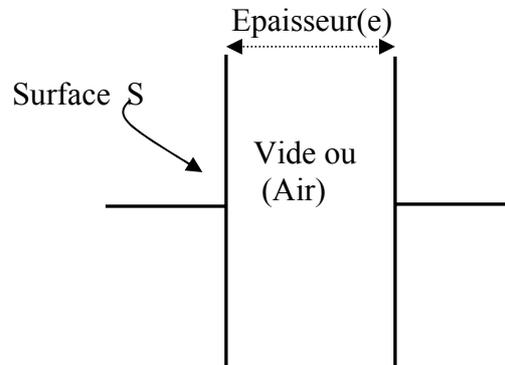


Figure 1.6: Conduction dans une plaque d'épaisseur e et de surface S

I.4.2 Les pertes diélectriques

I.4.2.1 Permittivité relative

Soit un condensateur plan à vide (ou à air) :



Sa capacité est :

$$C_0 = \varepsilon_0 \frac{S}{e} \quad (1.15)$$

Où $\varepsilon_0 = 8.85 \cdot 10^{-12} \text{ F/m}$ est la permittivité absolue du vide (ou de l'air). Si le même condensateur est rempli par un isolant, sa capacité devient :

$$C = \varepsilon_r C_0 = \varepsilon_r \varepsilon_0 \frac{S}{e} \quad (1.16)$$

La **permittivité relative** est définie par le rapport :

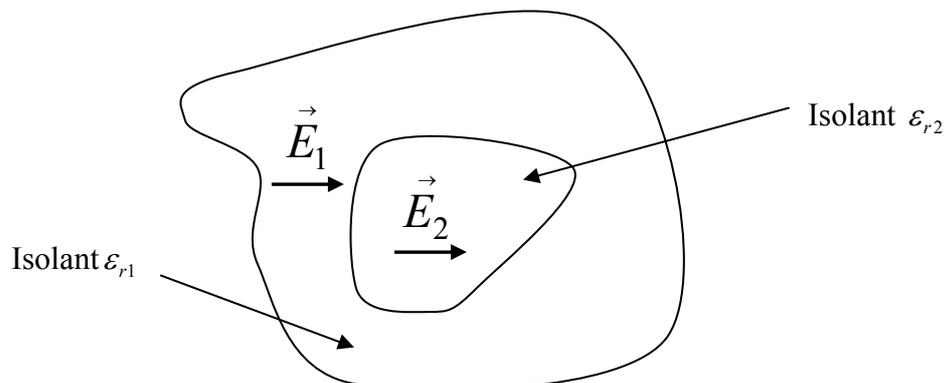
$$\varepsilon_r = \frac{C}{C_0} \quad (1.17)$$

La **permittivité absolue** est :

$$\varepsilon = \varepsilon_0 \varepsilon_r \quad (1.18)$$

Pour l'air ou le vide $\varepsilon_r = 1$. Pour tous les autres isolants $\varepsilon_r > 1$.

Dans un isolant, le champ électrique est inversement proportionnel à la permittivité :



On a la relation suivante :

$$\varepsilon_{r1} \cdot \vec{E}_1 = \varepsilon_{r1} \cdot \vec{E}_2 \quad (1.19)$$

I.4.2.2 Angle de pertes diélectriques et facteur de dissipation diélectrique

A - Circuit équivalent d'une isolation en courant alternatif

Un isolant placé entre deux conducteurs peut être modélisé de manière simplifiée par le circuit équivalent suivant :

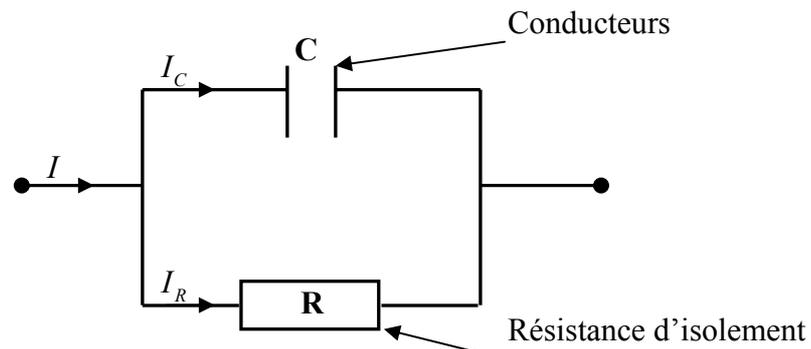


Figure 1.7 : Circuit équivalent d'une isolation

La capacité C représente les conducteurs, et la résistance R est la **résistance d'isolement** de l'isolant, elle est toujours $\geq 10^{12} \Omega$

B - Calcul de l'angle de pertes diélectriques

L'**angle de pertes** δ est défini comme étant l'angle complémentaire du déphasage entre la tension U entre les conducteurs et le **courant de fuite** I traversant l'isolant. [Fou-01]

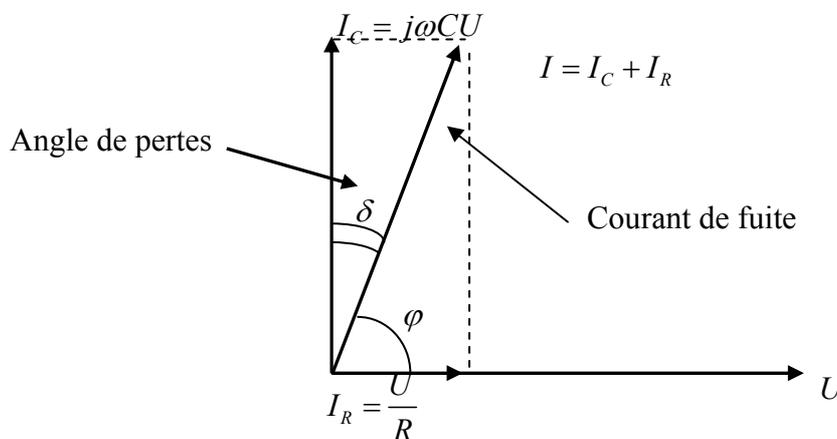


Figure 1.8 : l'angle de pertes

D'après la **figure 1.8** nous avons:

$$\tan \varphi = \frac{CU\omega}{\frac{U}{R}} = RC\omega$$

$$\Rightarrow \tan \delta = \frac{1}{\tan \varphi} = \frac{1}{RC\omega} \quad (1.20)$$

La valeur $\tan \delta$ est appelée **facteur de dissipation diélectrique**. On définit également l'**indice de pertes** :

$$\varepsilon_r'' = \varepsilon_r \tan \delta \quad (1.21)$$

Avec ε_r'' c'est la permittivité complexe.

L'angle de pertes caractérise la qualité d'un isolant :

- bon isolant \rightarrow résistance d'isolement R élevée $\Rightarrow \delta$ faible ;
- mauvais isolant $\rightarrow R$ faible $\Rightarrow \delta$ élevé.

En pratique, le facteur de dissipation $\tan \delta$ varie entre 10^{-2} et 10^{-4} . Il dépend de la fréquence de la tension appliquée selon une loi plus complexe. **[Fou-01]**

C - Calcul des pertes diélectriques

Un isolant est le siège d'un dégagement de chaleur dû aux **pertes diélectriques**. On montre que la puissance par unité de volume des pertes diélectriques est :

$$P_u = \frac{5}{9} \varepsilon_r'' f E^2 10^{-12} \quad (1.22)$$

Avec:

P_u : Pertes diélectriques (en W / cm^3) ;

ε_r'' : Indice de pertes $\varepsilon_r \tan \delta$;

f : Fréquence (Hz) ;

E : Valeur efficace du champ électrique (en V / cm).

I.4.3 Mécanismes de claquage dans les diélectriques solides

Le claquage est l'aboutissement d'une série de processus complexes et interactifs. Assez souvent, l'un des mécanismes est prédominant. On distingue le **claquage intrinsèque** du matériau qui est lié à des phénomènes électroniques, thermiques ou mécaniques et le **claquage pratique** qui survient au voisinage d'un défaut ou par suite d'une dégradation progressive du matériau.

Parmi les types de claquages on cite :

-claquage thermique : la circulation des courants dans le matériau isolant a pour conséquence un transfert d'énergie qui chauffe le matériau. Cet échauffement doit être compensé par les pertes thermiques par conduction dans le matériau pour que l'équilibre soit conservé.

-claquage électronique : deux grandes théories s'opposent dans le domaines du claquages électronique, les théories collectives et la théorie de l'avalanche. Pour Cette dernière elle suppose qu'un électron est présent à l'origine dans le matériau (rayon cosmique par exemple). Cet électron produit une ionisation par collision qui entraîne une avalanche. On estime généralement le nombre d'électrons nécessaires pour produire une avalanche suffisamment destructive à (10^{12}), ce qui correspond 40 générations de collision. [Men-01]

-claquage électromécanique : cette théorie de claquage prend en compte l'effort électrostatique qui tend à rapprocher deux électrodes entre lesquelles est appliquée une différence de potentiel. Cet effort comprime l'isolant solide placé entre deux électrodes. Le claquage survient lorsque cet effort de compression dépasse la résistance mécanique du matériau qui dépend de son module de Young. On rencontre ce type de claquage principalement dans les polymères thermoplastiques [Men-01].

I.4.4 Influence de divers paramètres sur la rigidité diélectrique

Un très grand nombre de paramètres influent sur la rigidité d'un diélectrique solide. Nous n'en citerons ici que quelques-uns.

- tension appliquée
- épaisseur et homogénéité de l'éprouvette
- température
- caractéristiques électriques et thermiques du milieu ambiant

On trouve à l'annexe un tableau qui contient quelques valeurs de rigidité électriques des matériaux isolants.

I.4.5 Vieillessement des isolants solides sous l'effet du champ

L'application d'un champ électrique très élevé sur un isolant conduit au claquage instantané du matériau. Cependant, l'application d'un champ électrique inférieur au champ disruptif peut entraîner une dégradation progressive du matériau qui se termine généralement par un amorçage destructif et irréversible.

Divers types de dégradation peuvent se présenter suivant les conditions d'environnement et les directions relatives du champ électrique par rapport à la surface du matériau. Dans un premier

groupe, la contrainte électrique est relativement faible, appliquée surtout parallèlement à la surface du matériau, et les conditions ambiantes jouent un rôle important par la présence d'humidité ou l'existence d'étincelles ou d'arcs. Il s'agit des phénomènes dits de cheminement. Dans un second groupe, la direction du champ électrique est perpendiculaire à la surface du matériau et la dégradation s'opère sous l'effet de décharges partielles externes ou internes, ou encore d'arborescences. La dégradation d'un isolant solide sous l'action des décharges partielles se produit aussi dans le cas où, à l'intérieur ou à la surface d'une isolation composite, le champ électrique disruptif est localement atteint sur une partie de son trajet. Généralement, ce phénomène affecte la phase gazeuse (ou parfois liquide) dont la permittivité et la rigidité diélectrique sont simultanément plus faibles que celles de l'isolant solide. L'ensemble de l'isolation composite peut ainsi supporter sans claquage un champ électrique élevé mais à chaque alternance (ou périodiquement dans le cas d'une tension continue appliquée) se produisent de petits amorçages qui provoquent une érosion progressive de l'isolant solide. [Men-01]

I.4.6 Tenue des matériaux isolants solides aux contraintes thermiques

L'une des contraintes les plus importantes auxquelles sont soumis les matériaux isolants dans les matériels électriques est, après la contrainte électrique, celle qui résulte des échauffements.

L'effet de la contrainte thermique revêt deux aspects :

- le premier concerne le comportement instantané des isolants aux températures élevées ;
- le second est relatif à la dégradation qui s'opère progressivement au cours du temps et affecte leur comportement à long terme.

Les principales caractéristiques thermiques des isolants sont donc généralement assez différentes suivant la catégorie à laquelle ceux-ci appartiennent. Parmi les caractéristiques thermiques, le coefficient de dilatation thermique linéaire, la conductivité thermique et la capacité thermique.

I.4.7 Vieillessement des isolants solides sous l'effet de l'environnement

Le problème de l'évaluation du vieillissement des matériaux isolants solides sous l'effet des contraintes d'environnement, est traité dans une norme française [Men-01], qui peut également s'appliquer à des matériels de petite taille, met en œuvre l'effet du rayonnement solaire et des intempéries (humidité, pluie, chaleur, froid, sable,...). L'essai consiste à soumettre des éprouvettes du matériau à un cycle combiné de contraintes climatiques.

I.4.8 Tenue des matériaux isolants solides aux contraintes mécaniques

Il est utile de rappeler que, dans de nombreux cas, les défaillances électriques sont initiées par des défaillances d'ordre mécanique [**Men-01**]. Une isolation satisfaisante doit donc être constituée de matériaux dont les caractéristiques mécaniques sont suffisantes pour leur permettre de ne pas être détériorés par leur mise en œuvre puis de supporter sans défaillance les diverses contraintes mécaniques rencontrées en service (efforts dus aux dilatations, vibrations, chocs thermiques, etc.). Il est donc d'une importance capitale de choisir des matériaux isolants dont les caractéristiques mécaniques sont suffisantes pour l'emploi considéré. Parmi les caractéristiques mécaniques les plus importantes pour les matériaux utilisés dans la construction électriques

- Contrainte de rupture et allongement à la traction.
- Contrainte de rupture en compression
- Contrainte de rupture en flexion.
- Résistance au choc.
- Dureté.

I.5 CONCLUSION

Dans ce chapitre nous avons présenté un aperçu général sur les catégories des diélectriques et les divers paramètres influant directement ou indirectement sur l'état d'isolement d'un diélectrique, peut nous aider de faire la sélection des bons isolants à partir de leurs caractéristiques électriques physico-chimiques, mécaniques...

Dans le chapitre qui suit nous allons étudier les différentes méthodes de classification basées sur les caractéristiques propres des matériaux telle que (les propriétés électriques, et les propriétés physiques, chimiques, la couleur, la forme,...)

CHAPITRE II : L'ETAT DE L'ART DE LA CLASSIFICATION ET LES SYSTEMES COMPLEXE

II.1 INTRODUCTION

II.2 LES TROIS PRINCIPALES ETAPES DE LA CLASIFICATION

II.2.1 La préparation des données

- a) Variables et sélections**
- b) Distances et similarités**

II.2.2 Le choix de l'algorithme

- a) La taille des données**
- b) La nature des données**
- c) La forme des clusters**
- d) Le type de résultats attendus**

II.2.3 L'exploitation des clusters

II.3 LA SIMILARITE

II.3.1 Propriétés formelles de base

II.3.2 Similarité et variables numériques

II.3.3 Similarité et variables symboliques

II.4 LES DIFFERENTES METHODES DE CLUSTERING

II.4.1 Le clustering hiérarchique

- a) Les hiérarchies strictes**
- b) Les pseudo- hiérarchies**

II.4.2 Le clustering par partitionnement

- a) Le partitionnement strict**
- b) Pseudo- partitions et partitions floues**

II.5 PHENOMENE D'EMERGENCE ET L'INFORMATIQUE

II.6 CARACTERISATION D'UN PHENOMENE EMERGENT

II.7 L'AUTO- ORGANISATION COMME TECHNIQUE D'EMERGENCE

II.7.1 Définitions

II.7.2 Les mécanismes de l'auto- organisation

II.8 L'INTELLIGENCE EN ESSAIM

II.9 EMERGENCE ET SYSTEMES ARTIFICIELS COLLECTIFS

II.9.1 Algorithmes évolutionnaires

II.9.2 Les algorithmes génétiques

II.9.3 Les réseaux de neurones

II.9.4 Algorithmes à essaims de particules

II.9.5 Les algorithmes de fourmis artificielles

II.10 CONCLUSION

II.1 INTRODUCTION

La classification est un problème central en reconnaissance des formes. Les travaux sont abondants, par exemple pour la reconnaissance de l'écriture manuscrite, la reconnaissance de la parole ou encore l'interprétation de photos aériennes ou médicales. Pour chacune de ces problématiques, il s'agit de détecter certaines caractéristiques pour en extraire des informations exploitables par la suite. Par exemple, pour la reconnaissance de l'écriture manuscrite, pour un mot donné il s'agit de classer les formes de chacune des lettres le constituant dans une des 26 classes formant l'alphabet latin. Une fois cette phase de classification effectuée, pour chacune des formes composant le mot on dispose de la lettre correspondante.

Citons quelques exemples de classification :

-Un exemple courant d'application de la classification est le tri automatique du courrier par un dispositif de lecture et d'interprétation du code postal ou de l'adresse manuscrite. Pour un dispositif d'interprétation du code postal, 10 classes sont possibles (les chiffres de 0 à 9) et les variables descriptives peuvent être les niveaux de gris des pixels, provenant d'une image numérisée du chiffre à identifier.

II.2 LES TROIS PRINCIPALES ETAPES DE LA CLASSIFICATION

Le processus du clustering se divise en trois étapes majeures **figure 2.1** : **(1)** la préparation des données, **(2)** l'algorithme de clustering et **(3)** l'exploitation des résultats de l'algorithme. Nous discutons ici des problématiques générales liées à chacune de ces étapes.

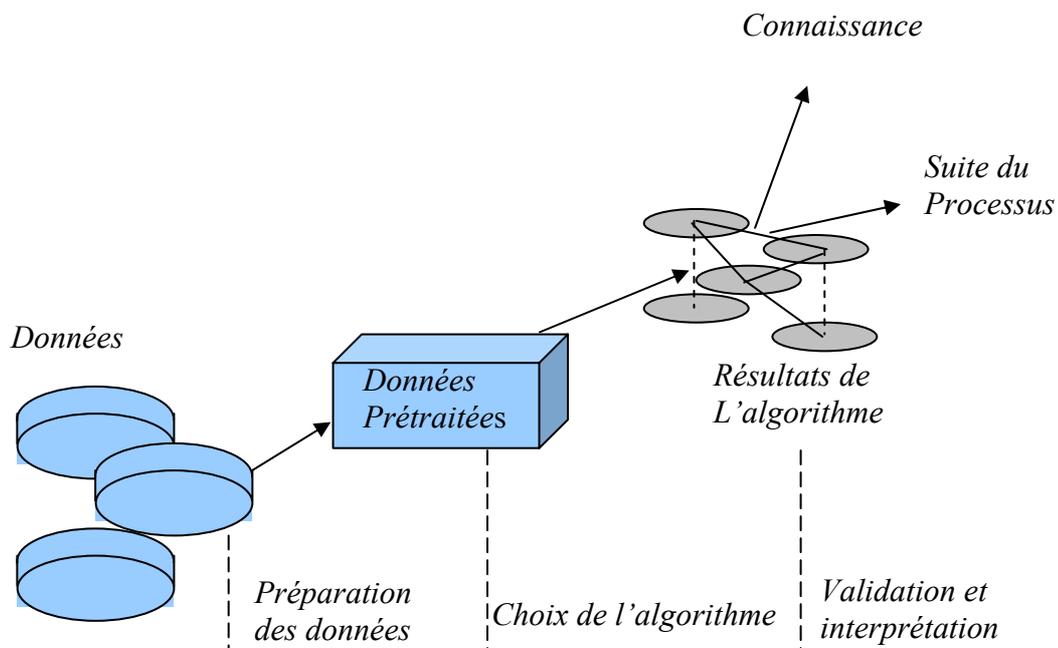


Figure 2.1 Les différentes étapes du processus de clustering

II.2.1 La préparation des données

a) Variables et sélections.[Liu-96]

Les objets sont décrits par des variables $\{v_j\} j=1...p$, aussi appelées attributs, descripteurs ou traits. Ces variables sont de différentes natures :

- variables *quantitatives* : continues (la taille d'une personne), discrètes (le nombre de personnes) ou sous forme d'intervalles (la période de vie d'une personne),
- variables *qualitatives* : non - ordonnées (la couleur des cheveux) ou ordonnées (la taille petit, moyen, grand, etc.),
- variables *structurées*: par exemple la forme d'un objet (polygone, parallélogramme, rectangle, ovale, cercle, etc.) L'étape de préparation consiste à sélectionner et/ou pondérer

ces variables, voire à créer de nouvelles variables, a fin de mieux discriminer entre eux les objets à traiter. En effet les variables ne sont pas nécessairement toutes pertinentes : certaines peuvent être redondantes et d'autres non - pertinentes pour la tâche ciblée. Ce problème de sélection de variables a été largement étudié en classification supervisée mais reste très ouvert pour une approche non- supervisée.

Dans un cadre supervisé, chaque variable peut être évaluée relativement à son pouvoir discriminant par rapport aux classes prédéfinies. Deux types de méthodes se dégagent : les méthodes "filtre" (filter) et les méthodes "enveloppe" (wrapper). Dans le premier cas il s'agit d'enlever les variables "non pertinentes" avant la phase d'apprentissage alors que les approches "enveloppe" utilisent de manière explicite le classifieur pour choisir un sous-ensemble de variables réellement discriminantes. **[Loi-06]**

En revanche, peu de travaux concernent la sélection de variables dans une perspective non supervisée, principalement parce que, sans les étiquettes de classe, il est difficile d'évaluer la pertinence d'une variable. Cette difficulté peut être contournée en effectuant une première étape de clustering à partir de l'ensemble des variables, puis de considérer chaque cluster comme une classe et de réitérer ce processus **[Cle-04]**. De ce fait, cette technique se place parmi les approches "enveloppe" puisqu'elle est fortement dépendante de l'algorithme de clustering et des paramètres utilisés (nombre de clusters, etc.). Récemment, des approches filtres ont été envisagées dans ce contexte non- supervisé.

b) Distances et similarités

La plupart des algorithmes de clustering utilisent une mesure de proximité entre les objets à traiter. Cette notion de proximité est formalisée à l'aide d'une mesure (ou indice) de similarité dissimilarité ou encore par une distance. Le choix ou la construction de cette mesure est déterminant pour la suite du processus. En effet, deux mesures différentes peuvent induire deux schémas de classification différents. Finalement, chaque domaine d'application possèdent ses propres données, il possède également sa propre notion de proximité ; il faut concevoir alors une mesure différente pour chaque domaine d'application, permettant de retranscrire au mieux les différences (entre les objets) qui semblent importantes pour un problème donné. On note par exemple que les données textuelles (mots, documents, etc.) font appel à des indices ou mesures de similarités basées sur des concurrences (coefficient de Dice **[Cle-04]**, indice d'équivalence, etc.). En revanche, les données spatiales sont généralement étudiées dans des espaces métriques en utilisant des mesures de distances traditionnelles (distance Euclidienne, distance de Manhattan, cosinus, etc.). Cependant, ces mêmes mesures peuvent être réutilisées dans d'autres

domaines d'applications à condition qu'elles respectent les contraintes (propriétés) liées aux domaines considérés. Ainsi, il n'est en général pas nécessaire de redéfinir une telle mesure dans tous les cas. Pour être plus précis sur cette notion fondamentale, nous proposons une présentation Formelle de la similarité ainsi que ses enjeux dans le cadre du clustering, en section II.3.

II.2.2 Le choix de l'algorithme

Le choix de l'algorithme de clustering doit donner lieu à une analyse globale du problème : quelle est la nature (qualitative et quantitative) des données ? Quelle est la nature des clusters attendus (nombre, forme, densité, etc.) ? L'algorithme doit alors être choisi de manière à ce que ses caractéristiques répondent convenablement à ces deux dernières questions. Les critères de décision peuvent être : la quantité de données à traiter, la nature de ces données, la forme des clusters souhaités ou encore le type de schéma attendu (pseudo-partition, partition stricte, dendrogramme ou arbre hiérarchique, etc.).

a) La taille des données

La quantité d'objets à traiter est un premier facteur de décision. En effet, pour des données de très grande taille (par exemple en traitement d'images), les algorithmes de complexité plus que linéaires sont quasiment prohibés. Ainsi des méthodes telles que l'algorithme des k-moyennes, proposé en 1967 [Mac-06] ou plus généralement la méthode des nuées dynamiques, étant de complexité linéaire, sont très souvent utilisés. En revanche, lorsque l'on souhaite organiser quelques milliers, voire quelques centaines d'objets, il est possible d'avoir recours à des méthodes plus complexes et nécessitant un temps de traitement plus important (méthodes hiérarchiques ou de partitionnement plus élaborées).

b) La nature des données

Comme nous l'avons précisé, beaucoup d'algorithmes de clustering s'appuient sur une matrice de similarité ou dissimilarité. Le plus souvent, cette matrice est obtenue à partir des descriptions des données. La nature de ces descriptions (variables qualitatives et/ou quantitatives), détermine alors le choix de la mesure de (dis) similarité utilisée.

Par ailleurs, on peut ne pas souhaiter traduire les données dans une telle matrice et conserver la table initiale des descriptions. Certaines méthodes, telles que le clustering conceptuel et notamment l'algorithme ou sa variante numérique, le permettent.

Quand bien même une matrice de similarité existent, certaines méthodes de clustering s'appuient sur la notion d'espace métrique. Pour l'algorithme des k-moyennes par exemple, cet

espace métrique permet de définir de nouveaux objets (ici les centroïdes) absents de l'ensemble initial des données. Dans le cas où un tel espace n'est pas présent, le processus de clustering se base uniquement sur la mesure de similarité. Des variantes sont alors envisageables telles que l'algorithme des nuées dynamiques proposé par Diday [Did-72]. Ou plus simplement les algorithmes " k-médoïdes ". Les nuées dynamiques sont en fait une généralisation de l'algorithme de clustering bien connu des k-moyennes.

c) La forme des clusters

Certaines méthodes de clustering aboutissent à des clusters de formes spécifiques. Par exemple, les méthodes hiérarchiques ou de partitionnement qui consistent à déterminer des centroïdes/ médoïdes aboutissent le plus souvent à des clusters de forme convexe (des hyper- sphères). En revanche, des algorithmes tels que CHAMELEON [Kar-99] ou plus simplement l'algorithme agglomératif hiérarchique du simple lien, construisent des clusters de formes variées. La question est alors de savoir s'il est raisonnable de formuler a priori, une hypothèse sur la forme des clusters. Les variations de taille et densité ainsi que la gestion des " outliers " sont également à prendre en compte dans le choix de l'algorithme de clustering. On entend, par variation de taille, la capacité d'un algorithme à obtenir à la fois des clusters contenant beaucoup d'objets, et des clusters contenant peu voire très peu d'objets. De même, la prise en compte de la densité permet ou non d'obtenir des clusters contenant des objets plus ou moins proches, au sens de la mesure de similarité établie. Enfin, la méthode de clustering peut être ou non sensible à la présence d'outliers.

d) Le type de résultats attendus

La sortie d'un algorithme de clustering peut être, par exemple, une partition (ou pseudo - partition), une fonction ou encore un dendrogramme (arbre hiérarchique). De même, chaque cluster obtenu peut être défini soit par l'ensemble des objets qui le composent, soit par une description relative aux variables initiales. On parle d'une définition en extension, dans le premier cas, et en intension, dans le second. Précisons que la plupart des approches proposent une classification à partir d'une définition en extension des clusters.

Une nouvelle fois, le choix de la méthode de clustering devra être fait en fonction du type de résultat souhaité et donc de l'exploitation envisagée de ce résultat.

II.2.3 L'exploitation des clusters

La tentation est grande, pour un non-spécialiste, de considérer comme acquis le résultat d'un processus de clustering. Autrement dit, les clusters obtenus ne sont généralement ni remis

en cause ni évalués en terme de disposition relative, dispersion, orientation, séparation, densité ou stabilité [Gna-77]. Pourtant, il est sans aucun doute utile de distinguer les classes pertinentes obtenues, des autres. De même, cette étape d'analyse permet d'envisager le recours à une autre approche de clustering plus adaptée. Deux situations sont possibles : soit la tâche de clustering s'inscrit dans un traitement global d'apprentissage, soit les clusters générés par clustering constituent un résultat final.

Dans le premier cas, l'analyse des clusters obtenus (mesures statistiques de qualité) peut aider à orienter le traitement suivant. Une description des clusters n'est pas nécessaire dans cette situation.

En revanche, dans le cas où le clustering constitue à lui seul un processus global de découverte de classes, l'exploitation des clusters pour une application donnée passe par une description de ces derniers. Lorsque les objets se présentent sous la forme d'une matrice de (dis) similarité, il existe peu de méthodes pour décrire les classes (médoïdes, k objets représentatifs et mesures de cohésion). Lorsque les objets sont décrits par un ensemble de variables, on peut avoir recours à des méthodes de description des classes (descriptions conceptuelles).

II.3 LA SIMILARITE

Comme nous l'avons déjà évoqué, la similarité est une notion utilisée dans la plupart des processus de clustering. Les quatre principaux domaines qui étudient la similarité sont : l'Analyse des Données (AD), la reconnaissance des Formes (RF), l'Apprentissage Symbolique (AS) et les Sciences Cognitives (SC). Les trois tâches majeures utilisant la notion de similarité sont : la classification, l'identification et la caractérisation. Nous nous intéressons dans ce travail, aux mesures de similarité / dissimilarité utilisées dans le cadre de la classification non-supervisée, une tâche commune à l'Analyse de Données et à la reconnaissance des formes.

II.3.1 Propriétés formelles de base

Une mesure de similarité est une application symétrique s de $X * X$ dans R^+ telle que $s(x_i, x_i)$ est maximale et $s(x_i, x_j)$ est d'autant plus élevée que les descriptions des objets x_i et x_j sont similaires. De façon opposée, on peut définir une mesure de dissimilarité, dont la nature mathématique peut différer suivant les propriétés vérifiées par cette mesure. Les propriétés de minimalité, symétrie, identité, inégalité triangulaire et ultramétrique permettent de définir les notions d'indice de dissimilarité ou de distance, de distance simple ou ultramétrique et enfin

d'écart simple ou ultramétrique. Nous définissons rapidement ces propriétés puis présentons chacune des mesures relativement aux propriétés qu'elles vérifient, dans le **Tableau 2.1**.

Propriété 1.1. (Minimalité) Une mesure de dissimilarité $d : X * X \rightarrow R^+$ vérifié la propriété de minimalité si et seulement si :

$$\forall x_i \in X, d(x_i, x_i) = 0 \quad (2.1)$$

Propriété 1.2. (Symétrie) Une mesure de dissimilarité $d : X * X \rightarrow R^+$ est symétrique si et seulement si :

$$\forall x_i, x_j \in X, d(x_i, x_j) = d(x_j, x_i) \quad (2.2)$$

Propriété 1.3. (Identité) Une mesure de dissimilarité $d : X * X \rightarrow R^+$ vérifié la propriété d'identité si et seulement si :

$$\forall x_i, x_j \in X, d(x_i, x_j) = 0 \Rightarrow x_i = x_j \quad (2.3)$$

Propriété 1.4. (Inégalité triangulaire) Une mesure de dissimilarité $d : X * X \rightarrow R^+$ vérifié l'inégalité triangulaire si et seulement si :

$$\forall x_i, x_j, x_k \in X, d(x_i, x_j) \leq d(x_i, x_k) + d(x_k, x_j) \quad (2.4)$$

Propriété 1.4. (Inégalité ultramétrique) Une mesure de dissimilarité $d : X * X \rightarrow R^+$ vérifié l'inégalité ultramétrique si et seulement si :

$$\forall x_i, x_j, x_k \in X, d(x_i, x_j) \leq \max \{d(x_i, x_k), d(x_k, x_j)\} \quad (2.5)$$

Type de mesure	Minimalité	Symétrie	identité	Inégalité triangulaire	Inégalité Ultramétriques
Indice de dissimilarité	x	x			
Indice de distance	x	x	x		
Distance	x	x	x	x	
Distance ultra métrique	x	x	x	x	x
Ecart	x	x		x	
Ecart ultra métrique	x	x			x

Tableau 2.1 : Propriétés mathématiques des différentes mesures de disimilarité.

Le plus souvent, on utilise la notion de distance pour évaluer la dissimilarité entre deux objets. Le passage d'une distance à une mesure de similarité peut par exemple être effectué par l'égalité suivante :

$$\forall x_i, x_j \in X, s(x_i, x_j) = d_{\max} - d(x_i, x_j) \quad (2.6)$$

Où d_{\max} correspond à la distance maximum séparant deux objets de X . Dans ce qui suit, nous présentons des mesures de (dis) similarités très utilisées en AD et RF, d'abord lorsque les données sont décrites par des variables numériques, puis lorsque toutes ou partie des variables sont de nature symbolique.

II.3.2 Similarité et variables numériques

La distance la plus connue, et sans aucun doute la plus utilisée, est la distance de Minkowski définie par :

$$d(x_i, x_j) = \left(\sum_{k=1}^p |v_k(x_i) - v_k(x_j)|^l \right)^{\frac{1}{l}} \quad (2.7)$$

Où $v_k(x_i)$ correspond à la valeur de l'objet x_i sur la variable v_k . Selon les valeurs prises par le paramètre l , on parle de distance Euclidienne ($l=2$), de Manhattan ($l=1$) ou de Chebychev ($l=\infty$). Une autre distance très utilisée, notamment dans des applications portant sur les données textuelles, est la distance du cosinus, correspondant au cosinus de l'angle θ formé par les deux vecteurs x_i et x_j :

$$d(x_i, x_j) = \cos(\theta) = \frac{x_i \square x_j}{\|x_i\| \|x_j\|}, \quad (2.8)$$

Où « \square » désigne le produit scalaire et $\|x_i\|$ la norme de x_i ($\sqrt{\sum_k v_k(x_i)^2}$).

Les mesures précédentes supposent l'indépendance entre les variables descriptives (ou attributs) ce qui n'est bien sur pas le cas en réalité pour la plupart des applications. La distance de Mahalanobis permet d'éviter cette hypothèse d'indépendance, elle se définit ainsi :

$$d(x_i, x_j) = \sqrt{(x_i - x_j)^T S^{-1} (x_i - x_j)} \quad (2.9)$$

Où S désigne la matrice de variance/covariance. Notons que pour $S = I$ (matrice identité), on se ramène alors à la distance Euclidienne. Enfin, la pondération des attributs est une étape

importante dans la préparation des données, qui permet d'éviter de considérer davantage les attributs dont les valeurs sont fortement dispersées.

II.3.3 Similarité et variables symboliques

Dans le cas où les données sont décrites (totalement ou partiellement) par des variables symboliques, les mesures précédentes ne conviennent plus, puisque la différence entre deux modalités d'un même attribut n'a plus de sens. Une stratégie pour évaluer la similarité entre deux objets décrits de façon catégorielle est de redéfinir l'espace de description à l'aide d'attributs binaires puis d'effectuer un comptage des propriétés partagées ou non par les deux objets. Les indices de Rand [Reg-99] notés respectivement R et J procèdent de cette façon :

$$R(x_i, x_j) = \frac{d_{++} + d_{--}}{d_{++} + d_{+-} + d_{-+} + d_{--}} \quad (2.10)$$

$$J(x_i, x_j) = \frac{d_{++}}{d_{++} + d_{+-} + d_{-+}} \quad (2.11)$$

Où d_{++} correspond au nombre de propriétés partagées par les deux objets, d_{--} au nombre de propriétés pas vérifiées par aucun des deux objets. Ces deux indices sont plutôt adaptés dans le cas d'attributs initialement binaires. Dans le cas contraire, la définition de nouvelles propriétés (binaires) impliquera une sur-représentation des attributs ayant de nombreuses modalités. L'indice Rand est alors le moins adapté pour cette dernière situation. Notons également que les attributs numériques sont généralement traités par discrétisation (découpage en intervalles). Il découle de ce qui précède, que la redéfinition proposée de l'espace de description ne convient pas dans le cas d'attributs non binaires et/ou numériques.

II.4 LES DIFFERENTES METHODES DE CLUSTERING

Dans cette partie, nous dressons un panorama des différentes méthodes de clustering rencontrées dans la littérature. Il est difficile de proposer une classification « logique » de ces méthodes ; les différentes études de synthèse proposent chacune leur propre organisation [Ber-02]. Présentent successivement les méthodes de clustering hiérarchique, par partitionnement et enfin les approches probabilistes, tandis que [Cle-04] choisit de considérer principalement les approches hiérarchiques et de partitionnement, en incluant les méthodes probabilistes dans cette dernière catégorie. Enfin, [Fun-01], distingue les méthodes de clustering paramétriques et non paramétriques. Ces différentes présentations des méthodes de clustering sont dues, d'une part, au fait que les classes d'algorithmes se recouvrent (certaines méthodes s'appuyant, par exemple, sur

des modèles probabilistes proposant un partitionnement) et d'autre part, différent selon que l'on s'intéresse plutôt aux résultats du clustering (hiérarchie . partitionnement, clustering , Clustering flou etc.), ou à la méthode utilisée pour parvenir à ce résultat (utilisation des fonctions probabilistes en fonction utilisation des graphes, etc.).

Il est possible de regrouper les méthodes de classification sous la forme d'une hiérarchie de méthodes appelée taxonomie. Nous présentons ci-après la taxonomie inspirée de celle de Jain et Al. Dans [Jain-88] (voir **Figure 2.2**).

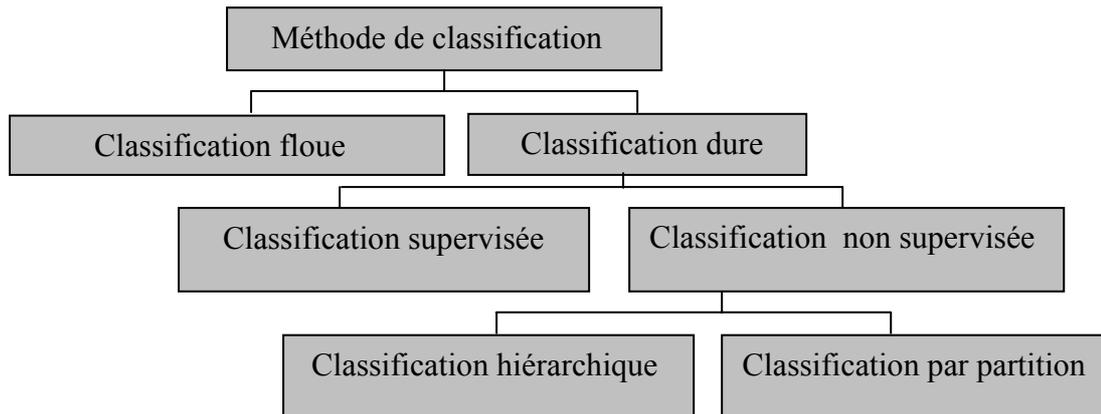


Figure 2.2 Taxonomie de la classification inspirée de celle de Jain et Al.

Les méthodes de classification peuvent être divisées en méthodes dures et méthodes floues. Dans une méthode de classification dure, en traitement d'image un pixel ne peut être affecté qu'à une seule classe dans la partition de l'image. Dans une méthode de classification floue, on affecte au pixel un degré d'appartenance pour chacune des classes de la partition qui indique la probabilité que le pixel y appartienne. La classe finale du pixel sera celle pour laquelle son degré d'appartenance est le plus élevé.

Dans la classification supervisée, le nombre de classes est connu et on dispose d'un ensemble de pixels déjà étiquetés, servant d'ensemble d'apprentissage. Il s'agit alors de pouvoir associer chaque nouveau pixel à la classe la plus adaptée en se servant des pixels déjà étiquetés. Dans la classification non supervisée, aucune information sur le nombre et le contenu des classes possible n'est fournie. L'objectif est alors de pouvoir regrouper automatiquement des pixels considérés similaires dans une même classe. Dans ce cas il s'agira de définir une fonction de similarité entre pixels qui sera maximum entre les pixels d'une même classe et minimum avec ceux des autres classes.

II.4.1 Le clustering hiérarchique

Le principe des algorithmes hiérarchiques est de construire un arbre de clusters (ou dendrogramme) tel que présenté à la **figure 2.3** :

- la racine de l'arbre est formée par le cluster X contenant l'ensemble des objets,
- chaque noeud de l'arbre constitue un cluster $C_i \in X$,
- les feuilles de l'arbre correspondent aux $\{x_1\}, \dots, \{x_n\}$ objets de noeud, pour l'exemple $n=6$
- l'union des objets contenus dans les fils d'un noeud donné, correspond aux objets présents dans ce noeud,
- les « paliers » ou les étages indicés relativement à l'ordre de construction.

Ces hiérarchies sont généralement appréciées puisqu'elles permettent une visualisation de l'organisation des données et du processus de clustering. A partir de ce dendrogramme, il est possible d'obtenir une partition de X en coupant l'arbre à un niveau l donné. Par exemple, le choix de $l=4$ dans le dendrogramme de la **figure 2.3** renvoie le partitionnement suivant : $C = \{\{x_1, x_2, x_3\}, \{x_4, x_5, x_6\}\}$. c'est à dire nous arrêtons la classification à $l=4$ nous trouvons deux classes par la méthode divisifs [**Joh-67**]. le dernier paramètre l peut être choisi relativement au nombre de clusters désiré ou à l'aide d'une analyse statistique de la qualité des différentes partitions que l'on peut extraire de l'arbre.

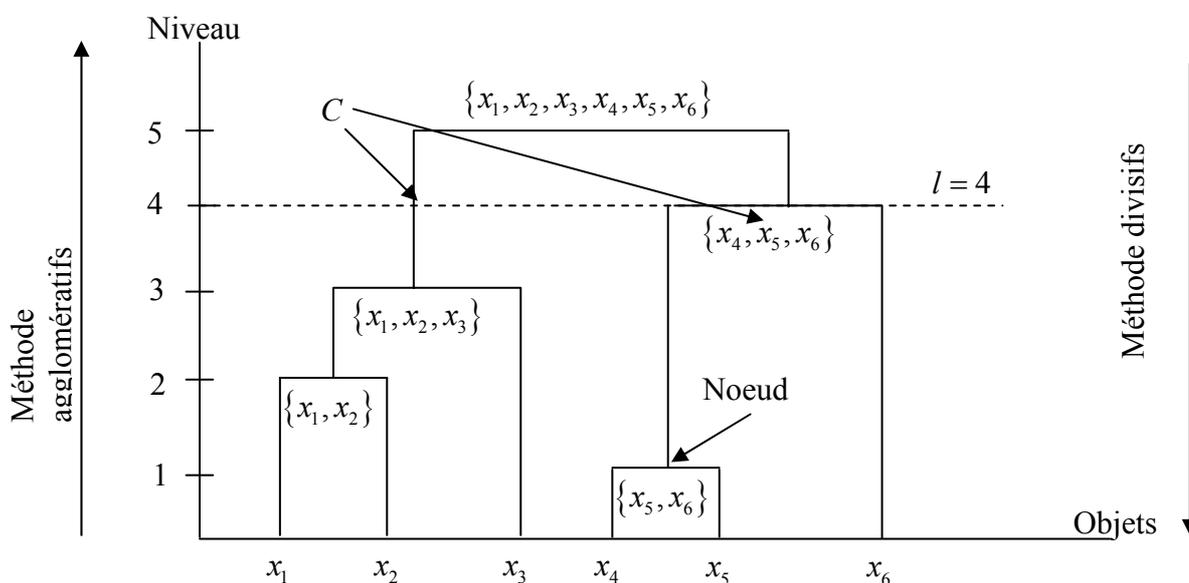


Figure 2.3 : Exemple de dendrogramme.

On distingue deux approches pour parvenir à un tel arbre hiérarchique : les algorithmes agglomératifs et divisifs. Un regroupement agglomératif construit l'arbre en partant des « feuilles » et procède par fusions successives des plus proches clusters jusqu' à obtenir un

unique cluster « racine », contenant l'ensemble des objets. Par opposition, les algorithmes divisifs considèrent d'abord la « racine » contenant tous les objets, puis procèdent par divisions successives de chaque noeud jusqu' à obtenir des singletons. Notons que pour chacune de ces deux méthodes, l'arbre hiérarchique n'est pas nécessairement construit totalement. Le processus peut être stoppé lorsque le nombre de clusters désiré est atteint ou lorsqu'un seuil de qualité est dépassé (distance maximale de fusion).

a) Les hiérarchies strictes

Nous présentons ici les principaux algorithmes de clustering par construction d'hiérarchies strictes. Il existe très peu d'algorithmes divisifs, notamment à cause de la difficulté à définir un critère de séparation d'un cluster. En effet pour un cluster de taille n , il y a $2^{n-1} - 1$ possibilités pour diviser ce cluster en deux sous-clusters. Dans le cas agglomératif, chaque fusion de 2 clusters parmi n , offre seulement $\frac{n(n-1)}{n}$ possibilités. Pour éviter d'explorer toutes les possibilités de divisions, l'algorithme DIANA, présenté en **figure 2.4**, recherche d'abord l'objet le plus « atypique » du cluster avant de lui agréger éventuellement d'autres objets proches de façon à distinguer deux sous-clusters.

Entrée : une matrice de similarité S sur l'ensemble des objets à traiter X

Sortie : une hiérarchie P

1. Initialisation à 1 cluster (racine), $c = \{\{x_1, \dots, x_n\}\}$, et $p = c$
2. Sélectionner le cluster $C \in c$ de diamètre maximum
3. Identifier dans C l'objet (ou l'un des objets) x^* ayant la plus faible similarité

$$x^* = \arg \min \frac{1}{|C|-1} \sum_{j \neq i} s(x_i, x_j)$$

x^* Initialise un nouveau cluster C^* ,

4. Pour chaque objet $x_i \notin C^*$ calculer :

$$s_i = \left[\text{moyenne des } s(x_i, x_j), x_j \in C / C^* \right] - \left[\text{moyenne des } s(x_i, x_j), x_j \in C^* \right]$$

5. Soit x_k l'objet pour lequel s_k est minimal. Si s_k est négatif alors ajouter x_k à C^*
6. Répéter les étapes 3 et 4 jusqu'à $s_k \geq 0$
7. Remplacer C par C / C^* et C^* dans c puis ajouter C / C^* et C^* dans P
8. Répéter les étapes 2 et 7 jusqu'à ce que chaque cluster de c soit réduit à un singleton
9. Retourner P , un ensemble de parties non vides sur X , correspondant aux nœuds de la hiérarchie

Figure 2.4 Algorithme divisif hiérarchique DIANA.

b) Les pseudo- hiérarchies :

Les « pyramides », introduites par E. Diday [Did-72], constituent une généralisation des approches hiérarchiques précédentes. La motivation première est la construction de « classes recouvrantes » avec la contrainte de pouvoir visualiser l'organisation des données. Les pyramides constituent alors ce que l'on appelle aussi des pseudo- hiérarchies. Une pyramide telle que nous la présentons à la **figure 2.5** autorise chaque noeud à avoir jusqu'à deux prédécesseurs. Lorsque l'on coupe la pyramide à un niveau l donné, les clusters obtenus forment alors une pseudo partition de X . Dans la **figure 2.5**, en choisissant de couper le dendrogramme au niveau de la droite pointillés, la pseudo partition $c = \{\{x_1, x_2, x_3, x_4\}, \{x_3, x_4, x_5\}, \{x_6\}\}$ est obtenue.

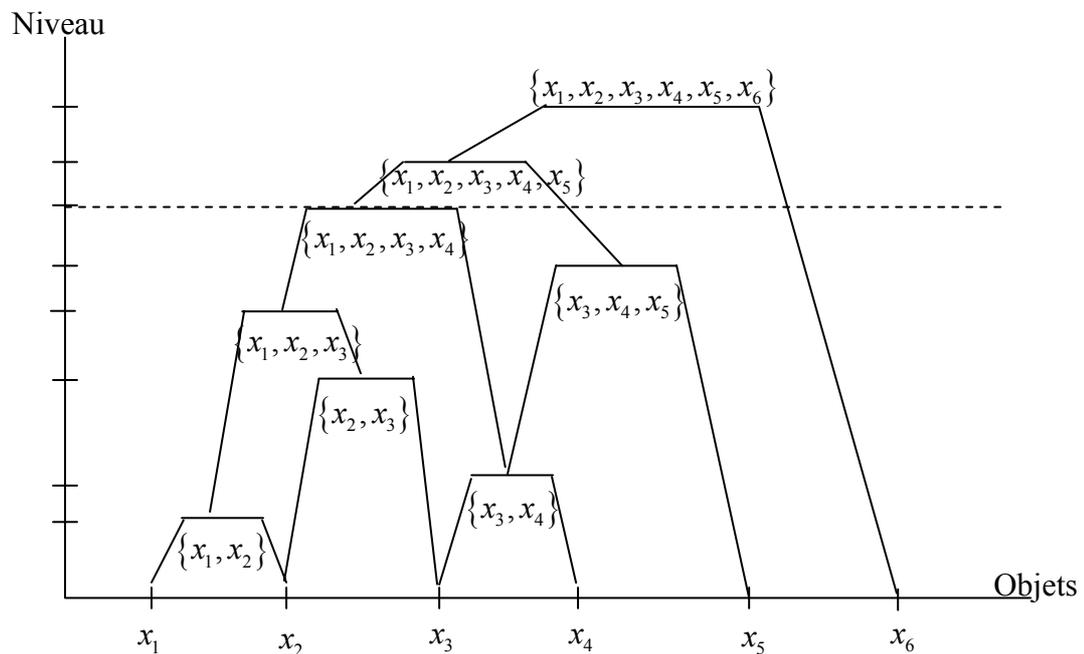


Figure 2.5 Exemple de pyramide

L'algorithme CAP (Classification Ascendante Pyramidale) a été proposé pour construire une pyramide ; nous présentons le principe global de cet algorithme dans la **figure 2.6**. Dans l'algorithme CAP, la condition *i*) assure que chaque noeud de la pyramide aura au plus deux prédécesseurs. Les conditions *ii*) et *iii*) permettent d'éviter les croisements illustrés sur la **figure 2.7**.

Dans le premier exemple proposé sur la **figure 2.7** (pyramide de gauche), la condition *ii*) n'est pas vérifiée puisque C_1 et C_2 sont agrégés alors qu'ils ne sont pas connexes à cause de C_3 . De même, dans le second exemple (pyramide du centre), c'est la condition *iii*) qui cette fois n'est pas

vérifiée puisque $\{x_5\}$ est agrégé à C_3 alors que $C_3 \subset C_5$ et C_3 ne contient ni x_1 , ni x_4 , extrémités de C_5 .

Algorithme CAP : classification ascendante pyramidale

Entrées : une matrice de similarité S sur l'ensemble des objets à traiter X ainsi que $\leq x$, un ordre sur X

Sortie : une pyramide P

1. initialisation à n singletons, $P = \{\{x_1\}, \dots, \{x_n\}\}$
2. on agrège les deux groupes $C_i, C_j \in P$ les deux proches selon S , et vérifiant les conditions suivantes :
 - i) C_i et C_j ont été agrégés au plus une fois,
 - ii) $C_i \cup C_j$ est connexe selon $\leq x$,
 - iii) Soit $C_k \in P$ tel que $C_i \subset C_k$ alors C_i contient des extrémités de la partie connexe C_k
3. ajouter $C_i \cup C_j$ dans P
4. on recommence l'étape 2 jusqu'à ce qu'un groupe de P soit égale à X
5. retourner P , un ensemble de parties non vides sur X , correspondant aux nœuds de la pyramide.

Figure 2.6 Algorithme de construction d'une pyramide.

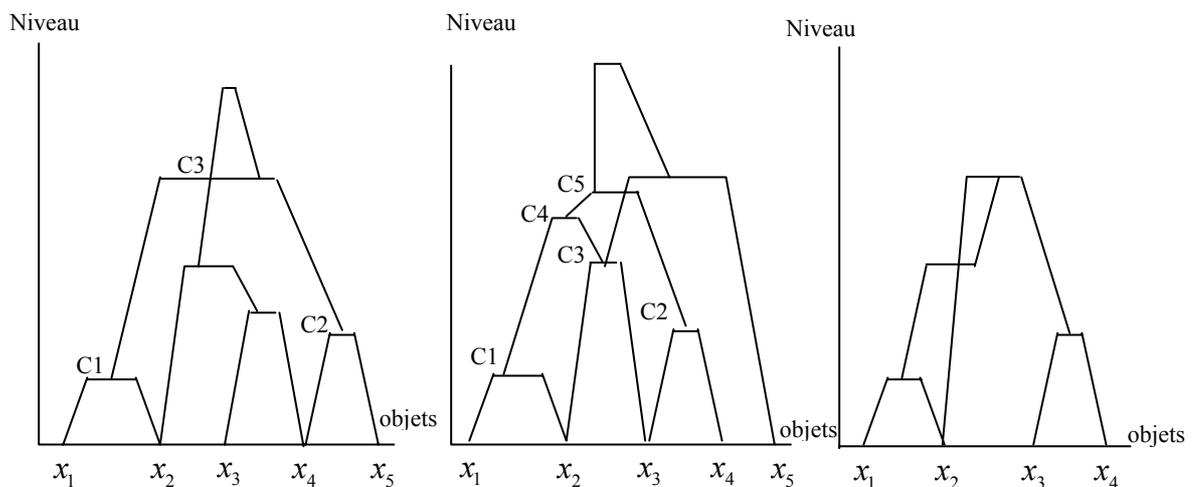


Figure 2.7 Exemples de croisements (gauche et centre) et d'inversions (droite) dans les pyramides.

II.4.2 Le clustering par partitionnement [Cle-04]

Contrairement aux approches hiérarchiques précédentes, les algorithmes de partitionnement proposent, en sortie, une partition de l'espace des objets plutôt qu'une structure organisationnelle du type « dendrogramme ». Le principe est alors de comparer plusieurs schémas de clustering (plusieurs partitionnements) afin de retenir le schéma qui optimise un critère de qualité. En pratique il est impossible de générer tous les schémas de clustering pour des raisons évidentes de complexité. On cherche alors un « bon » schéma correspondant à un optimum (le plus souvent « local ») pour ce critère. Cet optimum est obtenu de façon itérative, en améliorant un schéma initial choisi plus ou moins aléatoirement, par ré-allocation des objets autour de centres mobiles. Nous étudions, dans cette section, les différentes techniques de ré-allocation à partir de l'algorithme bien connu des k-moyennes et dans un souci constant d'évoluer d'une partition stricte vers une pseudo-partition.

a) Le partitionnement strict

L'algorithme des k-moyennes (k-means) [Mac-06] est sans aucun doute la méthode de partitionnement la plus connue et la plus utilisée dans divers domaines d'application. Ce succès est dû au fait que cet algorithme présente un rapport coût- efficacité avantageux. Nous présentons l'algorithme des k-moyennes en **figure 2.8**.

A partir d'un tirage aléatoire de k « graines » dans X , suivies du calcul de la position des nouveaux centres, dits « mobiles ». On peut montrer que, par ce processus, le critère de variance intra-cluster converge vers un minimum local, dépendant de l'initialisation de l'algorithme. Rappelons que ce critère est défini par :

$$V(t) = \sum_{h=1}^k \sum_{i=1}^n d(x_i, x_{ht}^*)^2 \quad (2.13)$$

Où x_{ht}^* désigne le centroïde du cluster C_h à l'étape t .

$V(t)$: le critère de variance intra-cluster.

Une autre version de l'algorithme des k-moyennes consiste à recalculer les centres mobiles au fur et à mesure des ré-allocations. Ainsi, chaque ré-allocation d'un objet à un centre mobile peut engendrer une modification immédiate sur les deux centres concernés : le centre auquel l'objet est nouvellement alloué et le centre auquel l'objet était anciennement alloué. Cette version, dite adaptative, permet, en pratique, de parvenir plus rapidement à une partition de bonne qualité. Cependant le résultat dépend cette fois de l'ordre de traitement des objets.

Algorithme k- moyennes :

Entrées : k le nombre de clusters désiré, d une mesure de dissimilarité sur l'ensemble des objets à traiter X

Sortie : une partition $c = \{C_1, \dots, C_k\}$

Etape 0 : 1. initialisation par tirage aléatoire dans X , de k centres $x_{1,0}^*, \dots, x_{k,0}^*$

2. constitution d'une partition initiale $c_0 = \{C_1, \dots, C_k\}$ par allocation de chaque objet $x_i \in X$ au centre le plus proche :

$$C_l = \left\{ x_i \in X / d(x_i, x_{l,0}^*) = \min_{h=1, \dots, k} d(x_i, x_{h,0}^*) \right\}$$

3. calcul des centroides des k classes obtenues $x_{1,1}^*, \dots, x_{k,1}^*$

Etape t : 4. constitution d'une nouvelle partition $c_t = \{C_1, \dots, C_k\}$ par allocation de chaque objet $x_i \in X$ au centre le plus proche :

$$C_l = \left\{ x_i \in X / d(x_i, x_{l,t}^*) = \min_{h=1, \dots, k} d(x_i, x_{h,t}^*) \right\}$$

5. calcul des centroides des k classes obtenues $x_{1,t+1}^*, \dots, x_{k,t+1}^*$

6. répéter les étapes 4 et 5 tant que des changements s'opèrent d'un schéma C_t à un schéma C_{t+1} ou jusqu'à un nombre τ d'itérations

7. retourner la partition finale C_{finale}

Figure 2.8 Algorithme de partitionnement k-moyennes.

b) Pseudo- partitions et partitions floues [Did-84]

Parmi les méthodes de clustering procédant par ré-allocations, il n'existe pas d'extension permettant directement la construction de pseudo-partitions. Pour obtenir une pseudo-partition, on considère généralement un processus de clustering ou, le schéma obtenu étant ensuite restreint par le biais de seuils d'affectation.

Nous choisissons de présenter à la **figure 2.9** la variante floue de la méthode des k-moyennes.

Algorithme k- moyennes flou :

Entrées : k le nombre de clusters, d une mesure de dissimilarité sur l'ensemble X des objets à traiter, T le nombre maximum d'itérations, un poids $m > 1$ et un seuil $\varepsilon > 0$

Sortie : une partition floue $c = \{C_1, \dots, C_k\}$ définie par les fonctions d'appartenance $\{u_h\} h = 1 \dots k$

1. ($t = 0$) choisir ou tirer aléatoirement une partition initiale $\{u_{h,t}\} h = 1 \dots k$
2. calculer les centre de gravité $x_{1,t}^*, \dots, x_{k,t}^*$ de chacune des k classes (à l'instant t)

$$u(x_{h,t}^*) = \frac{1}{\sum_{x_i \in X} [u_{h,t}(x_i)]^m} \cdot \sum_{x_i \in X} [u_{h,t}(x_i)]^m \cdot u_j(x_j)$$

3. calculer les nouvelles valeurs d'apparence $\{u_{h,t+1}(x_i)\} h = 1 \dots k$ de chaque objets x_i à chaque centre de classe $x_{h,t}^*$

$$u_{h,t+1}(x_i) = \frac{[d(x_i, x_{h,t}^*)]^{2/(1-m)}}{\sum_{h=1}^k [d(x_i, x_{h,t}^*)]^{2/(1-m)}}$$

4. calculer les centres de gravité de chaque classe $x_{1,t+1}^*, \dots, x_{k,t+1}^*$ de chacune des k classes (à l'instant $t+1$)
5. calculer le déplacement global $E_t = \sum_{h=1}^k d(x_{h,t+1}^*, u_{h,t})$
6. si $E_t \leq \varepsilon$ alors retourner la partition flous définie par $(\{x_{h,t+1}^*, \dots, u_{h,t+1}\} h = 1, \dots, k)$, sinon ($t = t + 1$) retourner en 3)

Figure 2.9 Algorithme de partitionnement k-moyennes flou.

L'algorithme du k-moyennes flou est alors fondé sur le principe d'optimisation itérative d'un critère de variabilité intra-classe :

$$j_m = \sum_{i=1}^n \sum_{h=1}^k [u_h(x_i)]^m \cdot d(x_i, x_h^*) \quad (2.14)$$

j_m : est donc une erreur quadratique

u_h : Fonction d'appartenance.

n : Le nombre des données.

m : Indice de flou.

x_k^* : Centres de gravité de la classe.

Le paramètre m est un paramètre appelé fuzzifier permettant d'amplifier les différences entre les objets éloignés et les objets centraux d'une même classe. Ce paramètre peut prendre sa valeur dans $]1, \infty[$. Quand $m \rightarrow 1$ les fonctions d'appartenance prennent leurs valeurs dans l'ensemble binaire $\{0, 1\}$, conduisant ainsi à une partition stricte des données. En revanche, lorsque $m \rightarrow \infty$ les valeurs d'appartenance convergent vers une distribution uniforme $\{u_j(x_i)\}_{j=1\dots k} \rightarrow \frac{1}{k}$ et la partition obtenue est alors trop floue pour y rechercher une quelconque organisation des données. Les paramètres $m = 2$ ou $m = 1.25$ sont généralement choisis. **[Cle-04]**

Plusieurs extensions de cet algorithme ont été proposées afin de considérer des données relationnelles et d'optimiser un critère (ou fonction objective) plus robuste. En ce sens, l'algorithme des k-moyennes ou relationnel (Relational Fuzzy k-Means) **[Kau-90]**, consiste à minimiser itérativement le critère suivant :

$$j_m = \sum_{l=1}^k \frac{\sum_{i=1}^n \sum_{j=1}^n u_l(x_i)^m u_l(x_j)^m \cdot d(x_i, x_j)}{2 \cdot \sum_{j=1}^n u_l(x_j)^m} \quad (2.15)$$

Ce critère correspond à la somme sur chaque classe, de la distance intra-classe moyenne. Dans le cas $m = 2$, le modèle flou se rapproche de l'algorithme FANNY (Fuzzy ANaLYsis). Cependant, ces derniers algorithmes restent coûteux puisque leur complexité est quadratique sur le nombre n d'objets dans X .

Afin de nous introduire dans les systèmes des phénomènes émerger, Il existe d'autres méthodes de classification qui ne sont pas cités dans la phase précédente par exemple :

- Le clustering par mélange de densités de probabilités.
- Le clustering par grilles.
- Le clustering par densités.
- Le clustering conceptuel.

II.5 PHENOMENE D'EMERGENCE ET L'INFORMATIQUE :

L'Émergence : apparition plus ou moins soudaine d'une idée, d'un fait social, politique, économique .D'après Petit Larousse Illustré.

Bien que l'émergence soit encore aujourd'hui l'une des notions les plus floues et les plus discutées, elle est actuellement la plus utilisée pour la conception des systèmes artificiels. Son origine viendrait d'après Ali et Zimmer [Oua-06], du postulat datant de l'antiquité Grèce : "le tout est plus que la somme de ses parties". Ce concept se retrouve aussi dans des écrits de Thalès et Anaximandre et bien plus tard dans le "tout avant les parties" d'Aristote et dans les écrits de J. W. von Goethe fondateur de la "gestalt".

Vers les années 1920, on assistera à l'apparition du mouvement dit proto-émergentisme, auquel participeront de nombreux chercheurs de différentes disciplines ayant pour objectif principal l'étude du phénomène d'émergence afin de reconnaître un phénomène émergent et le différencier de phénomènes explicables grâce à d'autres théories ou modèles [Abd-06]. Ainsi dans le cadre du proto-émergentisme, le phénomène d'émergence est vu comme une boîte noire possédant des entrées de plus bas niveaux et des sorties de plus haut niveau (**figure 2.10**). Ce modèle permet l'identification de tout phénomène émergent mais sans pour autant pouvoir l'expliquer.

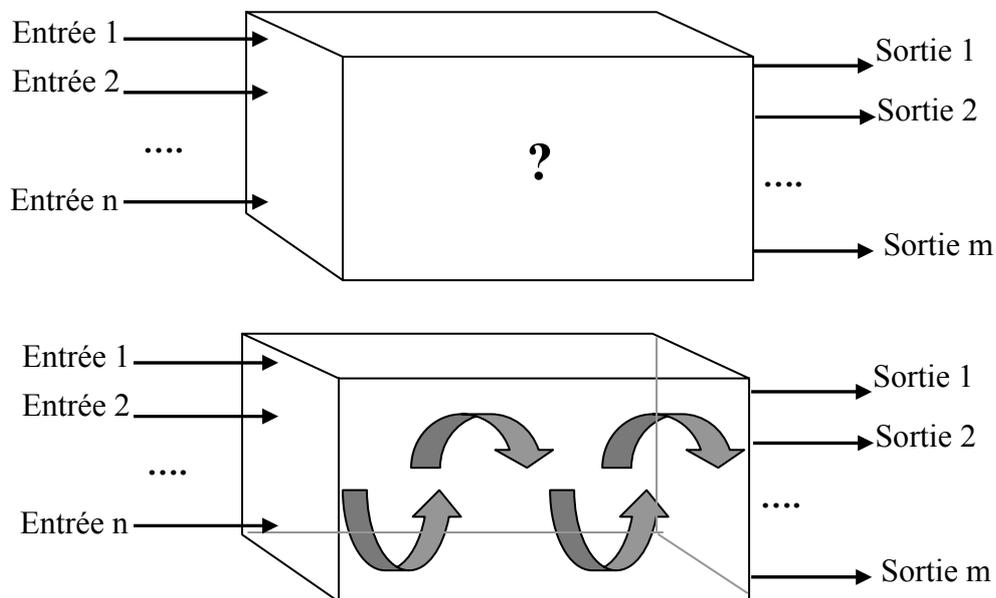


Figure 2.10 Le proto-émergentisme (en haut) et le néo-émergentisme (en bas)

Ce dernier point a donné naissance à un nouveau courant le « **néo-émergentisme** » qui a pour ambition d'explorer cette boîte noire afin de comprendre les processus contrôlant les phénomènes émergents (**Figure 2.10**).

D'une manière intuitive, la notion d'émergence peut être définie comme une propriété macroscopique d'un système qui ne peut pas être déduite à partir de son fonctionnement microscopique. On parle d'émergence quand il y a apparition de structures, et de comportements plus complexes que ceux des entités qui forment le système. Ces comportements sont non programmés explicitement et donc non prévisibles. Cette propriété prise en compte dans les domaines de la biologie, de la thermodynamique est reprise depuis quelques années dans le domaine de l'informatique et a été exploitée pour la conception des systèmes artificiels.

II.6 CARACTERISATION D'UN PHENOMENE EMERGENT

Un phénomène émergent peut être traité selon deux points de vue : un point de vue de l'observateur qui cherche à identifier le phénomène d'émergence et le point de vue du concepteur qui cherche à développer un système qui va produire le phénomène émergent. [OUA-06]

Selon le premier point de vue, l'émergence doit être définie afin de reconnaître un phénomène émergent. Plusieurs définitions de l'émergence ont été proposées dans la littérature. Nous tirons à partir d'elles des propriétés inter-reliées et communes qui permettent d'identifier un phénomène comme émergent:

* **L'observation de nouveauté** : l'émergence signifie qu'il y a apparition au niveau macro d'un phénomène nouveau non observé au niveau micro. Il est alors impossible de décrire, de prédire ou d'expliquer à base des conditions de base définies au niveau micro. [Dré-03]

* **L'observation d'un phénomène ostensible** : il s'impose à l'observateur au niveau macro sans que rien au niveau micro ne permet de le prédire. Autrement dit en terme d'irréductibilité il est impossible de déduire à partir des propriétés de micro -niveau celles de macro -niveau. [Dor-96]

* **L'observation d'une cohérence et d'une corrélation du phénomène** : il a une identité propre mais liée aux parties de micro- niveau qui le produise. Selon Goldstein, l'émergence est définie comme l'apparition de processus d'auto- organisation dans un système complexe [Goldstein, 1999]. Le système doit être capable de modifier de son comportement en fonction de son environnement sans l'aide d'un superviseur [OUA-06]

* **L'observation d'une dynamique** : le phénomène n'est pas connu au départ, s'auto- crée et s'auto- maintient. Langton définit l'émergence en termes de relation de feedback entre les niveaux (macro et micro) dans un système émergent. Autrement dit, les interactions entre les éléments du bas niveau causent celles du haut niveau et les macro -interactions contraignent les micro –interactions.[Ode-01]

Du point de vue concepteur d'un système produisant un phénomène émergent, le système doit présenter les caractéristiques suivantes :

- **La non- linéarité** : La linéarité permet l'interprétation du comportement du niveau global à partir des activités du niveau local. Pour qu'il y soit émergence, il faut donc que les interactions entre les éléments du système soient non linéaires. Ceci signifie que tout composant du système peut être influencée de manière indirecte par d'autres composants du système (qui ne lui sont pas directement liés).

- **Un état proche de l'équilibre** : Au lieu de s'intéresser aux points qui conduisent à l'équilibre du système, on s'intéresse au voisinage de ces points, comme en théorie de la complexité. En de tels points, l'apparition de phénomènes non prévisibles explique le caractère inattendu de l'émergence [S2].

- **L'auto- organisation** : Un système émergent est un système capable de s'auto -organiser et de changer de comportement en fonction de son environnement.

- **Les attracteurs** : Contrairement aux premiers systèmes pour lesquels il n'existait qu'un seul attracteur valide qui menait à l'équilibre, les systèmes émergents possèdent différents types : le point fixe, le cycle limite et l'attracteur étrange. Ces attracteurs ne sont pas pré- donnés et ne dictent pas au système l'état à atteindre, mais plutôt les moyens de passer d'état en état.

II.7 L'AUTO- ORGANISATION COMME TECHNIQUE D'EMERGENCE :

Depuis quelques années l'utilisation du concept d'émergence pour la conception de systèmes complexes s'est très vite répandue. En effet, de tels systèmes ne permettent pas d'appliquer l'approche cartésienne classique pour laquelle la tâche globale est décomposée en sous- tâches car les étapes de résolution ne peuvent pas être programmées explicitement au départ. L'approche émergentiste se révèle donc comme un moyen de passage entre l'activité du "micro-niveau" (les interactions locales entre les composants du système) et celui du "macro- niveau" (le comportement global). Tenant compte des définitions présentées dans les paragraphes

précédents, il apparaît que l'auto- organisation est un élément essentiel pour l'obtention d'un phénomène émergent. Dans ce qui suit, nous allons définir le terme « auto- organisation » ainsi que son principe.

II.7.1 Définitions

Depuis son apparition dans les domaines de biologie, de chimie et de physique, l'utilisation du terme « auto- organisation » s'est largement répandue ces dernières années pour la conception des systèmes informatiques artificiels. Plusieurs définitions du concept d'auto- organisation existent dans la littérature. Nous pouvons en citer les suivantes :

Définition1: « L'auto- organisation est une description d'un comportement, elle a une valeur heuristique et elle permet d'indiquer un phénomène. Elle est condamnée à rester une simple description, tant qu'on ne se préoccupe pas de rechercher le mécanisme qui est à son origine » [Var-88].

Définition2 : « Un système auto- organisateur est un système qui change sa structure de base en fonction de son expérience et de son environnement. » [Dré-03]

Définition3 : «L'auto- organisation est un processus où l'organisation (contrainte redondante) d'un système croît de manière spontanée, sans que cet accroissement soit contrôlé par l'environnement ou ce qui l'entoure ou encore un système externe. »

Définition4 : « L'auto- organisation est définie par un ensemble de mécanismes dynamiques, permettant à des structures d'apparaître au niveau global d'un système (niveau macro) à partir des interactions de ses composants de plus bas- niveau (niveau micro). Les règles spécifiant les interactions entre les composants du système, sont exécutées sur la base d'informations purement locales (niveau micro), et ne font en aucun cas référence à la configuration globale (du niveau macro). Cette configuration globale est une propriété émergente du système plutôt qu'une propriété imposée au système, sous l'influence d'une commande extérieure ». [Bon-97]

Définition5 : « L'auto- organisation est définie comme un processus dans lequel un modèle de niveau global émerge uniquement d'un grand nombre d'interactions entre les comportements de bas niveau du système. De plus les règles spécifiant les interactions entre composantes du système sont suivies en utilisant uniquement des informations locales sans références au modèle global ».

Toutes ses définitions font référence aux mêmes concepts : structuration, organisation, interaction, autonomie et enfin émergence d'un comportement global à partir de plusieurs

comportements locaux. L'auto- organisation peut être alors définie comme un moyen permettant à un système de se structurer et de se maintenir sans aucune intervention de l'extérieur. Chaque composant du système réagit au stimulus par des règles locales simples et modifie ainsi son environnement et donc le comportement des autres composants (par exemple le dépôt de traces de phéromones chez les fourmis). De ce processus émerge une intelligence collective qui permet au système de réaliser des tâches difficiles voire complexes non explicites. On voit là que le concept d'émergence est fortement lié à celui de l'auto- organisation.

II.7.2 Les mécanismes de l'auto- organisation :

Devant la complexité des structures obtenues par le phénomène d'auto- organisation, on se demande comment ces structures émergent-elles sans un plan prédéfini. En fait leur apparition est due à de nombreuses interactions entre les éléments du système ainsi qu'avec l'environnement externe. Nous pouvons distinguer en particulier les principaux facteurs suivants :

- **La rétroaction** : pendant son exécution, les composants du système interagissent ensemble ou avec leur environnement. On considère une forme particulière d'interaction est la **rétroaction** (ou feed-back) qui est obtenue lorsque les résultats (obtenus par transformation des données d'entrée) sont retransmis au système sous la forme de nouvelles données d'entrée. Deux types de rétroaction peuvent alors être observés :

* **La rétroaction positive** : appelé aussi « auto- catalyse », c'est un mécanisme qui permet de renforcer une action impliquant une convergence très rapide. Il est observé quand les résultats agissent de la même manière que les données d'entrée c'est à dire renforcent le processus de transformation. Ce phénomène est observé chez les fourmis lors de la recherche de nourriture. Le cumul de traces de phéromones déposées par chaque fourmi sur le chemin qu'elle emprunte, permet de renforcer ultérieurement le choix de ce chemin par les autres fourmis de la même colonie.

* **La rétroaction négative** : c'est un mécanisme de maintien d'équilibre qui va agir dans le sens inverse de l'amplification des fluctuations du système. Une rétroaction positive peut conduire à une stagnation prématurée et à une explosion du système alors qu'une rétroaction négative le stabilise. Ce phénomène est aussi observé dans le comportement de fourrage des fourmis avec le mécanisme d'évaporation des traces de phéromones. Les chemins les moins choisis par les fourmis seront de moins en moins imprégnés de phéromones. [Den-87]

- **La gestion des flux** : ce sont des moyens de communication entre les composants du système et avec leur environnement. La communication peut être directe par messages ou signaux ou bien indirecte par le biais de modifications de l'environnement. Cette deuxième possibilité de communication a été appelée par le chercheur Pierre-Paul Grassé « stigmergie » à partir des racines stigma qui signifie piqûre, et ergon. Qui veut dire travail ou oeuvre. [Den-87]

En effet Grassé a montré vers la fin des années 1959 que chez les termites [Oua-06], la construction d'une bâtisse est guidée par la construction elle-même et ne dépendait pas seulement des termites bâtisseuses : l'insecte ne dirige pas son travail mais il est guidé par lui. Ainsi, toute nouvelle forme construite, devient un nouveau point de départ matériel pour les autres termites produisant ainsi une nouvelle forme stimulante, qui peut orienter et déclencher en retour une nouvelle activité bâtisseuse chez les autres membres de la colonie. Grâce à la stigmergie, les termites arrivent à s'auto-organiser pour réaliser un travail collectif sans aucune coordination directe.

II.8 L'INTELLIGENCE EN ESSAIM

Les études éthologistes ont pu identifier chez certains animaux et insectes des comportements auto-organisés menant à l'émergence de tâches qualifiées de complexes sans contrôle global [Bon-97]. En effet, ces insectes sociaux sont capables de créer des communautés géantes et réalisent collectivement des tâches extraordinaires telles que le fourragement chez les fourmis, la construction de véritables cathédrales de terre chez les termites, les déplacements collectifs chez les oiseaux et les bancs de poissons. Ce comportement global complexe émerge à partir des comportements locaux et simples de chaque élément du groupe.

Cette capacité qui passent de comportements individuels simples à des comportements collectifs complexes est appelée "intelligence en essaim" ("swarm intelligence") ou « intelligence collective » sous domaine de l'intelligence artificielle distribuée. D'après Eric Bonabeau et Guy Théraulaz dans [Bon-94]. « ces capacités cognitives collectives qui caractérisent les phénomènes d'intelligence collective, sont mises en oeuvre sans régulateur central capable de détecter et de compenser des déviations globales, et sans non plus qu'une connaissance suffisante existe au niveau de chaque élément pour que l'on puisse lui attribuer une perception globale de la situation ». Dans une approche basée sur l'intelligence en essaim, les entités formant l'essaim ont un comportement relativement simple, que l'on ne peut pas qualifier d'intelligent. Cependant ces comportements individuels peuvent faire émerger grâce à des interactions locales inter-individus et avec l'environnement des règles locales un comportement complexe, adaptatif sans aucun

contrôle central. « Dans les sociétés d'insectes, le «projet» global n'est pas programmé explicitement chez les individus, mais émerge de l'enchaînement d'un grand nombre d'interactions élémentaires entre individus, ou entre individus et environnement. Il y a en fait intelligence collective construite à partir de nombreuses simplicités individuelles.». L'intelligence en essaim tente alors de simuler les mécanismes produisant ce type de comportements, afin de proposer de nouvelles techniques pour la résolution de problèmes.

II.9 EMERGENCE ET SYSTEMES ARTIFICIELS COLLECTIFS

Comme nous l'avons souligné auparavant, la méthode classique de résolution de problèmes nécessite de formuler à priori le domaine du problème ainsi que les étapes de la résolution. Chose difficile lorsqu'il s'agit d'un domaine dynamique et incertain. Ces difficultés ont amené les informaticiens à chercher de nouveaux modèles, de nouvelles théories, de nouveaux systèmes pour résoudre de tels problèmes. D'où l'apparition de nouveaux systèmes à fonctionnalité émergente. La particularité de tels systèmes réside dans deux points. Le premier est que le calcul n'est pas explicite, ni programmé à l'avance mais est de nature émergente. On parle alors de « calcul émergent » ou « émergent computation ». Le deuxième point est qu'ils sont constitués d'entités dont le comportement est en général très simple et réactif, cependant ils sont simples capables, au niveau du groupe, de comportements d'apprentissage, d'adaptation aux changements, de robustesse vis-à-vis des cas non prévus. Pour cette raison, ces systèmes sont généralement qualifiés de « systèmes collectifs intelligents ».

La plupart des applications utilisant ce type de systèmes l'ont été dans le domaine de l'optimisation. Il s'agit de maximiser ou de minimiser une fonction objective afin de trouver les meilleures solutions à un problème donné. C'est généralement un problème NP-complet, pour lequel les méthodes exactes classiques ne donnent pas de bons résultats en un temps acceptable d'où la nécessité d'utiliser des heuristiques. Dans la suite, nous allons présenter certains de ses systèmes.

II.9.1 Algorithmes évolutionnaires

En 1859, Charles Darwin évoqua, dans son livre « De l'origine des espèces par voies de sélection Naturelle » [Dar-59], les premiers principes de la théorie de l'évolution. Selon lui l'évolution des systèmes vivants au cours des générations s'opère en deux étapes : la sélection et la reproduction :

- La sélection naturelle est le mécanisme central qui opère au niveau des populations, en entraînant la mort sélective des plus faibles et la survie des individus les mieux adaptés à leur environnement.
- La reproduction fait subir des modifications successives aux individus de la population par mutations et recombinaisons, aboutissant ainsi à une grande diversité entre deux populations d'une même espèce.

De son côté, John Mendel, le fondateur de la génétique, avait effectué des travaux sur l'hérédité, entre 1859 et 1866 [Men-65]. Qui expliquent les lois de transmission des caractères à travers les générations dans le cadre d'une reproduction sexuée. Les caractéristiques héréditaires sont ainsi localisées dans le génome, qui constitue le patrimoine génétique de chaque individu. Le patrimoine génétique est composé de gènes, codant les protéines responsables de l'architecture et du fonctionnement au niveau cellulaire. Les gènes constituent, à ce titre, l'unité sémantique du langage génétique. La reproduction sexuée met en jeu deux mécanismes fondamentaux et complémentaires : le croisement et la mutation génétiques. Le croisement implique la combinaison du patrimoine génétique des deux parents pour former le patrimoine de l'enfant. Il permet ainsi de former des individus distincts, bien que possédant des traits propres à chaque parent. La mutation intervient durant la phase de croisement et se traduit par la modification spontanée de quelques gènes.

La synthèse des ses deux théories augmentée de la découverte de l'ADN a donné naissance au néo-darwinisme qui a inspiré plus tard les chercheurs informaticiens pour le développement des modèles artificiels de l'évolution intégrant des propriétés telles que la mutation et le croisement et leur application au domaine de l'optimisation. Parmi ces modèles on trouve une classe d'algorithmes regroupés sous le nom générique les algorithmes évolutionnaires (AE) apparus dans les années 70.

Comme exemple des AE, les Algorithmes génétiques (AG) sont certainement la branche des AE la plus connue et la plus utilisée de ces techniques qui se différencient dans leurs façons de coder les individus (donc modéliser le problème à résoudre) et par leur façon de faire évoluer la population, mais ces versions sont toutes basées sur les mêmes principes de base.

II.9.2 Les algorithmes génétiques :

Le principe de base est de mimer ces deux mécanismes pour faire évoluer une population de solutions (représentant des individus) afin d'obtenir des solutions de qualité de plus en plus

meilleure. Un algorithme génétique est simple à mettre en oeuvre et se décrit par les points suivants :

- 1) Choix d'un codage approprié pour les individus de la population. Ce codage doit être complet et capable de coder toutes les solutions possibles. Dans un premier temps, Holland a utilisé un codage sous forme de chaîne de bits de longueur fixe afin de maintenir le plus possible l'analogie avec la structure protéinique de l'ADN. Le codage binaire possède l'avantage supplémentaire de fournir un langage quasi-universel, indépendant du problème à traité. Le codage binaire d'un individu est appelé « chromosome » et ses éléments sont appelés « gènes ». Plus récemment, d'autres types de codage sont apparus pour pallier les limitations du codage binaire, trop limitatif ou trop complexe à mettre en oeuvre, tels que le codage entier ou réel. [Gol-89]
- 2) Construction de la population initiale d'individus;
- 3) Association à chaque individu de la population une fonction d'évaluation la fitness qui mesure son degré d'adaptation à l'objectif visé. La (Figure 2.11) donne un exemple de codage d'un chromosome sur 8 bits. L'évaluation de cet individu consiste à transformer la chaîne 0/1 en une valeur réelle.

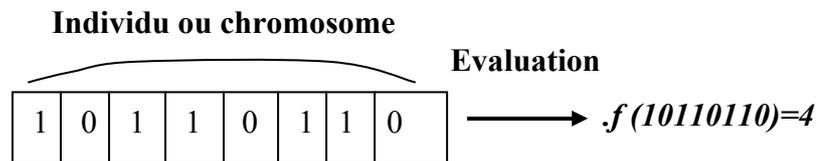


Figure 2.11 Codage de l'information

- 4) Faire évoluer progressivement, par générations successives, la composition de la population, en maintenant sa taille constante. Au cours des générations, l'objectif est d'améliorer globalement la performance des individus par l'application des opérateurs génétiques : sélection, croisement et mutation.

a. La sélection favorise la survie des individus les plus performants et la transmission de leurs caractéristiques à la génération suivante. La sélection peut se faire selon plusieurs procédures, les plus couramment utilisées sont : la sélection par la roue de la fortune, par rang de classement, et par tournoi. La Figure 2.12 représente la sélection par la roulette.

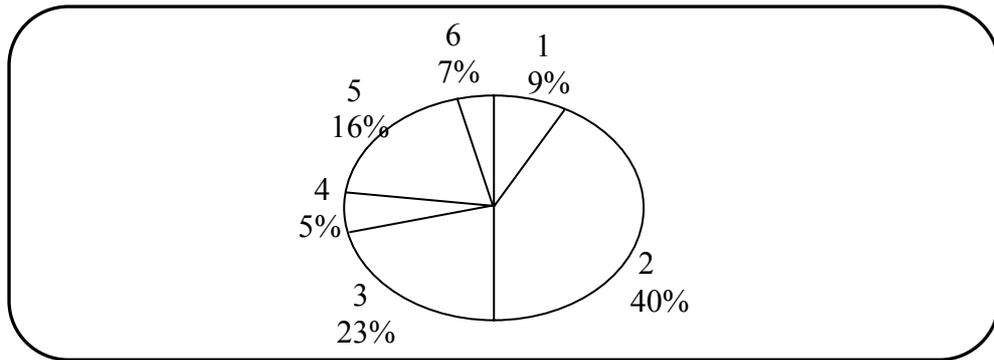


Figure 2.12 Modèle de la roulette.

b. Le croisement et la mutation sont des opérateurs de reproduction qui agissent sur un ou plusieurs chromosomes issus de la phase de sélection en fonction d'une certaine probabilité. Le croisement permet le brassage du patrimoine génétique présent dans la population en produisant de nouveaux chromosomes (enfants) à partir de chromosomes parents. Ainsi chaque chromosome a une probabilité P_c de se croiser avec un autre chromosome pour produire un chromosome nouveau qui sera insérer dans la nouvelle population. Il existe une variété de méthodes pour faire croiser deux chromosomes, les plus généralement utilisées sont le croisement à un ou plusieurs points **Figure 2.13**. D'autres variantes de croisement existent avec un nombre de points qui peut être de deux (trois, voire plus, ou encore le croisement uniforme. [Sys-91]

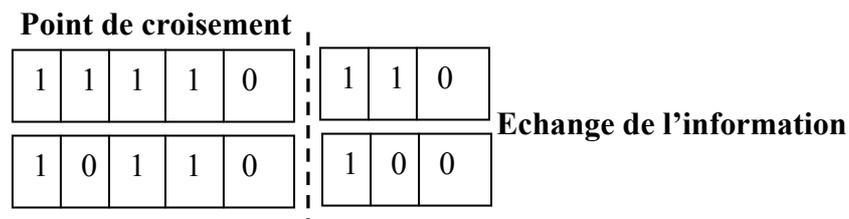


Figure 2.3 Opérateur de croisement mono point appliqué à deux chromosomes codés sur 8 bits.

c. L'opérateur de mutation agit sur un seul chromosome choisi avec une probabilité P_m . La mutation consiste à changer aléatoirement les valeurs de certains gènes du chromosome. Cet opérateur permet de créer une diversité dans la population afin d'éviter une convergence prématurée. La **Figure 2.14** montre un exemple de mutation sur un chromosome codé sur 8 bits.

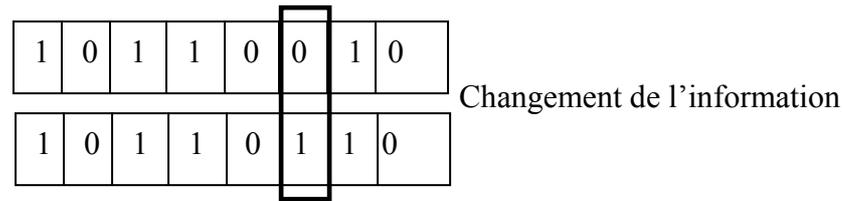


Figure 2.14 Opérateur de mutation appliqué à deux chromosomes codés sur 8 bits.

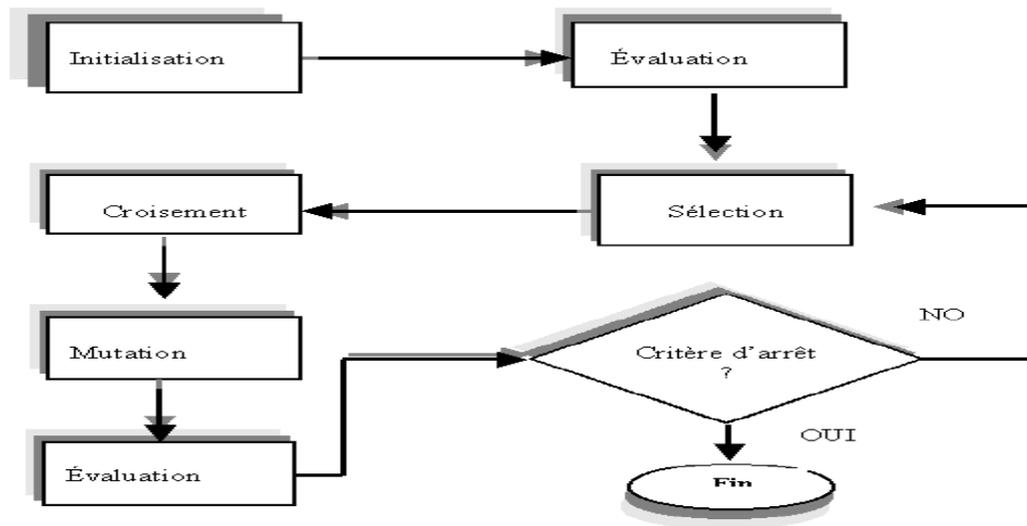


Figure 2.15 Structure générale d'un algorithme génétique.

Les algorithmes génétiques utilisent le même principe que l'évolution naturelle : la compétition entre individus, pour créer des systèmes complexes pouvant s'adapter à de nombreuses situations. Cela permet de résoudre efficacement des problèmes d'optimisation combinatoires en explorant complètement l'espace de recherche des solutions sans pour autant avoir un coût démesuré. Les algorithmes génétiques recherchent une « bonne » solution à un problème parmi un ensemble de solutions en tirant parti de l'historique de cette recherche afin de favoriser l'émergence des solutions les mieux adaptées. Les algorithmes génétiques ont fait leur preuve dans des domaines très variés. Afin de les appliquer à un problème particulier, il est nécessaire de définir convenablement la fonction d'évaluation et le codage d'une solution. Ces algorithmes nécessitent également la définition d'opérateurs, aussi bien de mutation que de croisement, en adéquation avec le problème.

Toutefois, la recherche d'une solution par les AG ne garantit pas l'obtention d'une solution optimale. Une population initiale mal choisie une convergence trop rapide vers un optimum local, etc., peuvent bloquer le processus de résolution. Il n'existe pas de méthodes qui permettent

de dire quelle est la taille optimale de la population, quel est le meilleur codage, quel taux de mutation et de croisement utiliser. Ces paramètres de fonctionnement dépendent de l'application et, bien souvent, ils sont fixés de façon empirique.

II.9.3 Les réseaux de neurones

De la même manière que le Calcul Evolutionnaire qui a été inspiré de l'évolution naturelle des espèces, le domaine des réseaux de neurones (RN) représente une tentative de reproduire artificiellement le fonctionnement du cerveau humain.

Les réseaux de neurones formels sont des modèles théoriques de traitement de l'information inspirés des observations relatives au fonctionnement des neurones biologiques et du cortex cérébral. Par analogie aux neurones biologiques, les neurones artificiels ont pour but de reproduire des raisonnements « intelligents » d'une manière artificielle. Ces neurones peuvent adopter de certaines qualités habituellement propres au biologique, c'est-à-dire, la généralisation, l'évolutivité, et une certaine forme de déduction.

Le modèle neurophysiologique

Les cellules nerveuses, appelées neurones sont les éléments de base du système nerveux central qui se compose d'environ 10^{12} neurones (mille milliards) [Mat-04] , Le neurone est une cellule composée de trois parties **Figure 2.16** :

- * Un corps cellulaire qui contient le noyau et se charge d'effectuer les transformations biochimiques nécessaires à la synthèse des éléments assurant la vie du neurone ;
- * Les dendrites qui sont des ramifications du corps cellulaire. Elles permettent au neurone de capter les signaux lui parvenant de l'extérieur ;
- * L'axone généralement plus long que les dendrites, il se ramifie à son extrémité où il communique avec les autres neurones. Il sert de moyen de transport pour les signaux émis par le neurone.
- * Les connexions entre neurones sont réalisées au niveau des synapses, lieu de proximité d'axone émetteur et de dendrites réceptrices.

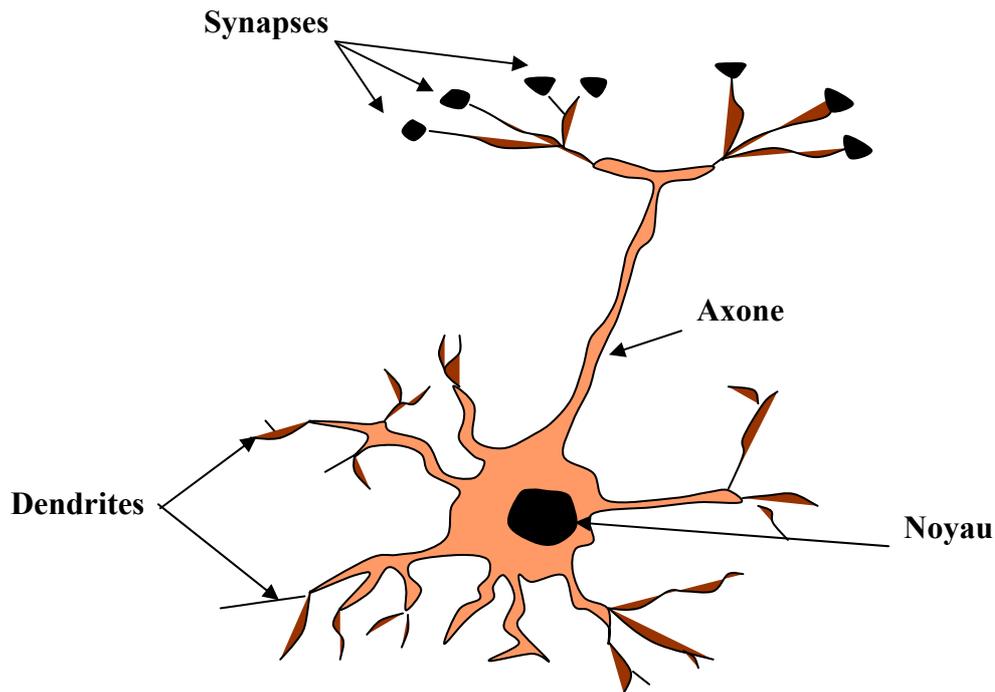


Figure 2.16 Structure d'un neurone biologique

Le neurone biologique reçoit des impulsions de ses neurones voisins avec lesquels il est connecté à travers les synapses. Les influx nerveux transmis par les dendrites sont sommés. Si la sommation dépasse un seuil, le neurone répond par un influx nerveux ou un potentiel d'action qui se propage le long de son axone. Si la sommation est inférieure au seuil, le neurone reste inactif. Les premières cellules qui alimentent le réseau peuvent être constituées par des capteurs (cellules sensorielles) comme les cellules de la rétine de l'oeil, par exemple.

Du neurone biologique au neurone formel

En 1943, Warren McCulloch et Walter Pitts en s'inspirant de leurs travaux sur le neurone biologique, ont proposé un des premiers modèles de neurone artificiel [McC-43], qui deviendra la base des réseaux de neurones artificiels. Un RN est un réseau composé généralement d'un grand nombre d'unités de calcul très simples appelés neurones formels. Ces neurones formels sont interconnectés entre eux, de sorte que les signaux sortant (outputs) des neurones deviennent des signaux entrant (inputs) dans d'autres neurones. Chaque neurone artificiel est un automate binaire à deux états « actif » et « inactif ». Il reçoit un nombre variable de signaux x en provenance des autres neurones. A chacun de ces signaux est associé un poids w représentatif de la force de la connexion. Le neurone change d'état suivant la somme des connexions pondérées dépasse ou non un seuil d'activation θ **figure 2.17**.

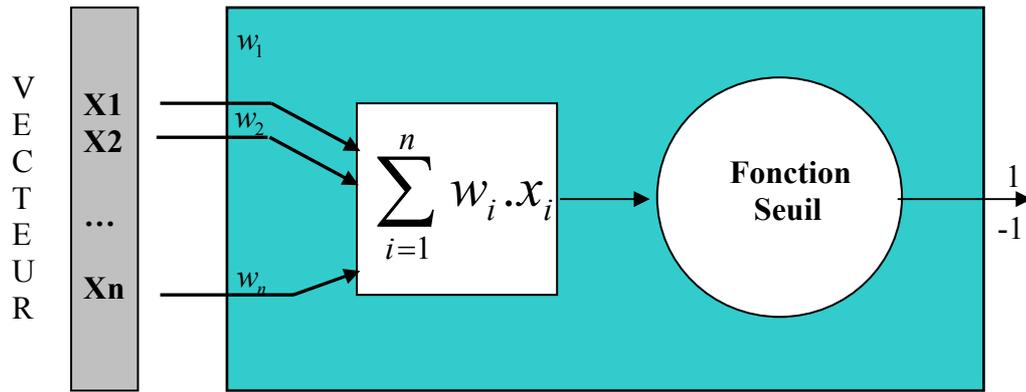
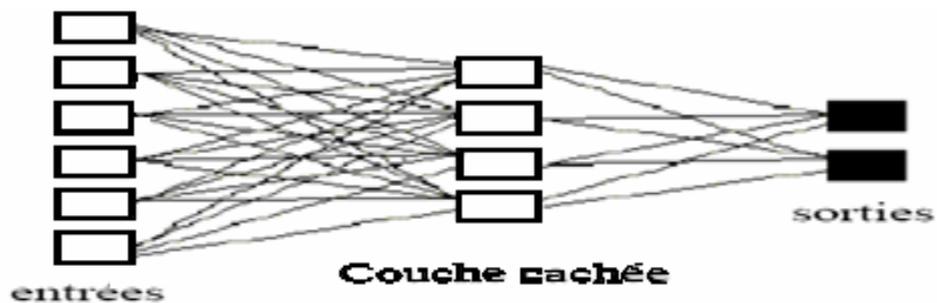


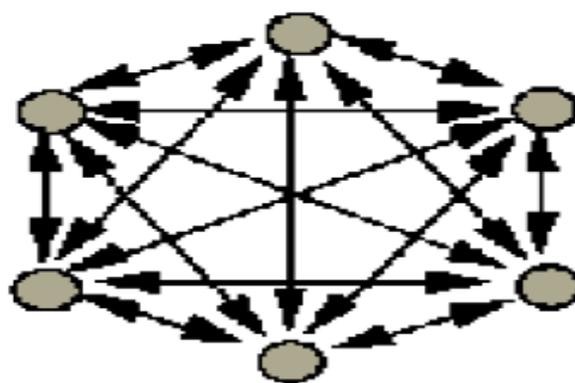
Figure 2.17 le neurone formel de Mac Culloch et Pitts

Un RN est caractérisé par sa topologie **figure 2.18.a** qui dépend de la façon dont les neurones sont reliés (réseaux en couche, complètement connecté, récurrent), par sa fonction d'activation **figure 2.18.b** et par le mode d'apprentissage utilisé (supervisé, non supervisé).



Réseau à couches

(a)



Réseau complètement connecté

(b)

Figure 2.18 Quelques formes de réseau de neurones, (a) Réseau à couches, (b) Réseau complètement connecté

D'une façon plus générale, on définit un neurone formel par les cinq paramètres suivants :

- 1-la nature des entrées (booléenne ou réelle)
- 2 -la fonction d'entrée totale, définissant le prétraitement effectué sur les entrées,
- 3-la fonction de seuillage (appelée aussi fonction d'activation) du neurone définissant son état interne en fonction de la somme pondérée de ses entrées. .
- 4-la fonction de sortie calculant la sortie du neurone en fonction de son état d'activation,
- 5-la nature des sorties du neurone

Dans leur principe de base, les RN reposent entièrement sur les concepts d'auto organisation et d'émergence : des règles locales de base d'interaction entre de nombreuses entités qui produisent au niveau global des comportements de plus haut niveau qui ne sont pas décrits au sein des règles. Si on leur rajoute, en respectant toujours ces principes, des capacités de création de neurones formels et d'apparition de nouveaux liens, ils présentent une technique intéressante pour la résolution de problèmes par émergence.

Grâce aux propriétés d'apprentissage et à la capacité puissante d'approximation des réseaux de neurones artificiels, ils ont trouvé des applications dans de nombreux domaines différents dont le contrôle de signaux, la robotique, le domaine médical (reconnaissance d'images de visages humains et d'objets) et en authentification (reconnaissance de signature, de caractères manuscrits, détection de fraudes). Ce paradigme s'avère souvent efficace pour résoudre des problèmes de classification et d'optimisation.

II.9.4 Algorithmes à essaims de particules

Les algorithmes « d'optimisation par essaim de particules » (Particle Swarm Optimization – PSO) introduits pour la première fois par Kennedy et Eberhart de 1995 à 2001 sont inspirés des déplacements collectifs observés chez certains animaux sociaux tels que les poissons et les oiseaux migrateurs. En effet, il est étonnant de voir comment ces animaux se déplacent en groupe dans une seule direction, se divisent parfois en plusieurs groupes afin d'éviter un obstacle ou un prédateur, puis reforment un groupe compact. Avec des règles locales très simples comme « rester proche des autres individus », « aller dans la même direction », « aller à la même vitesse », ces animaux sont capables d'éviter un prédateur par des mouvements d'explosion puis reforment le groupe originel, tout en maintenant la cohésion du banc. Dans l'algorithme à essaim

de particules, les individus de l'algorithme sont appelés particules et la population est appelée essaim.

L'approche par PSO trouve ses racines dans les travaux de l'infographiste Craig Reynolds qui fut le premier à modéliser ce genre de comportement collectif émergent par ses fameux boids [Ken-95] **Figure 2.19**. Les boids sont des individus autonomes. Chaque individu n'a qu'une vision locale de son environnement **Figure 2.20** et son comportement va être le résultat d'une somme d'attractions et de répulsions engendrées par son environnement direct, codées par des règles très simples **Figure 2.21**.

- 1 Chaque individu doit éviter de heurter ses voisins.
- 2 Chaque individu tend à s'approcher des vitesses et directions générales du groupe local, c'est-à-dire des voisins immédiats.
- 3 Chaque individu cherche à s'approcher du centre de gravité du groupe local.

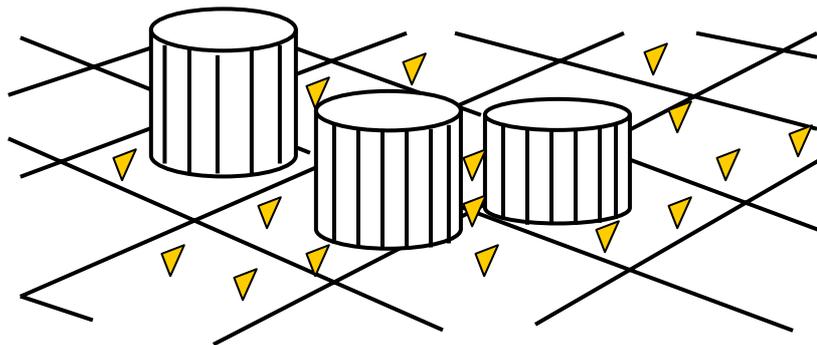


Figure 2.19 Les boids de Reynolds

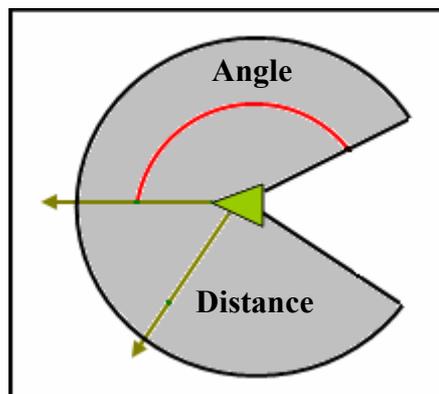


Figure 2.20 Le voisinage local d'un boid.

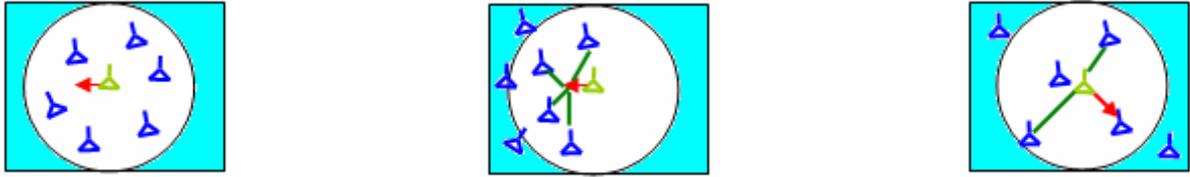


Figure 2.21 Les trois règles trouvées par Reynolds pour programmer des mouvements collectifs complexes.

La technique PSO utilise des règles de comportements locaux pour faire émerger à travers une dynamique collective auto-organisée la valeur optimale d'une fonction de fitness. Chaque particule est caractérisée par sa position (où la meilleure valeur pour sa fonction de fitness a été rencontrée) et un vecteur de changement de position (appelé vitesse). À chaque itération, une particule décide de son prochain mouvement en fonction de sa propre expérience, qui est dans ce cas la mémoire de la meilleure position qu'elle a rencontrée, et en fonction de son meilleur voisin. Ce voisinage peut être défini spatialement en prenant par exemple la distance euclidienne entre les positions de deux particules ou sociométriquement (position dans l'essaim de l'individu)

II.9.5 Les algorithmes de fourmis artificielles

En plus des sociétés animales, les sociétés d'insectes telles que les fourmis, les termites, les araignées et certaines espèces d'abeilles et de guêpes, constituent-elles aussi des sources d'inspiration pour le développement de systèmes artificiels « Les sociétés d'insectes nous proposent un modèle de fonctionnement bien différent du modèle humain: un modèle décentralisé, fondé sur la coopération d'unités autonomes au comportement relativement simple et probabiliste, qui sont distribuées dans l'environnement et ne disposent que d'informations locales. ». [Mon-00]

L'exemple des fourmis est le plus répandu dans la littérature à cause de leur capacité à réaliser des tâches hautement complexes à partir des interactions d'insectes simples à l'intelligence très rudimentaire. [Deneubourg, 1990; Colomi, 1992; Dorigo, 1996; Bonabeau, 1997; Van Dyke Parunak, 1997; Bonabeau, 1999; Topin, 1999]. En particulier deux comportements collectifs ont été principalement étudiés chez les fourmis : l'optimisation de chemin et le tri des éléments du couvain.

Le premier comportement met en évidence la capacité des fourmis à optimiser la procédure de fourragement. En effet, au départ les fourmis se déplacent d'une manière aléatoire de la fourmilière vers une source de nourriture en laissant des traces de phéromones sur le chemin qu'elles empruntent. Cette trace tend à attirer les congénères qui, en la suivant, vont parvenir à la nourriture. Il a été constaté alors que les fourmis qui choisissent l'itinéraire le plus court, le terminent le plus vite. Les fourmis vont alors préférer progressivement cet itinéraire car il est marqué avec une plus grande quantité de phéromones. [Oua-06] De plus l'évaporation naturelle de la phéromone renforce encore plus ce choix des fourmis par l'affaiblissement des chemins plus longs.

Le second comportement collectif des fourmis concerne leur aptitude à nettoyer leur nid en organisant collectivement des cimetières composés de cadavres empilés les uns sur les autres. Le principe est le suivant : plus un cadavre est isolé, plus la fourmi a de chances de ramasser ce cadavre. La probabilité pour une fourmi porteuse de déposer ce qu'elle transporte suit une règle inverse : plus le monticule observé est important, plus la probabilité de déposer le corps au sol sera grande

Les premières modélisations algorithmiques de ces comportements collectifs et auto-organisés sont dues à de Deneubourg et son équipe [Deneubourg, 1990] et furent après repris par d'autres chercheurs pour donner naissance à différents algorithmes pour la résolution de problèmes par stigmergie. Nous détaillerons plus amplement ces algorithmes de fourmis qui vont être la base de notre travail dans le chapitre suivant.

II.10 CONCLUSION

Nous avons exposés dans ce chapitre les deux types de méthodes utilisées dans la classification des données, qui sont les méthodes classiques et les méthodes intelligentes. Vu les inconvénients des méthodes classiques et les problèmes liés à leurs implantation, nous avons opté pour l'une des méthodes intelligentes qui est la méthode des fourmis artificielles non supervisé qui sera exposée dans le chapitre suivant.

CHAPITRE III : LES FOURMIS ARTIFICIELLES

III.1 INTRODUCTION

III.2 GENERALITES SUR LES FOURMIS

III.3 QUELQUES CONCEPTS S DE BASE

III.3.1 Problème d'optimisation

III.3.2 Méthodes de résolution

a) Les heuristiques

b) Les métaheuristiques

III.3.3 Les fourmis artificielles

III.4 LES ALGORITHMES DE FOURMIS ARTIFICIELLES POUR L'OPTIMISATION COMBINATOIRE

III.4.1 Inspiration biologique : le fourragement collectif par stigmergie

a) Le pont binaire

b) optimisation du chemin pour la recherche de nourriture

c) Le pont avec un obstacle

III.4.2 Modèles de fourragement pour la résolution des problèmes d'optimisation

III.4.2.1 optimisation par colonies de fourmis et problème du voyageur de Commerce

III.5 LES ALGORITHMES DE FOURMIS ARTIFICIELLES POUR LA CLASSIFICATION AUTOMATIQUE

III.5.1 Inspiration biologique

III.5.2 Modèles du tri du couvain et application à la classification de données

a) Les pionniers

b) Modèle de Lumer et Faieta

III.6 CONCLUSION

III.1 INTRODUCTION

Les études éthologistes ont montré que dans la nature, les petites créatures que sont les fourmis, arrivent à résoudre collectivement des problèmes quotidiens nombreux et trop complexes pour une seule fourmi tels que : recherche de nourriture, construction du nid, division du travail et allocation des tâches entre les individus, avec une organisation extrêmement structurée et sans aucune supervision. Par les comportements simples de chacune des fourmis, des interactions limitées à travers une coopération inconsciente, émergent des comportements collectifs intelligents et des modèles d'auto-organisation. Les fourmis sont devenues dès lors une nouvelle source d'inspiration pour la conception de méthodes de résolution de problèmes complexes. De plus cette source d'inspiration n'est pas unique étant donné que les fourmis sont dotées d'une grande diversité de caractéristiques disjointes et de comportements collectifs variés. Une nouvelle classe d'algorithmes est alors apparue sous le nom « algorithmes de fourmis artificielles ». Leur popularité est due d'une part à la facilité de mise en oeuvre et d'autre part à la complexité des fonctions réalisables. Deux comportements collectifs ont été principalement étudiés chez les fourmis : l'optimisation de chemin et le tri des cadavres. **[Mon-00]**

Le premier comportement appelé aussi fourrageage permet aux fourmis de retrouver le plus court chemin entre leur nid et une source de nourriture grâce à un système de marquage de phéromones. Ce comportement naturel a été modélisé et transposé à la résolution de nombreux problèmes d'optimisation combinatoires sous le nom d'un nouveau méta heuristique « optimisation par les colonies de fourmis ou OCF ». Le deuxième comportement collectif des fourmis concerne la capacité de certaines espèces de fourmis à organiser collectivement des cimetières composés de cadavres empilés les uns sur les autres. Là aussi, les chercheurs ont exploité ce comportement pour fournir des algorithmes de classification pour lequel l'informatique classique n'a pas donné de solution satisfaisant. **[Mon-00]**

Dans la suite nous présentons une brève introduction au monde des fourmis, ensuite nous décrirons en détail chacun des modèles de fourmis artificielles et nous exposons un algorithme de classification qui permet de nous donner un aperçu général démarche de processus de classification avec les fourmis.

III.2 GENERALITES SUR LES FOURMIS

Avec les abeilles et les guêpes, les fourmis font partie de la famille des Formicidae, incluse dans l'ordre des Hyménoptères, c'est à dire des insectes dont les deux paires d'ailes sont membraneuses et fines. Les fourmis sont réparties en 11 sous-familles, et approximativement 10 000 espèces. La fourmi peut mesurer de quelques millimètres à plusieurs centimètres de long. Son corps qui se compose de 3 parties principales : la tête, le thorax et l'abdomen. La tête est ornée d'une paire d'antennes qui abritent plusieurs organes des sens : le toucher, l'odorat et le goût. On trouve également à l'extrémité de la tête, la bouche qui sert à la fois à l'alimentation et très souvent d'outil de saisie. Les yeux, au nombre de 2, sont composés et formés d'une multitude de facettes. Quelques fourmis sont complètement aveugles, sans aucun organe visuel. Le thorax, ou corselet, porte 3 paires de pattes, très longues et terminées par 2 griffes. Les organes tactiles sont bien développés et sont constitués par des poils répartis sur tout le corps. Leur densité est plus forte sur les antennes et aux extrémités des pattes. L'abdomen est relié au thorax par un étroit pédoncule ou pétiole. L'abdomen renferme le jabot social, l'estomac, l'intestin et les conduits respiratoires, et chez les reines et les mâles, l'appareil reproducteur. La fourmi a un estomac un peu spécial, le "jabot social". Elle y garde en réserve de la nourriture, qui peut être régurgitée et transmise aux autres fourmis. La carapace (ou exosquelette) est composée d'une substance, résistante et imperméable que l'on nomme chitine. Elle renferme différents pigments, bruns, rouges, ou noirs qui sont responsables de la couleur de l'animal. [Oua-06]

Selon les espèces, les fourmis vivent dans le sol, sous un caillou, dans le bois mort ou dans les arbres. Certaines fourmis arboricoles construisent leurs nids dans les arbres, en cousant les feuilles entre elles ou en élaborant une architecture faite de grains de sable, de salive et de déjections. D'autres, comme les célèbres fourmis "légionnaires" d'Afrique, n'ont pas de nid et se déplacent par centaines de milliers d'individus sur de vastes territoires, à la manière des nomades.

La fourmi est un insecte exclusivement social vivant en société dans des habitations collectives : la fourmilière. La fourmilière classique est constituée par un ensemble de cellules réunies entre elles par un réseau complexe de galeries qui peut être très important. Dans ces cellules, les fourmis déposent les oeufs, les larves et les nymphes. La fourmilière et ses environs constituent le centre de la vie communautaire. A l'intérieur de la fourmilière les tâches sont divisées entre les fourmis selon la spécialité de chacune d'elles. Les activités des communautés

de fourmis sont caractérisées par un certain degré de division du travail souligné par une différenciation fonctionnelle et anatomique des individus. Une fourmilière peut abriter de 50.000 à plus de 1.000.000 individus bien différenciés tant au niveau physique qu'au niveau des comportements et des tâches à accomplir. On les sélectionne en castes :

Les reines : Dans une fourmilière on trouve une ou plusieurs reines. Les reines sont nettement plus grosses que les autres fourmis et peuvent vivre jusqu'à dix ou quinze ans. Leur rôle se résume essentiellement à pondre des oeufs et sont donc les fondatrices de nouvelles colonies.

Les ouvrières : Elles forment la majorité des habitants de la cité et se chargent de la défense et de l'entretien de la colonie, qui comprend la construction des galeries, les soins apportés aux jeunes, la quête de la nourriture, etc...

Les soldats : Ils sont plus massifs que les ouvrières, et possèdent souvent de grosses mandibules. Leur rôle est de défendre la fourmilière, et de transporter des charges lourdes. Mais certains, comme chez les fourmis "Grand Galop", il participent aux soins des larves, et donnent à manger aux fourmis qui le demandent. Chez cette espèce, qui est la plus grosse fourmi vivant à La Réunion, on peut distinguer des formes intermédiaires entre la petite ouvrière grêle et le puissant soldat.

Les jeunes sexués : ce sont des fourmis femelles et mâles. Ils sont facilement reconnaissables par leur plus grande taille, par la présence de deux paires d'ailes membraneuses sur le thorax et par 3 ocelles disposés en triangle sur le dessus de la tête. Les femelles sont plus grosses que les mâles. Ce sont les futurs rois et reines qui iront fonder de nouvelles colonies.

Le couvain : Il est constitué par les oeufs, les larves et les nymphes. Au bout de quelques jours les oeufs donnent naissance à des larves qui, bien nourries par les ouvrières pendant 15 jours à 3 semaines, se transforment en nymphes. Pendant la nymphose, la larve ne se nourrit plus.

Son corps tout entier subit de profondes mutations internes et externes, qui vont faire d'elle, petit à petit, une fourmi.

Les communications interindividuelles entre fourmis sont de plusieurs types et varient d'une espèce à l'autre. Les insectes sociaux en général, et les fourmis en particulier, ont développé des mécanismes de communication très élaborés pour les insectes sociaux et (pour les animaux en général). Il a été défini douze types de réponse mettant en oeuvre une forme de communication:

1. l'alarme.
2. l'attraction simple.
3. le recrutement (pour une source de nourriture ou un site de nidification).
4. l'entretien et la mue.
5. la trophallaxie (échange de liquides).
6. l'échange d'aliments solides.
7. les effets de groupe (augmentation ou inhibition d'une activité).
8. la reconnaissance des apparentés ou de caste.
9. la détermination de caste.
10. la compétition pour la reproduction.
11. le marquage du territoire et du nid.
12. la reproduction (différenciation du sexe, de l'espèce, de la colonie...).

Les principaux moyens de communication sont :

La communication tactile : lorsque deux fourmis se rencontrent, elles procèdent à quelques attouchements d'antennes ou de pattes, et se reconnaissent aussitôt comme membres de la même fourmilière ;

La communication sonore : les fourmis peuvent également utiliser des stimuli vibratoire comme moyen de communication. Elles frappent les parois de la fourmilière avec leur abdomen pour prévenir les autres. Elles tapotent aussi leurs antennes pour se parler. Ce signal est utilisé en fonction de l'espèce comme signal de détresse en cas de danger, comme signal de qualité de l'alimentation pour le recrutement pour une source de nourriture, comme signal de demande d'aide en cas où la nourriture trouvée est de grande taille.

La communication visuelle : développée chez certaines espèces, permet aux fourmis de retrouver leur chemin, guidées par des marqueurs visuels.

La communication chimique : les fourmis sont équipées de glandes produisant des phéromones, substances chimiques volatiles et odorantes qu'elles peuvent sentir par leurs antennes. Ce signal chimique porte l'information à la fois sur l'espèce, la société mais aussi la caste et le stade de développement auxquelles appartiennent les fourmis rencontrées. C'est en sécrétant cette substance qu'une fourmi éclaireuse marque le chemin qu'elle a utilisé et revient avertir ses pairs

de la présence de nourriture ou d'un danger, ainsi que de sa localisation. C'est le moyen de communication le plus efficace et plus utilisé par toutes les espèces de fourmis

La totalité des espèces de fourmis vit en effet en communautés plus ou moins importantes, qui sont qualifiées d'"eusociales" par [Lab-03], ce qui signifie qu'elles sont caractérisées par la présence simultanée et constante des trois critères suivants:

- * La coopération des membres de la société pour élever les jeunes et le couvain ;
- * L'existence d'une division des rôles, d'abord au niveau de la fonction reproductrice, ensuite au niveau des tâches à accomplir ;
- * L'existence d'une division des rôles, d'abord au niveau de la fonction reproductrice, ensuite au niveau des tâches à accomplir ;

III.3 QUELQUES CONCEPTS S DE BASE

III.3.1 Problème d'optimisation :

Un problème d'optimisation est tout problème défini par un espace de recherche des solutions, d'une fonction objectif qui associe un coût à chaque solution possible et un ensemble de contraintes. On cherche alors à trouver la solution optimale qui correspond à une solution de coût minimum ou maximum selon qu'il s'agit de minimiser ou de maximiser la fonction objectif.

Un problème d'optimisation combinatoire est tout problème d'optimisation pour lequel il faut trouver une solution optimale avec un espace de recherche de solutions fini mais extrêmement *grand*. Ce type de problème est dit « difficile ».

III.3.2 Méthodes de résolution :

Les méthodes de résolution des problèmes d'optimisation sont de deux types :

- 1- Les méthodes exactes (déterministes) : elles fournissent une solution optimale au prix d'un temps de résolution qui risque d'être exponentiel en fonction de la taille des données du problème.
- 2- Les méthodes approchées : pour un problème d'optimisation dit « difficile » aucune méthode exacte n'est capable de le résoudre exactement en un temps raisonnable. Dans ce cas on fait appel à ces méthodes permettant une optimisation approchée. Ce type de méthodes présente une solution contenue dans un certain intervalle autour de la solution

optimum avec un temps de calcul acceptable. Elles représentent un compromis entre la qualité de la solution trouvée et le temps de calcul nécessaire.

Parmi les méthodes de résolution approchées, on trouve :

a) Les heuristiques

Une heuristique est une méthode approchée simple, rapide et dédiée à un problème donné. Elle exploite les propriétés structurelles d'une solution et tente de la rendre rapidement une solution admissible par des critères de décision déduits de la connaissance du problème. Aucune garantie quant à l'optimalité de la solution trouvée ne peut être fournie.

b) Les métaheuristiques

Une métaheuristique est une méthode approchée générique dont le principe de fonctionnement repose sur des mécanismes généraux indépendants de tout problème. Les métaheuristiques sont stochastiques et donc peuvent éviter d'être piégés dans des minimums locaux. Elles sont principalement guidées par le hasard (exploration aléatoire de l'espace de recherche), cependant elles sont souvent alliées à d'autres algorithmes afin d'en accélérer la convergence.

III.3.3 Les fourmis artificielles

Une fourmi artificielle est une entité simple dotée d'un comportement rapproché ou étendu à celui de la fourmi réelle. Ce comportement doit être élémentaire, restreint et donc facile à programmer. A l'intérieur d'une colonie, les fourmis sont concurrentes et asynchrones, elles coopèrent inconsciemment ensemble pour la résolution du problème considéré. Les fourmis artificielles communiquent entre elles indirectement par stigmergie via des modifications de leur environnement (par exemple par dépôt de traces de phéromone artificielle) qui représente la mémoire collective de la colonie. Elles sont de plus enrichies par des contraintes et de comportements qu'on ne trouve pas dans leurs congénères réelles mais qui sont spécifiques au problèmes qu'elles résolvent.

III.4 LES ALGORITHMES DE FOURMIS ARTIFICIELLES POUR L'OPTIMISATION COMBINATOIRE

Dans la nature, les fourmis arrivent à résoudre différents problèmes d'optimisation liés à leur survie. Cette capacité naturelle a été simulée et transposée pour la résolution de divers

problèmes d'optimisation combinatoire. Les premiers travaux dans ce sens sont ceux de Dorigo et ses collègues qui ont mimé le comportement collectif de fourrageage observé chez les fourmis réelles et l'ont appliqué pour traiter le problème du voyageur du commerce **[Lab-03]**.

Les bons résultats obtenus ont incité de nouveaux chercheurs à l'utiliser en lui ajoutant certaines extensions que nous allons présenter dans la suite.

III.4.1 Inspiration biologique : le fourrageage collectif par stigmergie

Les études éthologistes ont montré que dans la nature, les fourmis utilisent la stigmergie pour mener à bien la tâche de recherche de nourriture appelée aussi fourrageage. En se déplaçant de leur nid à la recherche d'une source de nourriture, la fourmi laisse sur le chemin qu'elle emprunte une substance chimique de nature volatile appelée « phéromone ». Par ce marquage naturel, elle incite ses congénères à suivre le même trajet. Au début, les fourmis explorent différents chemins en effectuant des déplacements aléatoires. Une fois qu'un chemin intéressant (menant à une source de nourriture) est découvert, elles y déposent une quantité de phéromones renforçant ainsi son importance et la probabilité d'être choisi par d'autres fourmis de la colonie. D'un autre côté, les mauvais chemins auront tendance à être oubliés voir même disparaître avec l'évaporation de la phéromone. Ce procédé basé sur le mécanisme de rétroaction positive, assure que pendant le fourrageage pour la nourriture, les fourmis utilisent la voie d'accès la plus courte car elle sera la plus imprégnée par la phéromone.

a) Le pont binaire

L'expérience la plus connue est celle du pont binaire de Deneubourg en 1990, dans laquelle, le nid des fourmis et la source de nourriture sont séparés par un pont binaire constitué de deux branches identiques **Figure 3.1**. Comme au départ, il n'y a aucune trace de phéromone sur les deux branches, chacune peut être choisie avec la même probabilité **Figure 3.2**. Après une phase de fluctuations, la quantité de phéromone a tendance à s'accumuler sur un chemin et il a été observé alors que les fourmis prenaient collectivement le même itinéraire après quelques minutes **Figure 3.3**.

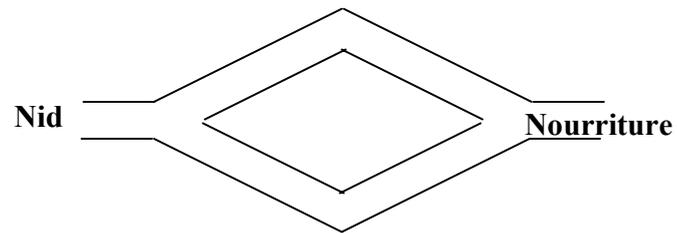


Figure 3.1 Pont Binaire de Deneubourg



Figure 3.2 Les fourmis suivent indifféremment les deux branches du pont.



Figure 3.3 Les fourmis ont tendance à emprunter le même chemin (celui d'en bas).

b) optimisation du chemin pour la recherche de nourriture

Les fourmis ont la particularité d'employer pour communiquer, des substances chimiques volatiles appelées les phéromones. Elles sont très sensibles à ces substances, qu'elles perçoivent grâce à des récepteurs situés dans leurs antennes. Ces substances sont nombreuses et varient selon les espèces. Les fourmis peuvent déposer des phéromones au sol, grâce à une glande située dans leur abdomen, et former ainsi des pistes odorantes, qui pourront être suivies par leurs congénères.

Les fourmis utilisent les pistes de phéromones pour marquer leur trajet, par exemple entre le nid et une source de nourriture. Une colonie est ainsi capable de choisir (sous certaines conditions) le plus court chemin vers une source à exploiter sans que les individus aient une vision globale du trajet [Mon-00].

En effet comme l'illustre la **Figure 3.4** les fourmis les plus rapidement arrivées au nid, après avoir visité la source de nourriture, sont celles qui empruntent les deux branches les plus courtes. Ainsi, la quantité de phéromone présente sur le plus court trajet est légèrement plus importante que celle présente sur le chemin le plus long. Or une piste présentant une plus grande concentration en phéromone est plus attirante pour les fourmis, elle a une probabilité plus grande d'être empruntée. La piste courte va alors être davantage renforcée que la longue, et à terme, sera choisie par la grande majorité des fourmis.



Figure 3.4 Expérience de sélection des branches les plus courtes par une colonie de fourmis : (a) au début de l'expérience, (b) à la fin de l'expérience.

c) pont avec un obstacle

Dans une troisième expérience, alors que les fourmis suivent leur chemin **Figure 3.5.a**, un obstacle est placé afin de leur barrer la route entre le nid et la source de nourriture **Figure 3.5b**. Les fourmis doivent alors contourner cet obstacle par la droite ou par la gauche afin de poursuivre leur chemin. Comme initialement, il n'y a aucune trace de phéromone le long de l'obstacle, la moitié des fourmis passent par le côté gauche et l'autre moitié par le côté droit. Il y'aura autant de fourmis **Figure 3.5b**. Après un certain temps, les chercheurs ont observé que les fourmis qui ont choisi le chemin le plus court, parviennent à reconstituer plus rapidement la concentration en phéromones sur ce nouveau chemin que celles qui ont choisi le chemin le plus long. Ainsi la concentration de phéromone sur le plus court chemin va augmenter, incitant ainsi d'autres fourmis à choisir le chemin riche en phéromones. Du à ce processus auto-catalytique, les fourmis vont finalement choisir le chemin le plus court **Figure 3.6c**.

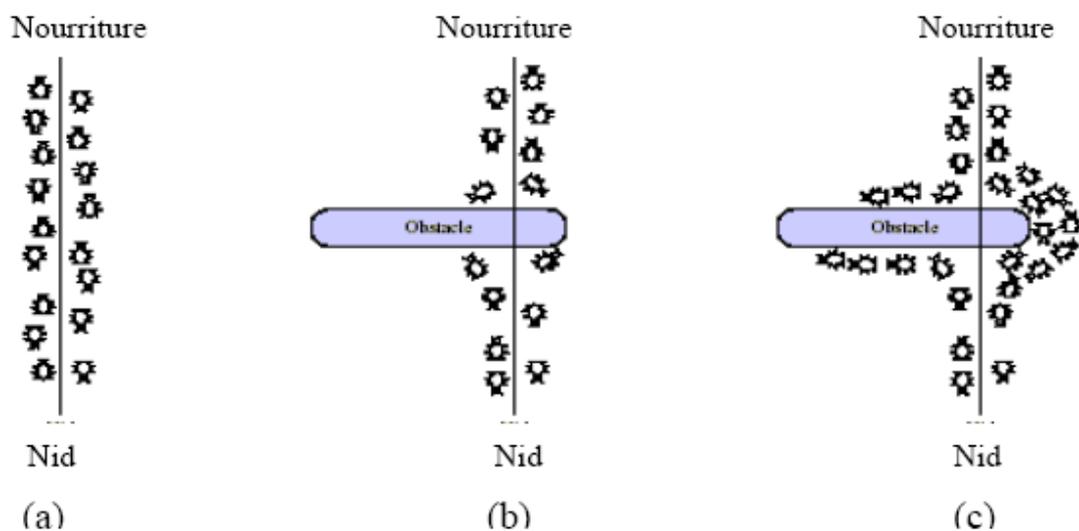


Figure 3.5 Effet de dépôt d'un obstacle entre la source de nourriture et le nid

III.4.2 Modèles de fourragement pour la résolution des problèmes d'optimisation

III.4.2.1 optimisation par colonies de fourmis et problème du voyageur de commerce

Le problème du voyageur de commerce consiste à chercher le trajet le plus court reliant n villes données, chaque ville ne devant être visitée qu'une seule fois. Le problème est plus généralement défini dans un graphe complètement connecté (A, N) , où les villes sont les noeuds N et les trajets entre ces villes, les arêtes A .

Dans l'algorithme du Ant-System (AS) [S7], à chaque itération, $t(1 \leq t \leq t_{\max})$ chaque fourmi a ($a = 1, \dots, j$) parcourt le graphe et construit un trajet complet de $n = |N|$ étapes (on note $|N|$ le cardinal de l'ensemble N). Pour chaque fourmi, le trajet entre une ville g et une ville k dépend de :

- la liste des villes déjà visitées, qui définit les mouvements possibles à chaque pas, quand la fourmi a est sur la ville g .
- l'inverse de la distance entre les villes : $\eta_{gk} = 1/d_{gk}$, appelée visibilité. Cette information statique est utilisée pour diriger le choix des fourmis vers des villes proches,
- la quantité de phéromone déposée sur l'arête reliant les deux villes, appelée l'intensité de la piste. Ce paramètre définit l'attractivité d'une partie du trajet global et change à chaque passage d'une fourmi. C'est, en quelque sorte, une mémoire globale du système, qui évolue par apprentissage.

La règle de déplacement appelée " règle aléatoire de transition proportionnelle" est la suivante :

$$P_{gk}^a(t) = \begin{cases} \frac{(\tau_{gk}(t))^\alpha \cdot (\eta_{gk})^\beta}{\sum_{l \in J_g^a} (\tau_{gl}(t))^\alpha \cdot (\eta_{gl})^\beta} & \text{si } k \in J_g^a \\ 0 & \text{si } k \notin J_g^a \end{cases} \quad (3.1)$$

Où J_g^a est la liste des déplacements possibles pour une fourmi a lorsqu'elle se trouve sur une ville g , et α β sont deux paramètres contrôlant l'importance relative de l'intensité de la piste, $\tau_{gk}(t)$, et de la visibilité, $\eta_{gk}(t)$. Avec $\alpha = 0$, seule la visibilité de la ville est prise en compte ; la ville la plus proche est donc choisie à chaque pas. Au contraire, avec $\beta = 0$, seules les pistes de phéromone jouent. Pour éviter une sélection trop rapide d'un trajet compromis entre ces deux paramètres, jouant sur l'importance relative de la diversification et de l'intensification, est nécessaire.

Après un tour complet, chaque fourmi laisse une certaine quantité de phéromones sur l'ensemble de son parcours, quantité qui dépend de la qualité de la solution trouvée :

$$\Delta \tau_{gk}^a(t) = \begin{cases} \frac{Q}{L^k(t)} & \text{si } (g, k) \in T^a(t) \\ 0 & \text{si } (g, k) \notin T^a(t) \end{cases} \quad (3.2)$$

Où $T^a(t)$ est la tournée faite par la fourmi (a) à l'itération t , $L^k(t)$ et la longueur du trajet et Q un paramètre fixé.

L'algorithme ne serait pas complet sans le processus d'évaporation des pistes de phéromone. En effet, pour éviter d'être piégé dans des solutions sous optimales, il est nécessaire de permettre au système " d'oublier " les mauvaises solutions. La règle de mise à jour des pistes est donc :

$$\tau_{gk}(t+1) = (1-\rho).\tau_{gk}(t) + \Delta\tau_{gk}(t) \quad (3.3)$$

Où $\Delta\tau_{gk}(t) = \sum_{a=1}^m \Delta\tau_{gk}^a(t)$ et m est le nombre de fourmis. La quantité initiale de phéromone sur chaque arête suit une distribution uniforme d'une petite quantité $\tau_0 \geq 0$.

La **Figure 3.6** présente un exemple simplifié du problème du voyageur de commerce où l'on optimise le chemin le plus court avec les fourmis.

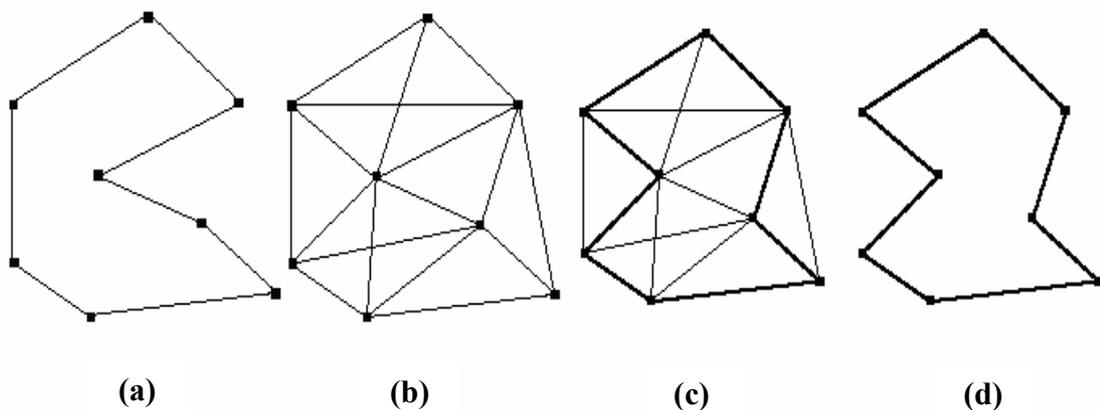


Figure 3.6 Le problème du voyageur de commerce optimisé par l'algorithme AS, les points représentent les villes et l'épaisseur des arêtes la quantité de phéromone déposée. (a) exemple de trajet construit par une fourmi, (b) au début du calcul, tous les chemins sont explorés, (c) le chemin le plus court est davantage renforcé que les autres, (d) l'évaporation permet d'éliminer les solutions les moins bonnes

Fourmis sont capables d'apprendre visuellement des routes familières pour se déplacer entre un site alimentaire et leur nid [S7] ; La plupart des caractéristiques qui intéressent les chercheurs

sont cependant collectives. Les caractéristiques individuelles ne sont évidemment pas une particularité des fourmis mais de tous les organismes vivants ayant un souci de survie.

III.5 LES ALGORITHMES DE FOURMIS ARTIFICIELLES POUR LA CLASSIFICATION AUTOMATIQUE

Les algorithmes de classification automatique sont un autre type d'algorithmes inspirés de comportements collectifs observés chez les fourmis : le tri collectif de couvains ou la constitution de cimetières. Les premiers travaux dans ce domaine ont été ceux de Deneubourg et son équipe en 1990, se basant sur une colonie de fourmis artificielles qui se déplacent aléatoirement sur une grille rectangulaire et sont capables de ramasser et de déposer des objets présents sur une grille dans le but de les regrouper selon un critère de similarité. Ces travaux ont été par la suite améliorés et étendus à différents domaines d'application.

III.5.1 Inspiration biologique

Certains travaux ont montré que certaines espèces de fourmis parviennent à organiser divers éléments du couvain tels que les oeufs, les larves [Mon-00]. Ce comportement est basé sur deux règles simples mais suffisantes pour former des tas d'éléments de même type et qui sont les suivantes :

- *- Lorsqu'une fourmi rencontre un élément du couvain, plus cet élément est isolé, plus elle a de chance de le ramasser ;
- *- Lorsqu'une fourmi transporte un élément du couvain, la probabilité qu'elle le dépose est d'autant plus grande que la densité d'éléments de même type dans le voisinage est grande.

Là aussi le tri du couvain est le résultat du processus de rétroaction positive et l'utilisation d'informations locales à chaque fourmi comme on le verra dans les prochaines sections.

III.5.2 Modèles du tri du couvain et application à la classification de données

Ces observations ont conduit à différentes modélisations qui avaient pour objectif de simuler et de reproduire ce comportement collectif et auto-organisé des fourmis afin de les appliquer par la suite à d'autres types d'applications..

a) Les pionniers

L'origine des algorithmes de classification par les fourmis est due aux recherches entreprises par Deneubourg et son équipe en 1990. Lors des expériences de simulation, les objets à classer à rassembler sont placés aléatoirement sur une grille à deux dimensions. Les

fourmis sont placées aussi aléatoirement sur la grille représentant l'environnement dans lequel elles évoluent. Chaque fourmi n'a qu'une perception locale de son environnement et a pour tâche de déplacer les objets en fonction de la concentration des objets de même type dans leur environnement proche appelé « voisinage »

Pour rassembler en tas un ensemble d'éléments (d'objets) de même type, les probabilités de ramasser un objet et de le déposer ont été explicitées : quand une fourmi ne transporte aucun élément, sa probabilité d'en ramasser un, rencontré sur son chemin, est donnée par

$$p_p = \left(\frac{k_1}{k_1 + f} \right)^2 \quad (3.4)$$

Où k_1 est une constante positive et f correspond à la proportion d'éléments perçus dans le voisinage de la fourmi. Quand il y a peu d'objets dans le voisinage de l'objet convoité par la fourmi, $f \ll k_1$ ce qui signifie que p_p est proche de 1 et l'objet a beaucoup de chance d'être ramassé. Inversement, quand le voisinage est dense en éléments, $f \gg k_1$ et alors p_p est proche de 0. Quand une fourmi chargée d'un objet se déplace, sa probabilité de déposer l'objet est donnée par :

$$p_d = \left(\frac{f}{k_2 + f} \right)^2 \quad (3.5)$$

Où k_2 est une constante positive. L'évaluation de f est proposée pour une implantation en robotique : f correspond au nombre d'objets rencontrés durant les T derniers déplacements divisé par le nombre maximum d'objets qui auraient pu être rencontrés. Comme les résultats dans plusieurs articles qui utilisent la méthode des fourmis pour la classification supervisée montrent l'apparition de groupes d'objets, les agents ainsi définis permettent donc de ranger une surface sur laquelle des objets ont été éparpillés. L'adaptation de ce principe à des objets de plusieurs types, par exemple deux : A et B , peut alors se faire en particulierisant f : f_A et f_B correspondent à la proportion d'objets de type A et B . Le comportement de rassemblement se transforme alors en tri, ce qui se rapproche de notre problème de partitionnement.

b) Modèle de Lumer et Faieta :

L'algorithme proposé par Deneubourg a été repris et étendu par Lumer et Faieta [Mon-00] pour la classification des données numériques. Les extensions introduites concernent en particulier les points suivants :

- *- Les données sont représentées par des vecteurs de caractéristiques (numériques).
- *- La similarité entre deux données est mesurée comme une distance euclidienne entre leur vecteur de caractéristiques.
- *- La fourmi est capable de percevoir une région R_s de $s.s$ cases autour de sa position courante sur la grille.

La **Figure 3.7**. Représente un exemple de grille utilisé dans l'algorithme de Faieta et Lumer. Les objets sont représentés par des cases de deux motifs décrivant leurs types et le rectangle en trait épais est la région perçue par la fourmi.

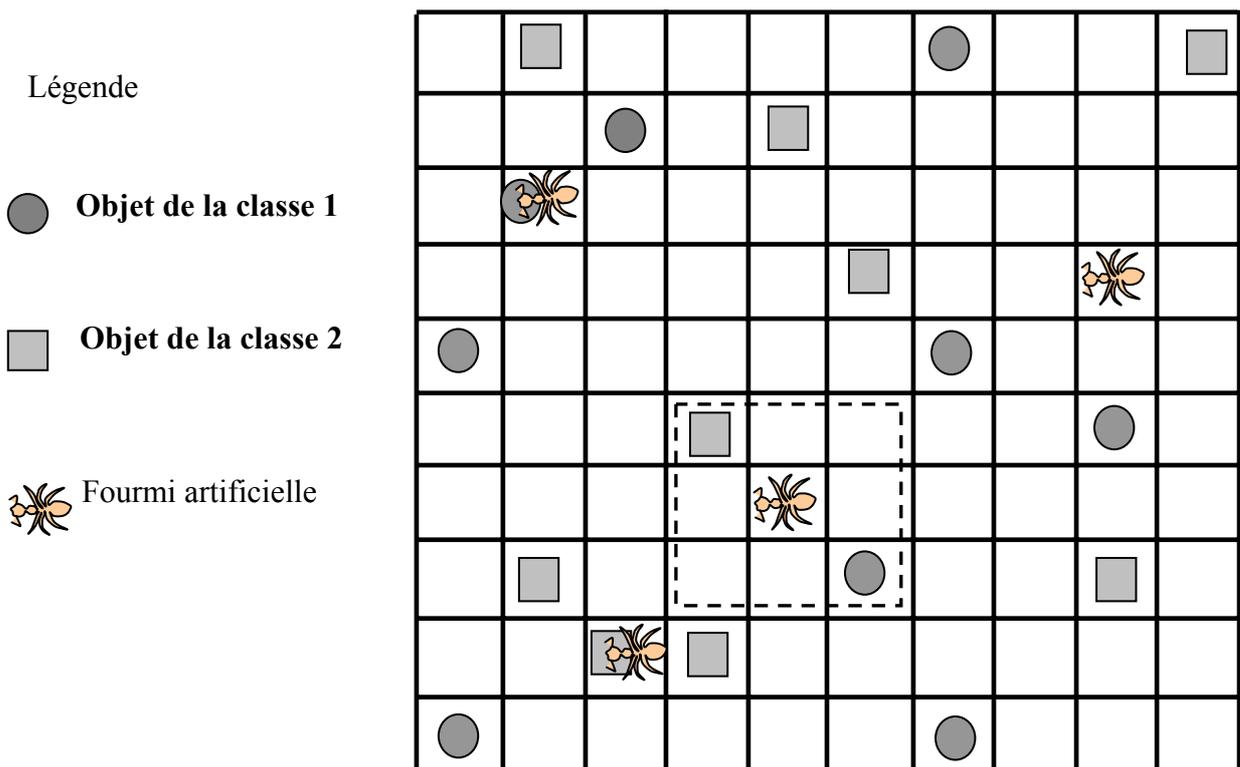


Figure 3.7. Grille de classification de Lumer et Faieta (d'après Labroche, 2003]

Les probabilités de déplacement et de dépôts des objets deviennent alors

$$p_p(i) = \left(\frac{k_1}{k_1 + f(o_i)} \right)^2 \quad (3.7)$$

$$p_d(i) = \begin{cases} 2.f(o_i) & \text{si } f(o_i) < k_2 \\ 1 & \text{si } f(o_i) \geq k_2 s \end{cases} \quad (3.8)$$

$$f(o_i) = \begin{cases} \frac{1}{s^2} \sum_{o_j \in R_s(r(o_i))} 1 - \frac{d(o_i, o_j)}{\alpha} & \text{si } f > 0 \\ 0 & \text{sinon} \end{cases} \quad (3.9)$$

$r(o_i)$ est la position de l'objet i sur la grille. $f(o_i)$ est une mesure de similarité moyenne de l'objet i avec les objets j de son entourage. Le facteur α contrôle la consistance de la fonction de dissimilarité entre les objets. Si α est trop élevé les objets différents seront mis dans la même classe. Dans le cas contraire les objets similaires ne seront pas regroupés ensemble. Pour Les tests de l'algorithme Lumer and Faieta les valeurs sont : $k_1 = 0.1, k_2 = 0.15, s = 3, \alpha = 0.5$

Algorithme 3.2: Algorithme LF (Lumer and Faieta, 1994)

-
- (1) Placer aléatoirement les N objets o_1, \dots, o_N sur la grille G
 - (2) **pour** $T = 1$ à T_{\max} **faire**
 - (3) **pour tout** $a_j \in \{a_1, \dots, a_A\}$ **faire**
 - (4) **si** la fourmi a_j ne transporte pas d'objet et $r(o_i) = r(a_j)$ **alors**
 - (5) Calculer $f(o_i)$ et $p_p(o_i)$
 - (6) La fourmi a_j ramasse l'objet o_i suivant la probabilité $p_p(o_i)$
 - (7) **sinon**
 - (8) **si** la fourmi a_j transporte l'objet o_i et la case $r(a_j)$ est vide **alors**
 - (9) Calculer $f(o_i)$ et $p_d(o_i)$
 - (10) La fourmi a_j dépose l'objet o_i sur la case $r(a_j)$ avec une probabilité $p_d(o_i)$
 - (11) **fin**
 - (12) **fin**
 - (13) Déplacer la fourmi a_j sur une case voisine non occupée par une autre fourmi
 - (14) **fin**
 - (15) **fin**
 - (16) **retourner** l'emplacement des objets sur la grille
-

Les résultats obtenus ont montré que l'algorithme génère un nombre de classes qui est très souvent très éloigné du nombre réel de classes. Afin de remédier à cela, Lumer et Faieta ont introduit trois extensions au comportement de base des fourmis artificielles :

- * Les fourmis se déplacent sur la grille avec une vitesse propre à chacune d'elles comprise entre 1 et v . Les fourmis les plus rapides sont moins sensibles aux dissimilarités entre deux objets, ce qui permet de diminuer le nombre de classes générées.
- * Chaque fourmi possède une mémoire à court terme lui permettant de se souvenir des positions des m derniers objets classés. Si une fourmi transporte un objet, elle cherche dans sa mémoire l'objet déjà classé qui est le plus proche de celui qu'elle transporte. Si elle le trouve, elle se déplacera (avec une certaine probabilité) vers ce nouvel emplacement pour y déposer son objet.
- * Si au bout d'un certain nombre d'itérations la fourmi stagne (ne fait plus aucun déplacement) elle peut détruire un groupe en ramassant l'objet le plus éloigné du groupe.

Pour notre algorithme la classification n'est pas faite d'une façon numérique mais d'une façon symbolique. Dans la modélisation nous avons trois types isolants, pour chaque type nous lui associe des propriétés par exemple (formes, couleurs,.....etc.). La **figure 3.8** montre les types des isolants, et quels sont les base de classification.

□ Isolant 1 type carré

△ Isolant 2 type triangle

○ Isolant 3 type cercle

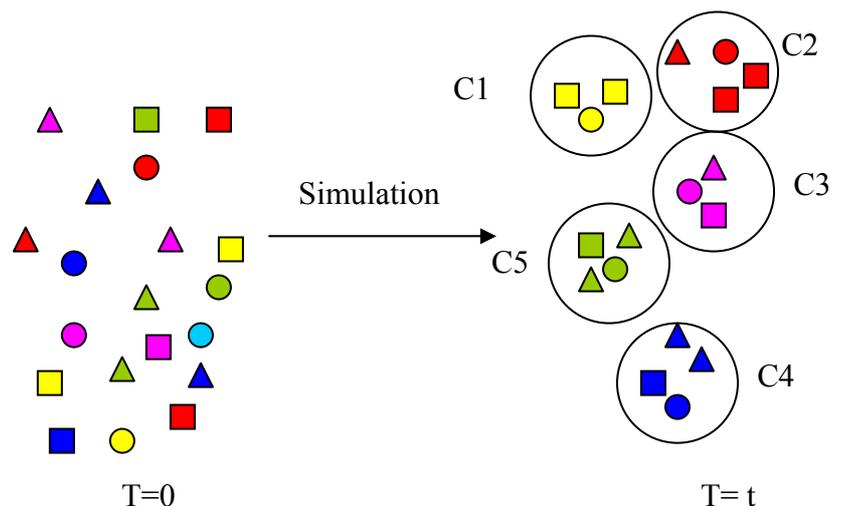


Figure 3.8 : une classification symbolique (prendre la couleur comme une base la (Classification) des isolants

Enfin, la méthode de classification peut prendre une seule propriété ou des multi propriétés.

III.6 CONCLUSION

Dans ce chapitre il a été question de donner un aperçu sur les fourmis réelles et sur le comportement qu'elles prennent pour résoudre les problèmes aux quelles elles sont confrontées dans leurs environnement. A partir de cela, les informaticiens s'inspirent pour mettre au point des modèles mathématiques pour faire face à des problèmes réels analogues à ceux rencontrés par les fourmis. Ce ci a permis de mettre au point des algorithmes permettant la résolution de ces problèmes notamment les problèmes liés à la classification que nous allons étudier au chapitre suivant.

CHAPITRE IV : SIMULATION ET RESULTATS

IV.1 INTRODUCTION

IV.2 LE LANGAGE DE PROGRAMMATION

IV.2.1 Les tortues

IV.2.2 Les patchs

IV.2.3 L'observateur

IV.2.4 Les commandes

V.2.4.1 Commandes primitives

IV.2.5 Reporters

IV.2.6 Boutons

IV.2.7 Les variables

IV.2.7.1 Les variables locales

IV.2.7.2 Les variables prédéfinies

IV.2.8 Les fonctions

IV.2.9 Affichage de résultats

IV.2.10 Exemple d'une interface du NetLogo

IV.3 LES RESULTATS DE CLASSIFICATION

IV.3 .1 La variation de rayon de détection

IV.3 .2 la variation des nombres des fourmis

IV.3 .3 la variation des nombres des d'objets

IV.3 .4 augmentation des types isolants

IV.3 .5 Changement de la taille de l'environnement

IV.4 CONCLUSION

IV.1 INTRODUCTION

Net Logo est un environnement de programmation pour la modélisation et la simulation des systèmes naturels complexes et aussi les phénomènes collectifs. Il se base sur des agents mobiles capables d'effectuer de nombreuses opérations simultanément avec une grande précision. Il est aussi bien adapté à la modélisation des systèmes multi agents « par exemples : les fourmis, les termites, les abeilles... ».il nous donne une gamme de liberté pour créer des modèles propres à la classifications, l'optimisation,... dans les divers domaines : en sociologie, biologie, médecine, physique, chimie, mathématique, informatique, économie et psychologie sociale.

IV.2 LE LANGAGE DE PROGRAMMATION

Un agent est une entité ayant la possibilité d'interagir avec l'environnement et avec d'autres agents. Cette entité peut être une personne, un animal, un insecte, une case, un pays, etc....Les agents ont la particularité de pouvoir suivre des instructions ou des lois que nous donnons. Le comportement de chaque agent est déterminé avec un ensemble de lois, en général stimulus-réponse (spécial et simple) [Abd-06].

Net Logo est un logiciel permettant de simuler les interactions entre un grand nombre d'agents.Net Logo permet de montrer que lorsque des agents suivent des lois simples, le résultat est parfois complexe et inattendu. Les agents vivent sur un support 2D constitué de cellules appelées « patches ». Ces cellules ont des coordonnées. Il y a trois types d'agents : les tortues (turtles), les patches, et les observateurs.

IV.2.1 Les tortues : dans Net Logo on peut manipuler des entités qu'on peut appeler fourmis,termites,loups, etc. ce sont des types de races. On remarque que par défaut, ces entités sont dénommées 'tortue'. Elles sont appelées « tortues » en hommage au langage de programmation dont dérive net logo dans lequel le programmeur contrôle des tortues sur un écran. Les tortues repentent à des commandes ou des fonctions. Elles peuvent des centaines, voire des milliers, sur l'écran au même moment. Les tortues ont des coordonnées (**xcor** et **ycor**). Ces coordonnées peuvent être flottants.

IV.2.2 Les patches: les patchs forment l'environnement des tortues, ce sont les cases de l'écran principal de net logo. Grâce à la programmation des patchs, on peut donc faire intervenir l'environnement des tortues ; ceci permet d'établir des modèles beaucoup plus proches de la réalité, quand on sait l'importance des interactions entre les individus et leurs environnement

dans l'établissement des comportements. Ils permettent par exemple, de mettre de la nourriture dans l'environnement des tortues, ou de laisser des phéromones.

Chaque patch est un carré (une case) sur lequel les tortues peuvent se déplacer. Le patch central a pour coordonnées (0,0). Les coordonnées d'un patchs sont appelées `pxcor` et `pycor`. Le nombre total de patches est déterminé par les variables (**screen-edge-x** et **screen-edge-y**)

Screen-edge-x: valeur maximale en abscisses.

Screen-edge-y : valeur maximale en ordonnées.

Par défaut, ces deux variables sont à 17. Il y a donc 1225 patches au total sur l'écran. On peut changer ce nombre dans la fenêtre net logo graphique. Les coordonnées des patches sont des valeurs entières. Notons que le support des patches n'est pas borné. Chaque patch a le même nombre de voisins. Ainsi si une tortue disparaît d'un coté de l'écran, elle réapparaît du coté opposé.

IV.2.3 L'observateur : C'est une sorte de superviseur dans le monde des tortues. L'observateur est extérieur au monde dans lequel vivent les tortues mais il peut superviser ce qu'il s'y passe et ordonner aux tortues d'exécuter des instructions. C'est lui que l'on utilise lorsque l'ont veut gérer un programme. L'observateur peut être utilisé pour attribuer des ordres spécifiques à des patches ou à des tortue. Il collecte également des données pour créer des graphiques.

IV.2.4 Les commandes : commandes qui définissent les actions à exécuter par les agents.

Elles sont de 2 types:

- ✓ **Primitives**: commandes prédéfinies dans NetLogo.
- ✓ **Procédures**: commandes définies par le programmeur.

Une commande procédure est définie par un *nom*, précède par le mot clé **to**, et se terminant par le mot clé **end**.

Exemple: commande procédure **setup**

```
to setup
  ca      ;; nettoyer l'écran
```

```

crt 100    ;; créer 100 tortues
end

```

IV.2.4.1 Commandes primitives : Ce sont des commandes prédéfinies dans NetLogo

- ca (clear all),
- crt (create turtles),
- lt (left turn),
- rt (right turn),
- fd (forward),
- set (affectation de variable),
- ask,...

Par exemple :

```

to setup

ca      ;;nettoyer l'écran
crt 100 ;;créer 100 tortues
ask turtles
    [set color red      ;;couleur des tortues en rouge
    rt random 3600    ;;orienter chacune au hasard
    fd 50]              ;;les faire avancer de 50 pas
ask patches
    [if (pxcor > 0)    ;;les patches du coté droit de la grille
    [set pcolor green]] ;;de l'écran en vert

```

End

Structure ask:

Cette structure permet d'appliquer une ou plusieurs instructions à toutes les variables d'un même type. En effet, la structure suivante :

```

ask turtles
[
.....]

```

Permet d'appliquer à toutes les variables de type 'tortue' les instructions qui se situeront dans le corps de la structure. On peut également affiner la sélection en précisant les spécificités de ces variables. En effet, la structure suivante :

```
ask turtles
  [ fd 10
    ... ]
```

Appliquer à toutes les variables de type tortue d'avancer 10 pas.

Exemple : procédure setup

```
to setup
  ca      ;;nettoyer l'écran
  crt 100 ;;créer 100 tortues
  ask turtles      ;;demander à toutes les tortues
    [set color red      ;;couleur des tortues en rouge
     rt random 360      ;;orienter chacune au hasard
     fd 50]             ;;les faire avancer de 50 pas
  ask patches      ;;demander à tous les patches
    [if (pxcor > 0)     ;;les patches du coté droit
     [set pcolor green]] ;;de l'écran en vert
```

End

Structure if:

Syntaxe: if condition [traitement]

Exemple: ask patches [if pxcor > 0 [set pcolor blue]]

Structure ifelse:

Syntaxe: ifelse condition [traitement_si_la_condition_est_vérifiée] [traitement_sinon]

Exemple: ask patches [ifelse pxcor > 0 [set pcolor blue] [set pcolor red]]

IV.2.5 Reporters:

Exécutent une opération et retournent un résultats soit à une commande soit à un autre reporter (un peu comme une fonction).

Reporters primitifs: prédéfinis dans NetLogo

Exemple : la fonction `Random`,...

Reporters procédures: définis par le programmeur.

IV.2.6 Boutons:

Les boutons de l'interface fournissent un moyen simple de contrôler le modèle.

En général un modèle ou:

- Un bouton **setup** pour initialiser l'état du monde.
- Un bouton **go** pour exécuter le modèle en continue.
- Eventuellement d'autres boutons pour exécuter d'autres actions.

Un bouton contient un ' code NetLogo '.

- Once-button:** le code est exécuté une fois.
- Forever-button:** le code s'exécute indéfiniment jusqu'à ce qu'on clique dessus à nouveau, ou si le code exécute la commande **stop**.

- Il faut spécifier qui exécutera le code associé au bouton:
 - L'observateur.
 - Toutes les tortues.
 - Tous les patches
 - (autre cas particuliers).

IV.2.7 Les variables:

Les différentes variables se définissent de la manière suivante:

- Variable globale : `globals [var1 var 2...]`, déclaration des variables.
- Variable tortue : `turtles-own [var1 var2 ...]`, Chaque tortue a une valeur pour chacune des variables de tortue: ***xcor***, ***ycor***, ***color***, ***heading***
- Variable patch: `patches-own [var1 var2]`, Pareil pour les variables de patch: ***pxcor***, ***pycor***, ***pcolor***

On utilise la commande "set" pour les initialiser. Par défaut, elles sont initialisées à 0.

Les variables globales peuvent être lues et modifiées par n'importe quel agent à tout moment. Une tortue peut lire et modifier les variables patches du patch sur lequel elle agit. Lorsque l'on veut qu'un agent lise ou modifie la variable d'un autre agent, il faut mettre `-of` après le nom de la variable et spécifier, après, de quel agent il s'agit.

Example:

```
set color-of turtle 5 red ;;la tortue n° 5 devient rouge.
```

set pcolor-of patch 2 3 green ;;le patch de coordonnées (2,3) devient vert

IV.2.7.1 Les variables locales :

Ce sont des variables exclusives à une fonction. On les définit de la manière suivante : **locals** [var1 var2 ...] Ces variables ne sont utilisables qu'à l'intérieur de la fonction. Elles doivent être définies au début de la fonction, avant toute commande.

IV.2.7.2 Les variables prédéfinies :

Il existe plusieurs variables prédéfinies relatives aux variables « tortues » :

breed : (= race)

Elle permet de sélectionner l'ensemble des tortues de la même race.

Example: if breed = chat [set color blue] , la race chat a un couleur blue.

breeds:

Ce mot clé permet de définir l'ensemble des races. Il ne peut être utilisé qu'au début du programme, avant toute définition de fonction.

Exemple : breeds [souris grenouille]....

color :

Elle définit la couleur d'une tortue.

heading :

Elle indique vers quelle direction la tortue est dirigée. heading [0 ; 360].

0 = Nord, 90 = Est, 180 = Sud, 270 = Ouest

hidden ?

Elle contient un booléen indiquant si la tortue est visible ou non. On peut modifier cette variable pour faire apparaître ou disparaître une tortue.

label :

Elle permet d'associer une valeur à une tortue. Cette valeur peut être de n'importe quel type.

label-color :

Elle détermine la couleur du label d'une tortue (si elle en a un). label-color [0 ; 140].

shape :

Elle contient une chaîne de caractères représentant le nom de la configuration courante de la tortue.

size :

La taille par défaut est 1.0. La tortue a la même taille qu'un patch. Toutes les tortues ont la même taille sauf si on coche la case « turtle-sizes ».

who :

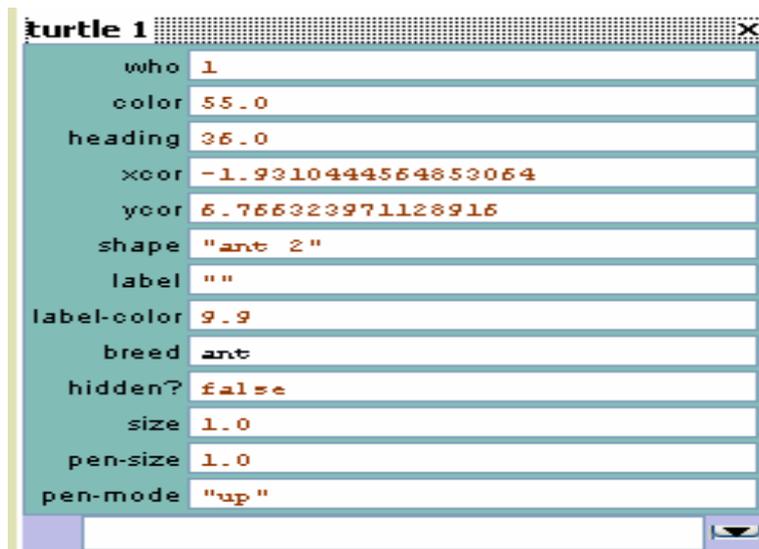
Elle contient l'identificateur de la tortue (≥ 0). On ne peut pas modifier cette variable.

Example: show values-from (turtles with [color = red]) [who] liste les identificateurs des tortues rouges.

xcor, ycor :

Ce sont les coordonnées de la tortue. Il existe également plusieurs variables patches prédéfinies : pcolor, plabel, plabel-color, pxcor, pycor. Leur fonction est la même que pour les variables tortues.

Tous les Informations sur une tortue ou sur une race inclue dans le tableau de la **figure 4.1**.



who	1
color	55.0
heading	36.0
xcor	-1.9310444564853064
ycor	6.766323971128916
shape	"ant 2"
label	" "
label-color	9.9
breed	ant
hidden?	false
size	1.0
pen-size	1.0
pen-mode	"up"

Figure 4.1 : tableau qui définit l'état d'une fourmis

IV.2.8 Les fonctions :

Une fonction se définit de la manière suivante :

```
To nom_fonction
  Corps_de_la_fonction
End
```

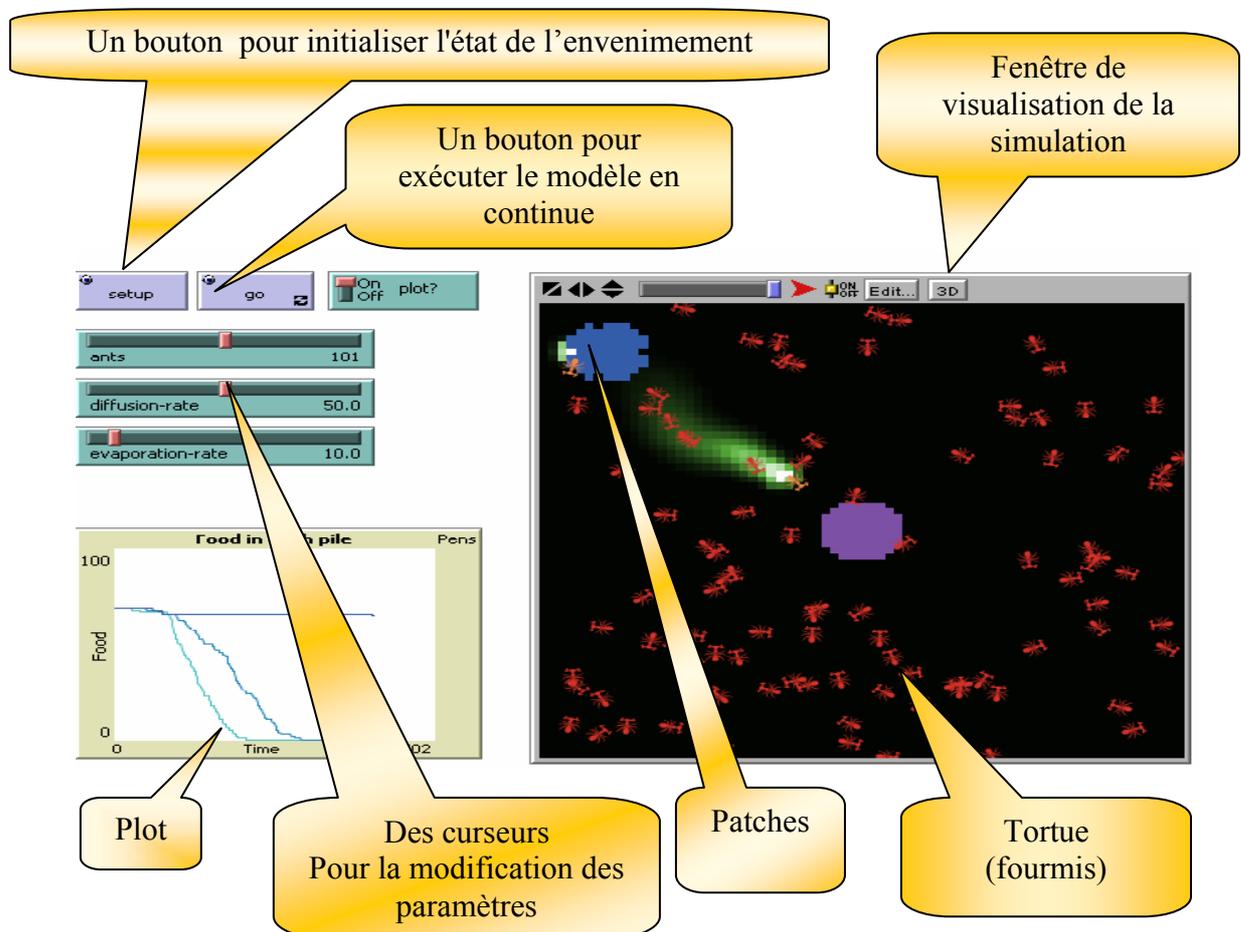
Les fonctions prédéfinies du langage NetLogo sont répertoriées dans une page du logiciel appelée *The Primitives Dictionary*.

IV.2.8 Affichage de résultats:

Pour ce qui concerne l'affichage d'une courbe : après avoir créé une fenêtre permettant l'affichage d'une courbe dans l'interface graphique, il suffit de mettre à jour à chaque pas de temps la valeur de la courbe (ou des courbes) que l'on souhaite afficher. Considérons les lignes de code suivantes :

- ❖ **Set-current-plot** "évolution des opinions" ;nécessaire seulement si plusieurs boîtes plot.
- ❖ **Set-current-plot-pen** "point" ; nécessaire si plusieurs stylo pour la boîte.
- ❖ **Plot opinion**

IV.3 Exemple d'une interface du NetLogo:



IV.4 LES RESULTATS DE CLASSIFICATION

Cette partie est consacrée à la présentation des résultats. Tout d'abords nous présentons la classification d'une façon générale et puis nous effectuons la variation de quelques paramètres

ont une influence sur les fourmis et leurs capacités de classifier, dans cette partie les propriétés d'isolant sont (le nom, la forme, la couleur).

× : Élément d'un groupes d'isolant de type 1.

★ : Élément d'un groupes d'isolant de type 2.

▲ : Élément d'un groupes d'isolant de type 3.

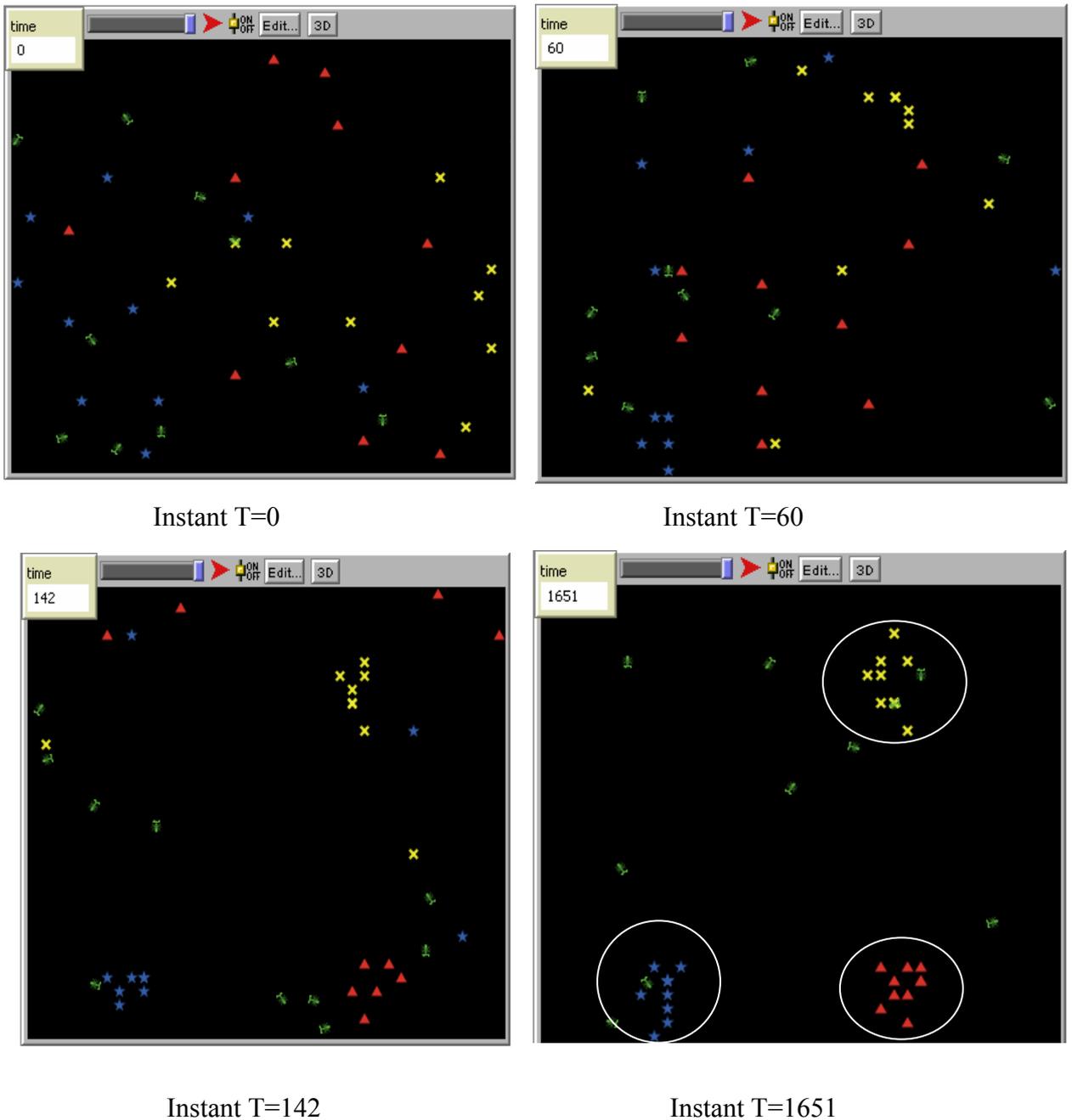


Figure 4.1 : les phases de classification

La **figure (4.1)** montre les phases de classification avec (le netlogo), elles représentent les étapes différentes de classification dans le temps. Ce qui implique que les fourmis améliorent la classification avec le temps.

IV.4 .1 La variation de rayon de détection

Pour la variation du rayon de détection nous avons choisis un rayon plus grand que celui utilisé dans la simulation précédente. Les résultats obtenus sont montrés dans la figure suivante.

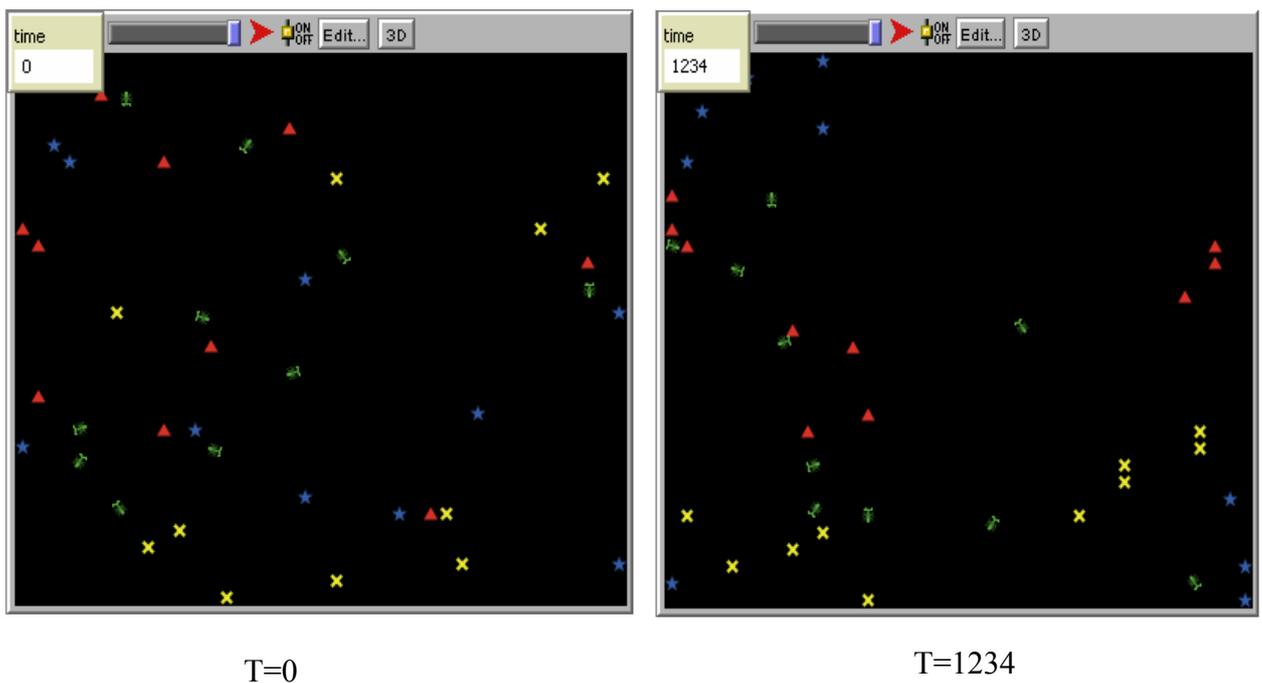


Figure 4.2 : les phases de classification avec une variation du rayon de détection
Avec un rayon $R=12$ pixels

La **figure (4.2)** ci-dessus montre les phases de classification avec une variation du rayon de détection, elle montre que les fourmis n'aboutissent jamais au but de classer les matériaux si elle n'a pas un rayon de détection préférable même si le temps de classification est des jours.

IV.4 .2 la variation des nombres des fourmis

Pour ceci, nous faisons varier le nombre d'agents-fourmis de 3 à 30 en fixant la taille de l'espace de déplacement des agents-fourmis et le nombre des objets à 20 pour tous les isolants. Les résultats obtenus sont présentés sur **figure 4.3**. Nous constatons que le temps de classification diminue avec l'augmentation du nombre de fourmis, cela implique que plus le nombre d'ensemble des fourmis est grand plus la classification des matériaux est plus facile et rapide.

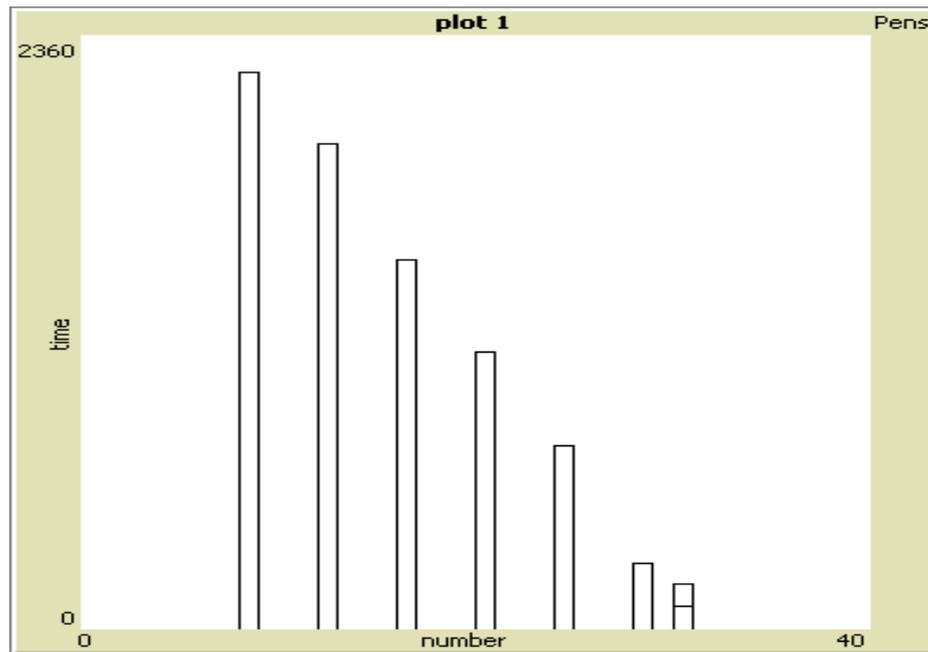


Figure 4.3 : le temps de classification en fonction du nombre des fourmis

IV.4 .2 la variation des nombres des d'objets

Dans ce test nous fixons le nombre d'agents égal a 10 agents fourmis et nous varions le nombre d'objets (isolants) de 10 à 50 et en fixant la taille de l'espace de déplacement des agents- fourmis. Les résultats obtenus sont présentés sur la **figure 4.4**. Dans cette figure nous constatons qu'il y'a une relation entre le temps de classification et le nombre de matériaux, quand le nombre augmente automatiquement le temps de classification augmente aussi.

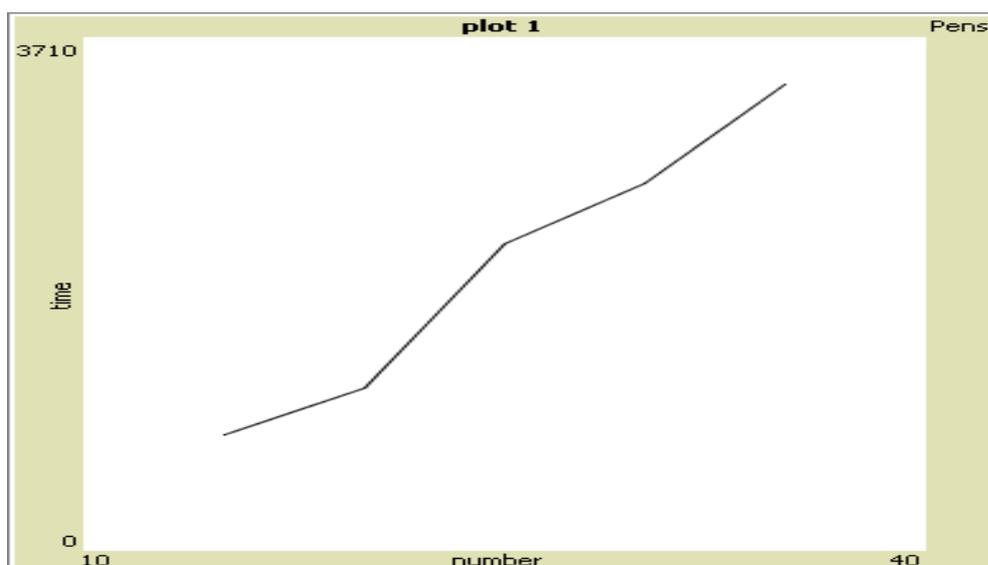
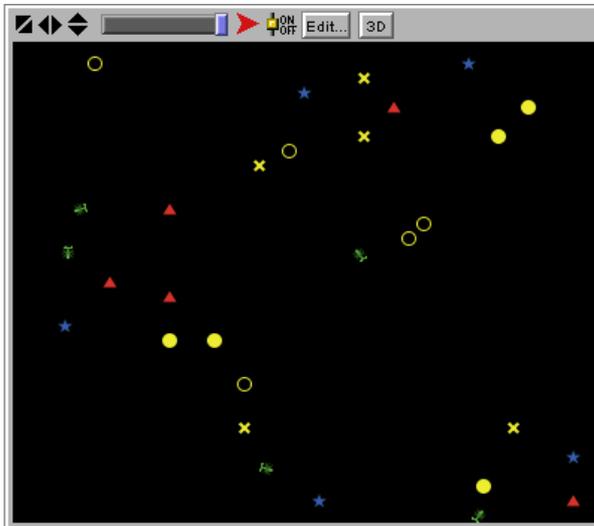


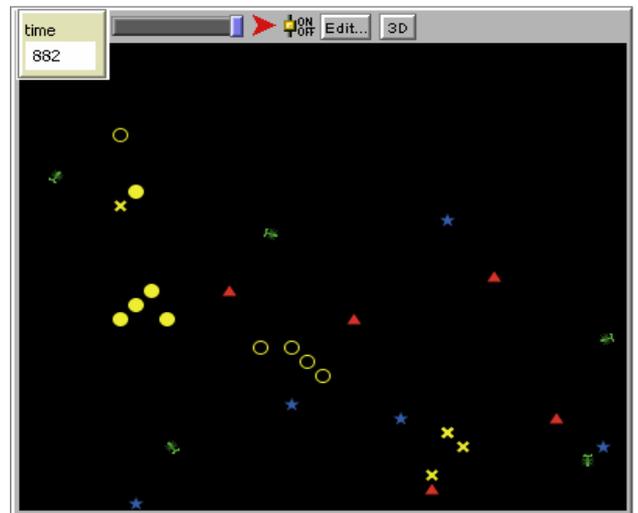
Figure 4.4 : le temps de classification en fonction du nombres d'objets

IV.4 .3 augmentation des types d'isolants

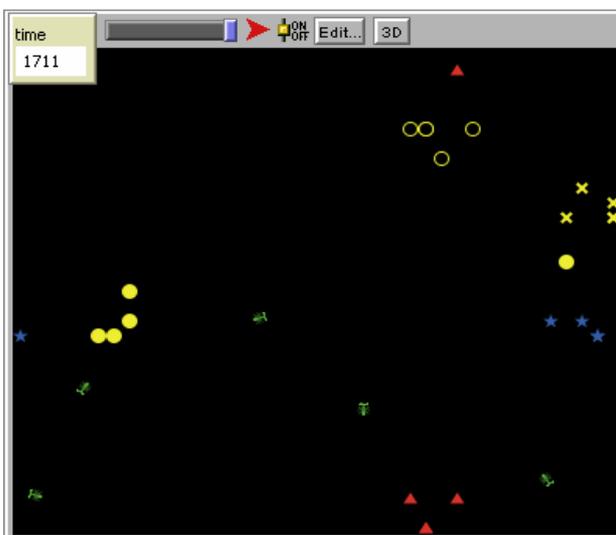
Dans ce test en va augmenter nombre du type d'isolants exemples (types, ●, ○, ☆, △, ×). Les résultats obtenus sont présentés sur **figure 4-5**.



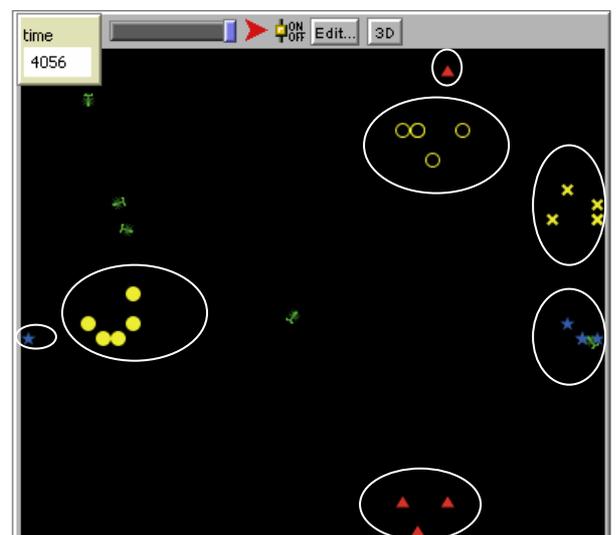
Instant T=0



Instant T=882



Instant T=1711

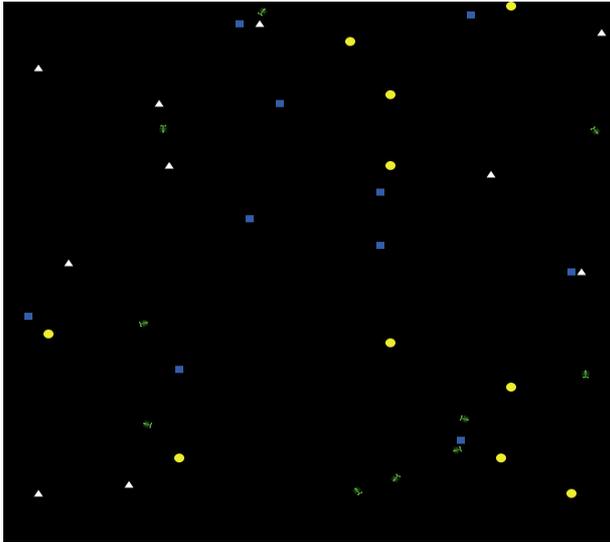


Instant T=4056

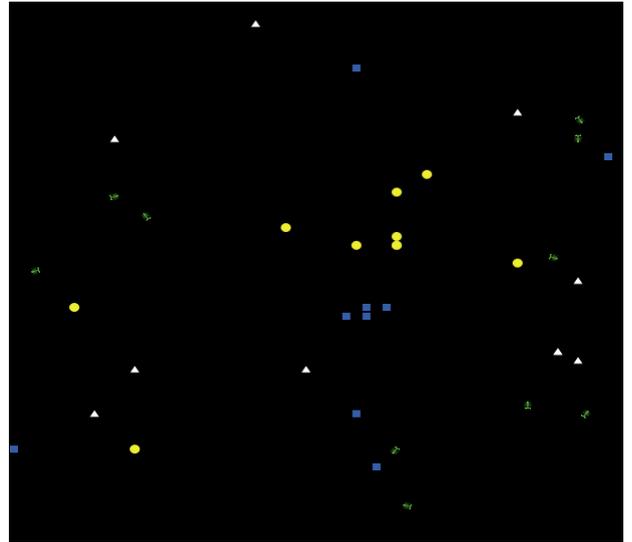
Figure 4.5 : addition d'autres types d'isolants

Pour la **figure (4.5)** elle présentent les différentes phases de classification quand nous ajoutons diverses races de matériaux, les fourmis classent avec succès mais dans un temps considérable.

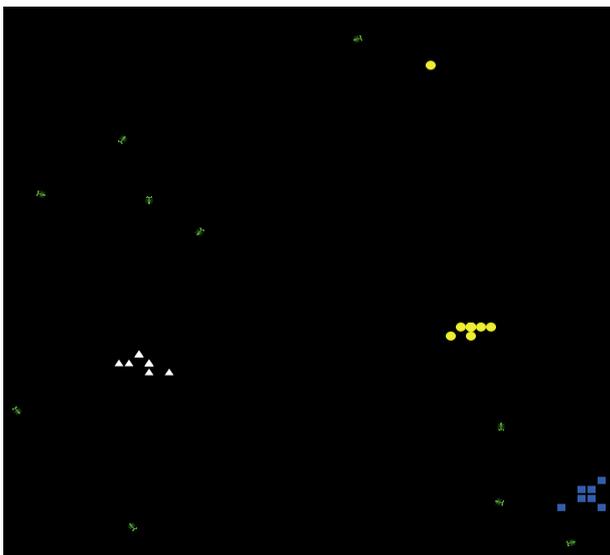
IV.4 .3 Changement de la taille de l'environnement : Pour ce ci, nous faisons varier la taille de l'espace de 19x16 à 40x40 en fixant le nombre d'agents fourmis à 10 et le nombre d'objets à 30. Les résultats obtenus sont présentés sur **Figure 4.6**



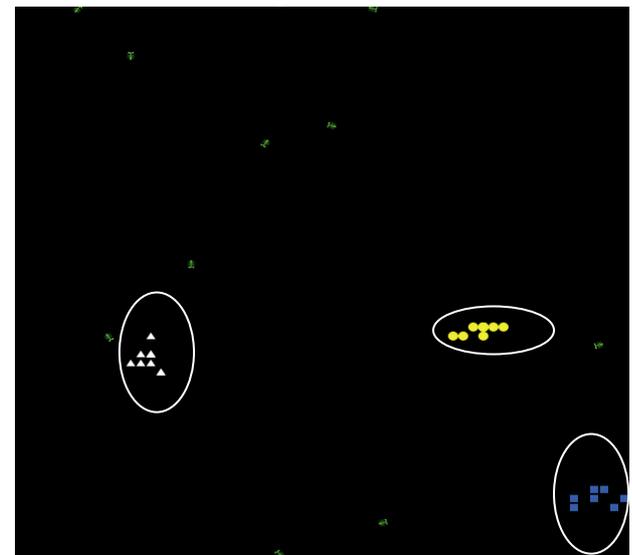
Instant T=0



Instant T=314



Instant T=9653



Instant T=10782

En ce qui concerne la mise à l'échelle de l'espace d'entraînement: plus la taille de l'espace augmente et plus le temps de regroupement augmente aussi. Ceci est tout à fait raisonnable puisque les agents fournis dans ce cas peuvent prendre plus de temps à se déplacer.

IV.5 CONCLUSION

Un système multi-agent est considéré constitué d'un ensemble d'agents placé dans un environnement composé d'objets qui ne sont pas des agents. Les agents appréhendent les objets et les actions des autres agents, réalisent des actions diverses en utilisant les objets disponibles de leur environnement et en unissant leurs actions pour définir des comportements collectifs. Pour cela il existe de nombreuses plates formes qui se basent sur les systèmes complexes, parmi ces logiciels il y a (madkit, starlogo, netlogo,), Dans notre travail le logiciel qui nous avons utilisée est le **Net logo**.

NetLogo est un logiciel de modélisation d'environnement pour simuler des phénomènes naturels et sociaux. Il est particulièrement adapté pour modéliser des systèmes complexes de développement à long terme. Les utilisateurs de NetLogo peuvent appliquer des instructions à des centaines ou des milliers d'agents opérant en parallèle. Le logiciel permet ainsi d'étudier la connexion qui existe entre le comportement d'individus isolés et d'en déduire des modèles qui découlent de l'interaction entre de nombreux individus. Ces simulations s'appliquent (entre autres) aux sciences naturelles et sociales, à savoir la biologie et la médecine, la physique et la chimie, les mathématiques et l'informatique, et l'économie. En plus Netlogo est bien entendu, gratuit sur Internet. D'autre part nous avons présenté les résultats de la simulation et l'interprétation des figures de la classification des isolants avec la méthode des fourmis artificielles dans l'environnement netlogo, Les résultats de la simulation présentés dans ce chapitre sont acceptables et reflètent la validité de la méthode utilisée. Bien que la méthode de classification non supervisée donne en général un nombre de classes égale au nombre de types, le fait qu'elle a donnée un nombre de classes égale au nombre de types dans notre cas est le simple fait du hasard.

CONCLUSION GENERALE

CONCLUSION GENERALE

Le travail présenté est une application de la classification non supervisée des matériaux diélectriques par la méthode des fourmis artificielles, basé sur des données symboliques (taille, couleurs, forme,...etc).

Ce travail nous a permis de nous hisser dans le monde des fourmis, et de nous familiariser avec les algorithmes de classification utilisant cette technique dans la résolution des problèmes de classification complexe, et avec les logiciels utilisés dans ce domaine.

La classification des matériaux diélectriques est l'un des domaines d'étude les plus important en électrotechnique. Dans ce mémoire un aperçu général sur les catégories des diélectriques et les divers paramètres influant directement ou indirectement sur l'état d'isolement a été présenté, ainsi qu'un exposé des méthodes utilisées dans la classification des données, qui Nous avons opté pour une méthode de classification intelligente par les colonies de fourmis vu leur facilité relative de leurs implantations par apport aux méthodes classiques, et leur aptitude d'auto - organisation (classification).

Les résultats de simulation obtenus par l'application des la technique des fourmis basé sur des données symboliques sont satisfaisants du coté précision comme du coté rapidité.

Une suite naturelle de ce travail serait de tenter d'utiliser la même technique avec un changement d'environnement et des données numériques réelles.

REFERENCES

- [Ahp-05].** J-Ah-Pine, J-Lemoine, H-Benhadda, " un nouvel outil de classification non supervisée de documents pour la découverte de connaissances et la détection de signaux faibles " Centre des Nouvelles Technologies de l'Analyse de l'Information,France 2005
- [Abd-06].** R. abdessemed " proposition d'une méthode de classification un environnement de robotique collective" mémoire pour l'obtention du diplôme de magister en informatique. Université de Batna en 07/06/2006.
- [Bar-05].** J-Pierre Barthélemy, F- Brucker "Analyse des données", en 15 décembre 2005
- [Ber-02].** P. Berkhin " Survey Of Clustering Data Mining Techniques". Rapport technique, San Jose, CA, Accrue Software, 2002.
- [Bou-03]** .A. Boubakeur , " Claquage des diélectriques liquides ", support de cours l'université de batna, 2002/2003.
- [Bon-94].**E. Bonabeau et G. Theraulaz éditeurs. " Intelligence Collective. Collection Systèmes Complexes ". Hermès, Paris, France, novembre 1994.
- [Bon-97].**E Bonabeau et G. Theraulaz G., " Auto- organisation et comportements collectifs : la modélisation des sociétés d'insectes, Auto- organisation et comportement " en 1997.
- [Bon-00].** E- Bonabeau ,G- Théraulaz " L'intelligence en essaim " ,en 14/04/2000
- [Buy-03].** P. Buyl, N- Tabti, A-Wagner "Auto organisation chez les insectes sociaux
- [Bou-06].** C Bouveyron " modélisation et classification des données de grande dimension application à l'analyse d'images " Thèse présentée pour l'obtention du grade de Docteur de l'université joseph fourier , spécialité mathématiques appliquées soutenue publiquement le 28 septembre 2006.
- [Cha-05].** J- C .chappelier, " Traitement informatique des donnes textuelles " cours **T.I.D.T** EPA lia département informatique, en 2006
- [Cle-04].**G. Cleuziou " Une méthode de classification non- supervisée pour l'apprentissage de règles et la recherche d'information" Thèse présentée pour l'obtention du grade de Docteur de l'Université d'Orléans , Discipline Informatique Soutenue le 8 décembre 2004.
- [Dar-59].** C. Darwin, "On the origin of species De l'origine des espèces". John Murrey, London, 1859

- [Did-71]**.E. Diday. " Une nouvelle méthode en classification automatique et reconnaissance des formes " : la méthode des nuées dynamiques. Revue de Statistique Appliquée, en 1971.
- [Did-72]**.E.Diday. " Optimisation en classification automatique et reconnaissance de formes " Note Scient. INRIA n° 6, 1972.
- [Did-84]**.E.Diday. " Une représentation visuelle des classes empiétantes", Les Pyramides. Rapport technique, INRIA n° 291, Rocquencourt 78150, France, 1984.
- [Den-87]**. J.L. Deneubour, BY C. Detrainj ,S Goss, Y. quinet "dynamics of collective exploration in the ant", Université Libre de Bruxelles, en 1987.
- [Dré-03]**. J-Dréo,P-Siarry " diverses techniques d'optimisation inspirées de la théorie de l'auto organisation dans les systèmes biologique " , Université de paris XII Val-de-Marne,en 24/08/2003.
- [Dor-96]**.M. Dorigo, V. Maniezzo, A. Colorni,, " The Ant System: Optimization by a Colony of Cooperating Agents " , IEEE Transactions on Systems, Man and Cybernetics-Part B, in 29-41. 1996.
- [Fun-01]**. G .Fung , "A comprehensive overview of basic clustering algorithms" en 2001.
- [Fou-01]** . R Fournié, " Techniques de l'ingénieur, traité Génie électrique "
- [Fra-06]** J.François Noubel " Intelligence Collective, la révolution invisible" www.TheTransitioner.org/ic
- [Gaf-91]**. T. Gafner, " Analyse critique des méthodes classiques et nouvelle approche par la programmation mathématique en classification automatique" Thèse pour obtenir le grade de docteur de l'université de Neuchâtel, en 18 décembre 1991.
- [Gna-77]**. R. Gnanadesikan, J. R. Kettenring et J.M. Landwehr "Interpreting and assessing the results of cluster analyses".Bulletin of the International Statistical Institute, vol.47, No 2, 1977, pp. 451- 463.
- [Gol-89]**. D. Goldberg Genetic Algorithms in Search," Optimization and Machine Learning", Addison-Wesley, in 1989.
- [Jain-88]** A.K Jain et R.C Dubes "Algorithms for clustering Data". Prentice Hall Advanced Reference series. en 1988
- [Joh-67]**. S.C Johnson , " Hierarchical Clustering Schemes " , *Psychometrika*, vol. 32, pp. 241-254, 1967.
- [Kar-99]**. G. Karypis, E-H.Han et V.Kumar. Chameleon " hierarchical clustering using dynamic modeling", computer, vol.32, no 8, 1999, pp. 68-75.

[Kau-90]. L. Kaufman et P. J. Rousseeuw. "Finding Groups in Data. An Introduction to Cluster Analysis" John Wiley & Sons, Inc., 1990.

[Kho-99]. L. Khodja, " Contribution à la Classification Floue non Supervisée " Thèse pour obtenir le grade de docteur de l'université de Savoie, en 25 décembre 1999.

[Ken -95] J. Kennedy , R-C. Eberhart, " Particle swarm optimization" . In Proceedings of the 1995 IEEE International Conference on Neural Networks, volume IV, pages 1942-1948, Piscataway, NJ, 1995. IEEE Service Center.

[Lab-03]. N. Labrousche " Modélisation du système de reconnaissance chimique des fourmis pour le problème de la classification non supervisée : application à la mesure d'audience sur Internet ", Thèse pour obtenir le grade de docteur de l'université de François Rabelais de Tours, en 4/décembre / 2003.

[Liu-96]. Liu.Huan et Setiono.Rudy. " Une approche probabiliste à la sélection de mode - une solution de filtre". Dans: Conférence internationale sur l'étude de machine, pp 319 - 327. - Bari, Italie, 1996.

[Loi-06]. M.Loï , M.Figueiredo et A. K- Jain " sélection de mode basé sur la classification", Dans: Avances dans les systèmes neuraux 15 en 2002 , pp 609-616. Vancouver, Canada, 2002

[Mac-06] J.macqueen "quelques méthodes pour la classification et l'analyse des observations multi variables", Proceedings of the Fifth Berkeley Symposium on Mathematical statistics and probability. pp. 281-297. Berkeley, 1967.

[McC-43]. W. McCulloch, and W. Pitts, "A logical calculus of the ideas immanent in nervous activity", Bulletin of Mathematical Biophysics 5: page 115-133, 1943.

[Men-65].G.J Mendel, "Experiments in Plant-hybridization". Princeton University Press, 1865.

[Men-01].C. Menguy, " Techniques de l'ingénieur, traité Génie électrique ".

[Mat-04] G.Mathieu " Système multi agent neuronal : Réalisation d'une bibliothèque d'agents neurones" Mémoire de stage présenté par Mathieu GALLET 11 juin 2004, Stage effectué à l'UFR des sciences du Havre.

[Mon-00]. N. monmarché, " Algorithmes des fourmis artificielles application à la classification et à l'optimisation. " Thèse pour obtenir le grade de docteur de l'université de tours, 20 décembre 2000

[Ode-01] J. Odell, H.V Parunak and B. Bauer Representing Agent Interaction Protocols in UML, Oriented Software Engineering, P. Ciancarini and M. Wooldridge Editors, Springer-Verlag, Berlin, pages 121-140,

[Oua-06]. S.Ouadfel " Contributions à la Segmentation d'images basées sur la résolution collective par colonies de fourmis artificielles " Thèse pour obtenir le grade de docteur de l'université de Batna, en 4/07/ 2006.

[Phi-03]. J.philippe rennard " auto organisation chez les insectes sociaux ", en 02/2003
" Printemps des sciences 2003

[Qui-04]. J.Quinqueton "Aspects socio organisationnels dans les systèmes multi agents:l'intelligence artificielle en essaim ", CERIC et LIRMM, Montpellier, en 9/12/2004.

[Rou-06]. M. Roux " Partition des centres mobiles pour données qualitatives " , Faculté des Sciences et Techniques (Case 462) ,Université Paul Cézanne (Marseille 3) ,Avenue Normandie-Niemen, 13397 Marseille Cedex 20, France

[Reg-99]. F.Regnier " développement de techniques d'analyse pour la construction d'un modèle de plaques tectoniques base sur l'itrf " mémoire présente en vue d'obtenir le diplôme d'ingénieur e.s.g.t. Soutenu le 5 Juillet 1999.

[Sys-91] Syswerda, G. "A study of reproduction in generational and steady-state genetic algorithms". In Rawlins, editor, Foundation of Genetic Algorithms. Morgan Kaufman, 1991.

[Vai-03]. P.Vaillant, " Classification automatique de textes ", Université des Antilles et de la Guyane, GEREC-F.en 03/12/2003

[Var-88].F. Varela "Autonomie et connaissance " essai sur le vivant Editions du seuil – 1988.

[Zia-06]. H.Ziadi,H-Zidani " formation des tas dans un environnement de robotique collective" mémoire présentée pour l'obtention du diplôme ingénieur d'état en informatique de l'université de Batna, en 2006.

[S1]. http://sis.univ-tln.fr/_tollari/ARTICLES/DEA2003/node3.html

[S2]. http://fr.wikipedia.org/wiki/Intelligence_collective#La_division_des_t.C3.A2ches

[S3]. "Self-Organizing Systems", <http://www.calresco.org/sos/sosfaq.htm>

[S4]. " Classification hiérarchique" , www.hds.utc.fr/sy09/documents/hierarchie.pdf

[S5]."Classification audio",www.lia.univavignon.fr/chercheurs/bonastre/Classification/pdf

[S6]." Classification AutomatiqueCours" www.math1.unice.fr/~rousseau/SV1/Classification-Cours.pdf

[S7]. http://fr.wikipedia.org/wiki/Algorithme_de_colonies_de_fourmis.

[S8]. http://edoc.bib.ucl.ac.be:61/ETD-db/collection/available/FUNDPetd-02232005-200950/unrestricted/05_chap1.pdf

[S9]. http://www.eecs.wsu.edu/~pedrow/HV_Engineering/lecture/bdgas/part1.